

3 – Node Hadoop Cluster Install Using Raspberry PI Hardware

***Group B:**

Pratap Timilsina

Steve Jones

John Ruo

Rooksana Sultana

Firasath Khan

Sam Whitaker



Group B Lessons Learned: we recommend that you read this list prior to completing the project.

- Keyboard from UK to US
- SSH early for easier configuration
- Wifi vs. Hardwired internet connection
- Label your Pi's so you don't mix them up
- Rename hadoop directory during install (Example: `hadoop -hadoop2.8.2`)

Table of Contents

1. Introduction and Solution Definition	5
1.1. What is Raspberry PI?	5
1.2. What is Hadoop?	5
1.3. Main Hardware Used During the Installation:	5
1.4. Installation Plan of the Cluster	6
1.5. Business Objective:	6
1.6. Skills required to execute steps listed in the installation guide	7
1.7. Logical diagram of the Hadoop Installation:	7
.....	Error! Bookmark not defined.
2.1. Install Raspbian	8
2.2. Pi Memory Enhancement	9
2.3. Network Configuration	10
2.4. Configure a static ip	11
If TP-Link: eth0	11
If Wifi: wlan	11
2.5. Java Verification	12
2.6. Update Hosts file with IP information	12
2.7. Update Hostname	13
2.8. Adding a Group, a User	13
2.9. Switch Users and Create SSH Key with No Passphrase	14
2.10. Enable SSH, Method # 1 and #2	15
2.11. Verification of Connection to NameNode	15
2.12. Confirm SSH is Enabled	15
3. Hadoop Software Install and Configuration	16
3.1. Install Hadoop Software	16
3.2. Extract All Files Under /opt	16
3.3. Updating /etc/bash.bashrc	17
3.4. Verification of .bashrc	17
3.5. Check Hadoop Version	17
3.6. Updating Hadoop Environment Variables	18
3.7. *Copy and paste the following statement to Hadoop-env.sh:	18
4. Hadoop Configuration File Settings	18
4.1. Update the core-site.xml File	19

4.2.	Update the mapred-site.xml File	19
4.3.	Update the hdfs-site.xml File	20
4.4.	Update the yarn-site.xml File	20
4.5.	Creating Folders and Permissions.....	21
4.6.	Start, Stop, and List Running Services.....	21
5.	Adding *Two More Nodes and Making a Hadoop Cluster	23
5.1	Copy configuration.....	23
5.2	Delete HDFS Storage and Add Permissions.....	23
5.3	Edit the /etc/hosts files on all three nodes.....	24
5.4	Update node1 slaves file	24
5.5	Update node1 Masters file	24
6	Scala and Spark Install	25
6.1	Get spark software	25
6.2	Untar spark software	25
6.3	Change the directory name and ownership to hduser.....	25
6.4	Add the following lines to spark-env.sh.....	25
6.5	Execute master script to start spark	26
6.6	Verify spark daemons	26
7	Issues and Anomalies	26
8	Cluster Verification and Examples	27
8.1	jps Command	27
8.2	GUI Verification Method.....	28
8.3	Run hdfs dfsadmin -report.....	29
8.4	Word Count Example	31
8.5	Spark Example	32
9	Glossary	32
10	How to Uninstall Cluster and Restore Hardware to Original State	33
11	References	33

1. Introduction and Solution Definition

The purpose of this installation guide is to install and configure a Hadoop Cluster using Raspberry Pi3 Model B boards.

1.1. What is Raspberry Pi?

Raspberry Pi is a very small, inexpensive, pocket size computer. Model B has a quad core ARM running at 900MHz with 1GB of RAM. * The Model B is equipped with Wifi which can be much more convenient than hardwired connection. Diagram 1 below depicts picture below.



Diagram 1

1.2. What is Hadoop?

Hadoop is a framework which allows for distributed computing.

The main components of Hadoop are HDFS (data storage), YARN (resource management), and MapReduce (data processing). Hadoop redundantly stores blocks of data across servers and copies the code/query to the data instead of moving the data to the code. Hadoop is highly scalable and cheaper to maintain when compared to most solutions that use other technologies as RAID and more proprietary in nature.

1.3. Main Hardware Used During the Installation:

- 3x Raspberry Pi 3 Model B with 32 GB micro SD cards
- 1x 5 port switch with Ethernet cables * (optional if not using Wifi)

- 1x TP-Link Internet Connection * (optional if not using Wifi)
- 3x mini power unit with power cables
- 1x wireless mouse and keyboard
- 1 Monitor (ideally hdmi connection supported)
- * QunQi case pack (this is to put the cluster together as one rack)

Below is the complete hardware inventory used



1.4. Installation Plan of the Cluster

- a) We will be installing a standalone Hadoop node named namenode and co-reside a datanode within.
- b) Then we will be adding the next two nodes as datanodes and make a 3-node cluster.
- c) When completed the cluster will have a namenode and two datanodes.

*** Important Note:** If the Rockhurst Network is not allowing for WiFi access, open a new tab. You should see a window which says “SafeConnect.” Click and provide credentials to login.

1.5. Business Objective:

***** This document will provide both high-level and detailed instructions to help Rockhurst students build a three-node cluster for the purpose of ingesting and analyzing big data.

Additional detail: There are many ways to install Hadoop for learning purposes. However, to create a cluster as multiple separate nodes are required there are many resource cost limitations. Raspberry PI is a good option to step away from virtualized cluster environment where students can build a cluster with actual physical hardware. Almost all training courses and training guides use virtualized environments when installing Hadoop. As we are using physical Raspberry PI boards it adds lot of value to installation and learning process. The diagram below explains the logical architecture of our cluster configuration with an explanation of which processes or Hadoop Daemons run on each node.

We have considered number of documents as listed in the reference section when coming up with these cluster configurations.

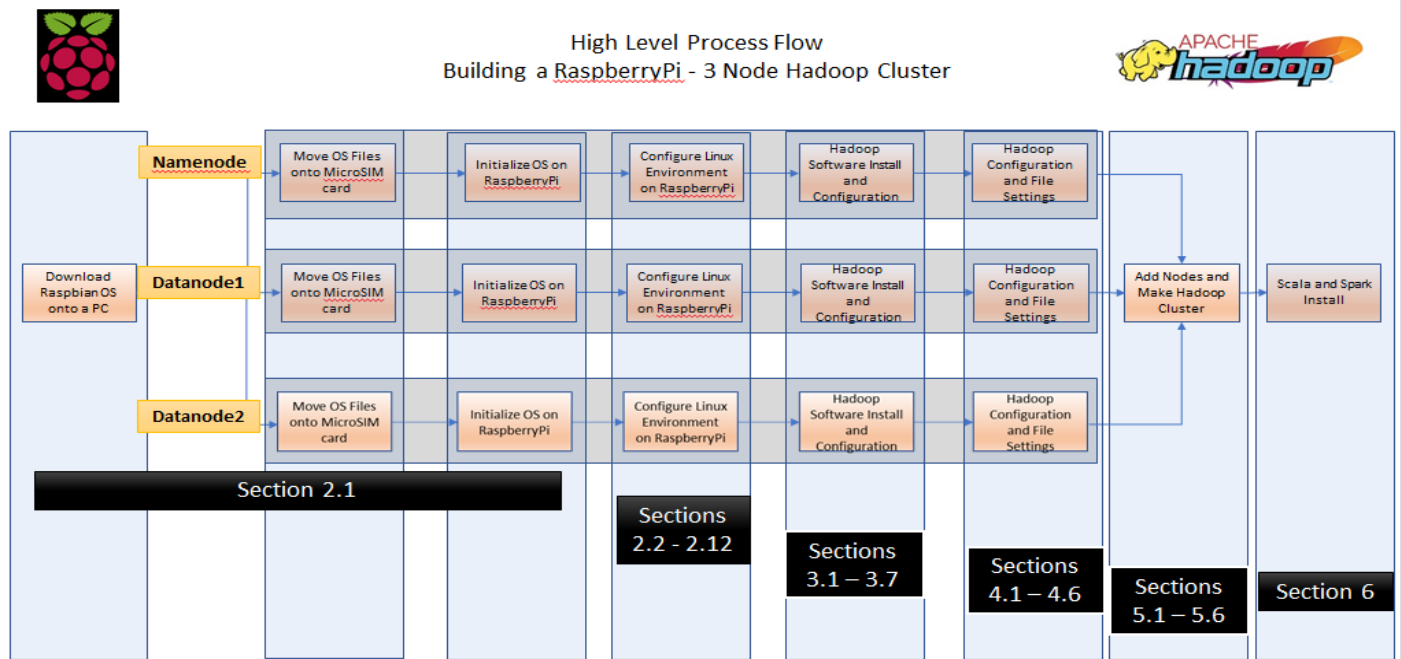
1.6. Skills required to execute steps listed in the installation guide

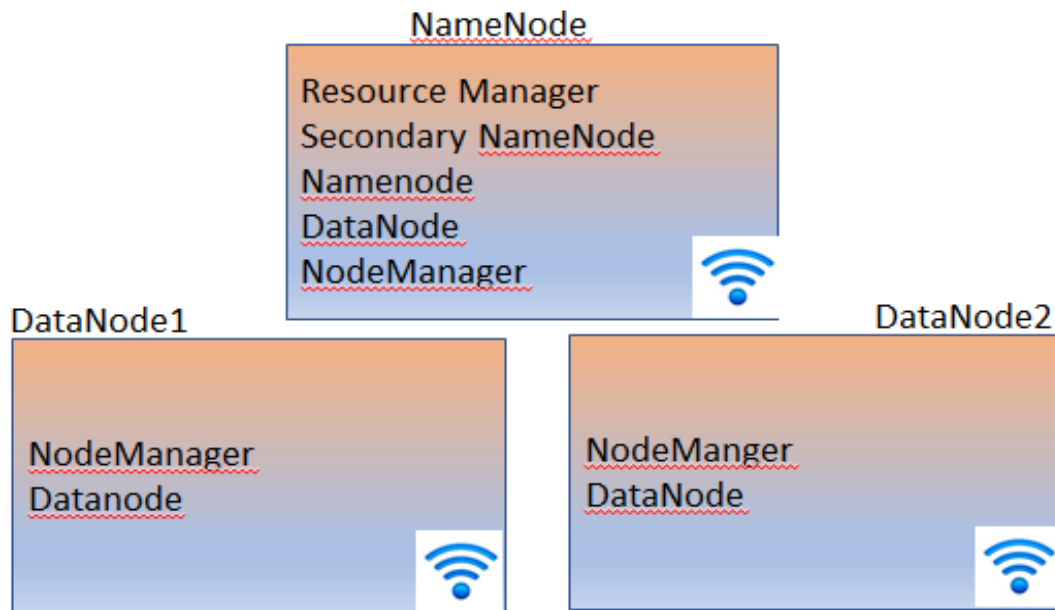
- Basic Unix/Linux OS skill set. A reference guide of general Linux command guide in the reference section. Even if someone is new to Unix/Linux one should still be able to run through the instructions.
- Basic Hadoop architecture knowledge will be helpful.
- * See embedded Word doc for common Linux commands used throughout the project:



Microsoft Word
Document

1.7. * Logical diagram of the Hadoop Installation:






Interpretation: by following the configuration diagram, you can actually perform setup one time and just copy two other times. This saves a significant amount of time since it saves you from having to perform two additional setups.

2. * Pre-Configuration of the Linux Environment for Hadoop

Note: Run this section as “pi” UNIX user unless mentioned differently. Pi user is the default user that gets created when Raspbian OS is installed on SD cards.

- Username: pi
- Password: raspberry

Note: if you want to use the pi’s for personal use, then we recommend you change the password. To do so, use command “passwd” and change to your preferred password!

Instructions	Screen Capture and Output	Time
2.1. Install Raspbian To start off we need to install Raspbian OS up and running before jumping into activity with Hadoop. <i>What is Raspbian?</i> Raspbian is a Linux system that is the operating system of Raspberry Pi Go to: https://www.raspberrypi.org/downloads/ .	Your steps: open this attachment and follow set-up pi steps  RaspberryPiDocumentation-PC group-D.pd	1hour per board

<p>Click on NOOBS and follow all prompts.</p> <p>Afterwards, you will place unzipped folders onto an SD card.</p> <p>Place SD into Raspberry Pi and follow all prompts.</p> <p>The complete guide of installing Raspberry PI OS is attached on the right.</p> <div><p>* <i>Note: before you start, we highly recommend changing the default keyboard from English (Great Britain) to English (United States)</i></p><p>* <i>Note: before you start, we highly recommend labeling each pi based on which purpose you want it to serve. We labeled master, node1, node 2</i></p></div>		
<p>2.2. Pi Memory Enhancement</p> <p>* We recommend applying this function to each raspberry pi in order to experience improved performance throughout the project.</p> <div><p>*Copy and paste the following</p><pre>sudo raspi-config</pre><pre>Adv.Option>>A3 Memory Split 64>>16</pre></div>		<p>~5 total minutes on 3 pi's</p>

2.3. Network Configuration

* *Note: from this point forward, you can choose to connect to a network via wifi or hardwired connection.*

* **If TP-Link:** During this install all Hadoop Daemons will be contained on the same node or one Raspbian PI 3 Model B.

***Copy and paste the following**

```
hostname -I to find the current id
```

* **If Wifi:** connect to wifi in top right corner of raspberry pi gui

Upon connecting, run the following to capture your wlan ip address. In our example, it was 192.168.1.14 but that will be different for you because you'll have a different raspbian. Note: # indicates output:

For static IP value in next step:

***Copy and paste the following**

```
ifconfig
```

output: #inet of wlan =<192.168.1.8>

* For static routers value in next step:

***Copy and paste the following**

```
netstat -nr
```

output: gateway <192.168.1.1>

File Edit Tabs Help

```
pi@raspberrypi:~$ scrot /home/pi/hadoop-screen-shots/
glib error: Saving to file /home/pi/hadoop-screen-shots/ failed

pi@raspberrypi:~$ scrot /home/pi/hadoop-screen-shots.png
glib error: Saving to file /home/pi/hadoop-screen-shots.png failed

pi@raspberrypi:~$ scrot -s
pi@raspberrypi:~$ scrot -s
pi@raspberrypi:~$ hostname -I
192.168.1.227 2605:a601:aa1:8c00:bc41:e46a:d65e:9516
pi@raspberrypi:~$
```

10min

2.4. Configure a static ip

* If TP-Link: eth0

run the following command in order to update dhcpd.conf:

***Copy and paste the following**

```
sudo nano /etc/dhcpd.conf
```

As you can see, the IP we provided is address is

192.168.1.8 which the static IP I assigned.

Make sure you are running as PI User.

Open a terminal and check the IP assigned by executing following command:

For this static IP to take affect reboot node.

***Copy and paste the following**

```
sudo reboot -i
```

If Wifi: wlan

run the following command in order to update dhcpd.conf:

* Note output from step 2.3 for both ip address and router:

***Copy and paste the following**

```
sudo nano /etc/dhcpd.conf
interface wlan0
static ip_address =192.168.1.8
static_routers=192.168.1.1
static_domain_name_servers=8.8.8.8
```

Make sure you are running as PI User.

Open a terminal and check the IP assigned by executing following command:

File Edit Tabs Help

GNU nano 2.2.6

File: dhcpd.conf

```
interface eth0
static ip_address=192.168.1.8/24
#static routers=192.168.1.1
static domain_name_servers=8.8.8.8 8.8.4.4
# A sample configuration for dhcpd.
# See dhcpd.conf(5) for details.

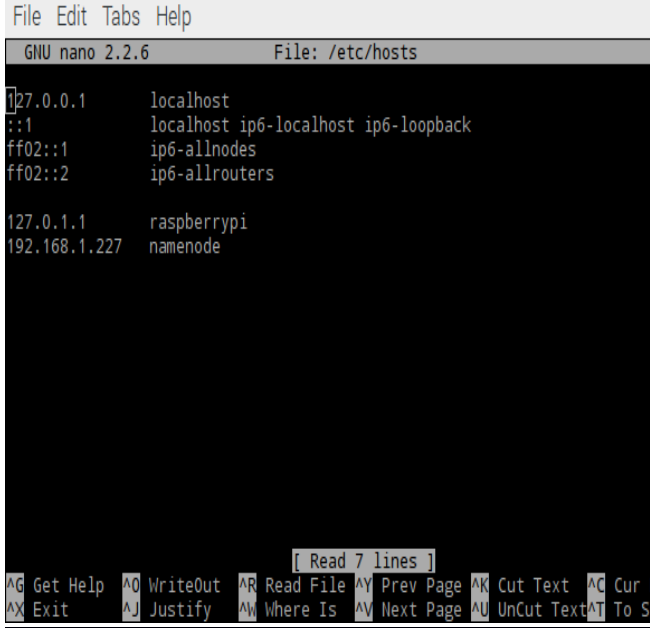
# Allow users of this group to interact with dhcpd via the control sock
#controlgroup wheel

# Inform the DHCP server of our hostname for DDNS.
hostname

# Use the hardware address of the interface for the Client ID.
clientid
# or
# Use the same DUID + IAID as set in DHCPv6 for DHCPv4 ClientID as per R
#duid

AG Get Help  AO WriteOut  AR Read File  AY Prev Page  AK Cut Text  AC Cur
AX Exit     AJ Justify   AW Where Is  AV Next Page  AU UnCut Text AT To S
```

10min

<p>For this static IP to take affect reboot node.</p> <div>*Copy and paste the following</div> <pre>sudo reboot -i</pre>		
<p>2.5. Java Verification</p> <p>Java is already installed on your pi's due to raspbian. Run the following command to verify java version. A version over 1.7 should work for Hadoop 2.7 or higher.</p> <div>**Copy and paste the following</div> <pre>java -version</pre>	<pre>java version "1.8.0_65"</pre> <p>Java(TM) SE Runtime Environment (build 1.8.0_65-b17)</p> <p>Java HotSpot(TM) Client VM (build 25.65-b01, mixed mode)</p>	5 min
<p>2.6. Update Hosts file with IP information</p> <p>Run the following command:</p> <div>**Copy and paste the following</div> <pre>sudo nano /etc/hosts</pre> <p>Add the IP assigned during the static IP setup in section 2.3 as IP address and provide the hostname as namenode.</p> <p><i>Note: The name "namenode" can be named as anything but going with Hadoop architecture as used this name.</i></p> <p>To save changes: press Control+X, then yes, then press enter.</p>	 <pre>File Edit Tabs Help GNU nano 2.2.6 File: /etc/hosts 127.0.0.1 localhost ::1 localhost ip6-localhost ip6-loopback ff02::1 ip6-allnodes ff02::2 ip6-allrouters 127.0.1.1 raspberry 192.168.1.227 namenode [Read 7 lines] ^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur ^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To S</pre>	1 min

<h2>2.7. Update Hostname</h2> <p>Run the following command:</p> <div>*Copy and paste the following</div> <pre>sudo nano /etc/hostname</pre> <p>Replace raspberrypi with namenode (Again as mentioned is step4 you can assign it anything for your own cluster.</p> <p>Also, I'm going to update my hosts file to make things a little easier when looking up each machine (once we get the other two nodes up and running). By “mapping,” you can call the node without remembering the IP address. You can call by “namenode” instead</p> <p>To save changes: press Control+X, then yes, then press enter.</p>	<div>File Edit Tabs Help</div> <div>GNU nano 2.2.6 File: /etc/hostname</div> <pre>namenode [Read 1 line] AG Get Help AO WriteOut AR Read File AY Prev Page AK Cut Text AC Cur AX Exit AJ Justify AW Where Is AV Next Page AU UnCut Text AT To S</pre>	5 min
<h2>2.8. Adding a Group, a User</h2> <p>Run the following commands:</p> <div>*Copy and paste the following</div> <pre>sudo addgroup hadoop sudo adduser --ingroup hadoop hduser sudo adduser hduser sudo</pre> <p>You'll need to enter a password but just use blanks/default values for everything else that is prompted.</p> <p>Note: hduser will be the Hadoop account userid.</p>	<div>File Edit Tabs Help</div> <pre>pi@datanode1:~\$ sudo addgroup hadoop Adding group 'hadoop' (GID 1001) ... Done. pi@datanode1:~\$ sudo adduser --ingroup hadoop hduser Adding user 'hduser' ... Adding new user 'hduser' (1001) 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] y pi@datanode1:~\$ sudo adduser hduser sudo Adding user 'hduser' to group 'sudo' ... Adding user hduser to group sudo Done. pi@datanode1:~\$</pre>	3 min

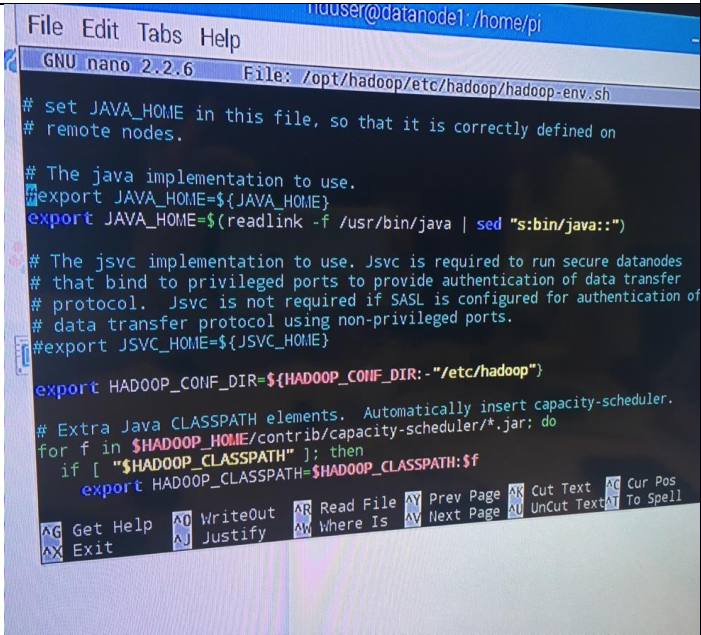
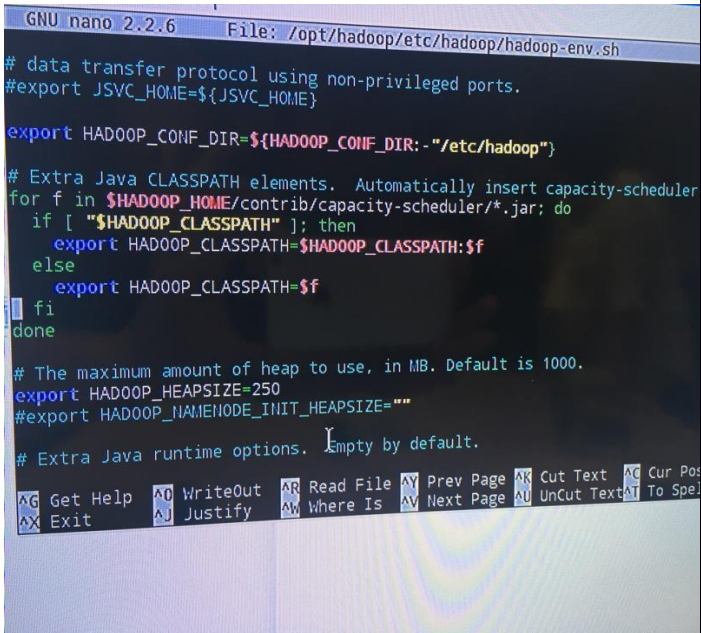
	<pre>hduser@namenode:/home/pi \$ id uid=1001(hduser) gid=1001(hadoop) groups=1001(hadoop),27(sudo) hduser@namenode:/home/pi \$</pre>	10 min
<p>2.9. Switch Users and Create SSH Key with No Passphrase</p> <p>Run the following commands:</p> <div data-bbox="97 816 712 1104" style="border: 1px solid black; background-color: #f4a460; padding: 10px;"> <p>*Copy and paste the following</p> <pre>su hduser mkdir ~/.ssh ssh-keygen -t rsa -P "" cat ~/.ssh/id_rsa.pub > ~/.ssh/authorized_keys</pre> </div> <p><i>The purpose of this setup is to make sure that passwordless login works. This is vital for cluster setup as the cluster used passwordless login to communicate among nodes during install. This is a critical step for node communication in Hadoop. Make sure this step works before moving on to next step.</i></p> <p><i>Note: we recommend that you perform 2.10 SSH after installing Raspbian in step 2.1.</i></p>	<div data-bbox="732 678 1373 1297"> <pre>File Edit Tabs Help pi@datanode1:~ \$ su hduser Password: hduser@datanode1:/home/pi \$ mkdir ~/.ssh hduser@datanode1:/home/pi \$ ssh-keygen -t rsa -P "" Generating public/private rsa key pair. Enter file in which to save the key (/home/hduser/.ssh/id_rsa): Your identification has been saved in /home/hduser/.ssh/id_rsa. Your public key has been saved in /home/hduser/.ssh/id_rsa.pub. The key fingerprint is: de:ee:11:87:31:01:60:7b:a8:62:e0:6a:1a:b2:31:25 hduser@datanode1 The key's randomart image is: +---[RSA 2048]-----+ 0... . 0 . . 0 . 0 + E.+ . \$ 0 . . + . . . 0 * . . . 0 += . . 0 . . 0 +-----+ hduser@datanode1:/home/pi \$ cat ~/.ssh/id_rsa.pub > ~/.ssh/authorized_keys hduser@datanode1:/home/pi \$</pre> </div>	1 min

<p>2.10. Enable SSH, Method # 1 and #2</p> <p>SSH has been disabled since 2016 in Raspbian.</p> <p>Enter <i>sudo raspi-config</i> into the terminal and follow the following steps to enable SSH:</p> <p>Select “Change User Password,” then drop down to “Interfacing Options.</p> <div style="border: 1px solid black; background-color: #f4a460; padding: 5px; margin-top: 10px;"> <p>*Copy and paste the following</p> <pre>sudo systemctl enable ssh sudo systemctl start ssh sudo reboot -i</pre> </div>		15 min
<p>2.10.5 Enable SSH, Method # 2</p> <p>a) The screenshot shows another method to update SSH by using the GUI.</p> <p>b) You can get to this point by clicking the applications menu, which has a raspberry as its symbol.</p>		
<p>2.11. Verification of Connection to NameNode</p> <p>Run the following commands:</p> <div style="border: 1px solid black; background-color: #f4a460; padding: 5px; margin-top: 10px;"> <p>*Copy and paste the following</p> <pre>su hduser ssh namenode exit</pre> </div> <p>If you are prompted to trust name/unknown_host, type yes.</p>		
<p>2.12. Confirm SSH is Enabled</p> <p>Run the following command:</p> <div style="border: 1px solid black; background-color: #f4a460; padding: 5px; margin-top: 10px;"> <p>*Copy and paste the following</p> <pre>ssh namenode</pre> </div> <p>You should not be prompted for a password.</p>		

3. Hadoop Software Install and Configuration

Instructions	Screen Capture and Output	Time
<p>3.1. Install Hadoop Software</p> <p>*Note: use the most recent stable version of hadoop. For us, it was 2.8.2.</p> <p>Run the following command as Pi User:</p> <div data-bbox="94 617 699 953" style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>*Copy and paste the following</p> <pre>wget http://www-eu.apache.org/dist/hadoop/common/hadoop-op-2.8.2/hadoop-2.8.2.tar.gzhttp://www-eu.apache.org/dist/hadoop/common/hadoop-op-2.8.2/hadoop-2.8.2.tar.gz</pre> </div>	<pre>--2017-06-25 16:09:39-- http://www-us.apache.org/dist/hadoop/common/hadoop-2.8.2/hadoop-2.8.2-src.tar.gz Resolving www-us.apache.org (www-us.apache.org)... 140.211.11.105 Connecting to www-us.apache.org (www-us.apache.org) 140.211.11.105 :80... connected. HTTP request sent, awaiting response... 200 OK Length: 18258529 (17M) [application/x-gzip] Saving to: 'hadoop-2.8.2-src.tar.gz' hadoop-2.8.2-src.ta 100%[=====>] 17.41M 2.36MB/s in 7.3s 2017-06-25 16:09:46 (2.39 MB/s) - 'hadoop-2.8.2-src.tar.gz' saved [18258529/18258529] pi@namenode:~ \$</pre>	30 min
<p>3.2. Extract All Files Under /opt</p> <p>Run the following commands. Make sure to enter your stable version of hadoop to enhance certainty that you're leveraging the version you want:</p> <div data-bbox="94 1495 699 1894" style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>*Copy and paste the following</p> <pre>sudo tar -xvzf hadoop-2.8.2.tar.gz -C /opt/ pi@datanode1:~ \$ cd /opt pi@datanode1:/opt \$ sudo mv hadoop-2.8.2 hadoop pi@datanode1:/opt \$ sudo chown -R hduser:hadoop hadoop pi@datanode1:/opt \$</pre> </div>		10 min

<p>3.3. Updating /etc/bash.bashrc</p> <p>Add to the end of /etc/bash.bashrc the following export lines:</p> <p>*Copy and paste the following</p> <pre>\$sudo nano ~/.bashrc # -- HADOOP ENVIRONMENT VARIABLES START -- # export HADOOP_HOME=/opt/hadoop export PATH=\$PATH:\$HADOOP_HOME/bin export PATH=\$PATH:\$HADOOP_HOME/sbin export HADOOP_MAPRED_HOME=\$HADOOP_HOME export HADOOP_COMMON_HOME=\$HADOOP_HOME export HADOOP_HDFS_HOME=\$HADOOP_HOME export YARN_HOME=\$HADOOP_HOME export HADOOP_COMMON_LIB_NATIVE_DIR=\$HADOOP_HOME/lib/native export HADOOP_OPTS="-Djava.library.path=\$HADOOP_HOME/lib"</pre>	<p>File Edit Tabs Help</p> <p>GNU nano 2.2.6 File: /etc/bash.bashrc Modified</p> <pre>/usr/lib/command-not-found -- "\$1" return \$? elif [-x /usr/share/command-not-found/command-not-found]; then /usr/share/command-not-found/command-not-found -- "\$1" return \$? else printf "%s: command not found\n" "\$1" >&2 return 127 fi } fi export JAVA_HOME=\$(readlink -f /usr/bin/java sed "s:bin/java::") export HADOOP_HOME=/opt/hadoop export HADOOP_INSTALL=\$HADOOP_HOME export YARN_HOME=\$HADOOP_HOME export PATH=\$PATH:\$HADOOP_INSTALL/bin</pre> <p>AG Get Help AO WriteOut AR Read File AY Prev Page AK Cut Text AC Cur Pos AX Exit AJ Justify AV Where Is AV Next Page AU UnCut Text AT To Spell</p>	5 min
<p>3.4. Verification of .bashrc</p> <p>Note: Now log back in as hduser.</p> <p>Run the following command:</p> <p>*Copy and paste the following</p> <pre>echo \$HADOOP_HOME</pre> <p><i>This should give the output as shown. Basically, printing of the path shows that bashrc profile was executed correctly.</i></p>	<p>/opt/hadoop</p> <p>hduser@namenode:~ \$</p>	2 min
<p>3.5. Check Hadoop Version</p> <p>Run the following command:</p> <p>*Copy and paste the following</p> <pre>hduser@namenode:/opt \$ hadoop version</pre>	<p>Hadoop 2.8.2</p> <p>Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r b165c4fe8a74265c792ce23f546c64604acf0e41</p> <p>Compiled by jenkins on 2016-01-26T00:08Z</p> <p>Compiled with protoc 2.5.0</p> <p>From source with checksum d0fda26633fa762bff87ec759ebe689c</p>	1 min

<div data-bbox="94 157 657 231"> <h3>3.6. Updating Hadoop Environment Variables</h3> </div> <div data-bbox="94 268 496 306"> <p>Run the following command:</p> </div> <div data-bbox="94 331 699 438"> <div data-bbox="233 331 574 363">*Copy and paste the following</div> <div data-bbox="110 386 643 415"> <pre>sudo nano /opt/hadoop/etc/hadoop/hadoop-env.sh</pre> </div> </div> <div data-bbox="94 501 677 583"> <p>On the page that appears, Copy and paste the following statement (java path):</p> </div> <div data-bbox="94 606 699 787"> <div data-bbox="233 606 574 638">*Copy and paste the following</div> <div data-bbox="110 661 587 735"> <pre>export JAVA_HOME=\$(readlink -f /usr/bin/java sed "s:bin/java::")</pre> </div> </div>	<div data-bbox="711 157 1408 787">  </div>	<div data-bbox="1421 157 1500 195"> <p>4 Min</p> </div>
<div data-bbox="94 911 639 987"> <h3>3.7. *Copy and paste the following statement to Hadoop-env.sh:</h3> </div> <div data-bbox="94 989 699 1247"> <div data-bbox="233 989 574 1018">*Copy and paste the following</div> <div data-bbox="110 1043 662 1220"> <pre>sudo nano /opt/hadoop/etc/hadoop/hadoop-env.sh export HADOOP_HEAPSIZE=250</pre> </div> </div>	<div data-bbox="711 911 1408 1539">  </div>	<div data-bbox="1421 850 1500 888"> <p>3 min</p> </div>

4. Hadoop Configuration File Settings

Note: All configuration files that needs to be updated in section 4 are located under \$HADOOP_HOME/etc/hadoop or /opt/hadoop/etc/hadoop/

Instructions	Screen Capture and Output	Time
<p>4.1. Update the core-site.xml File</p> <p>To locate core-site.xml file, use the following command:</p> <div data-bbox="94 405 699 516"> <p>*Copy and paste the following</p> <pre>nano \$HADOOP_HOME/etc/hadoop/core-site.xml</pre> </div> <p>The goal of this step is to insert the correct property between the configuration within core-site.xml</p> <p>On the page that comes up, will change the default value to what we have in red.</p>	<p>* core-site.xml</p> <pre><configuration> <property> <name>fs.default.name</name> <value>hdfs://namenode:9000</value> </property> </configuration></pre>	5 Min
<p>4.2. Update the mapred-site.xml File</p> <p>This is where we'll tell MapReduce to use the YARN framework. The file doesn't exist so you'll need to make a copy from mapred-site.template.xml and edit it.</p> <p>Run the following command:</p> <div data-bbox="94 1224 699 1535"> <p>*Copy and paste the following</p> <pre>cd \$HADOOP_HOME/etc/hadoop mv mapred-site.xml.template mapred-site.xml nano \$HADOOP_HOME/etc/hadoop/mapred-site.xml</pre> </div> <p>You will copy and paste everything in red into the page that appears.</p>	<pre><configuration> <property> <name>mapreduce.frame- work.name</name> <value>yarn</value> </property> </configuration></pre>	5 Min

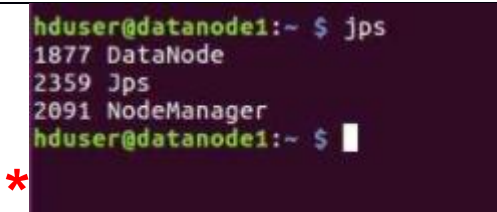
<p>4.3. Update the hdfs-site.xml File</p> <p>Run the following command to open the hdfs file:</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>*Copy and paste the following</p> <p><i>nano hdfs-site.xml</i></p> </div> <p>Input everything in the next column into the page that appears.</p>	<p>*</p> <pre> <property> <name>dfs.replication</name> <value>1</value> </property> <property> <name>dfs.namenode.name.dir</name> <value>file:/opt/Hadoop/hadoop_data/hdfs/namenode </value> </property> <property> <name>dfs.datanode.data.dir</name> <value>file:/opt/Hadoop/hadoop_data/hdfs/datanode< /val ue> </property> </pre>	<p>5 min</p>
<p>4.4. Update the yarn-site.xml File</p> <p>Next, we want to tell the Node Manager there is an auxiliary service called mapreduce.shuffle which needs to be implemented. Give class name as “mapreduce_shuffle”.</p> <p>Run the following command to open the yarn file:</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>*Copy and paste the following</p> <p><i>nano yarn-site.xml</i></p> </div> <p>Input everything in the next column into the page that appears.</p>	<p>*</p> <pre> <property> <name>yarn.nodemanager.aux-services</name> <value>mapreduce_shuffle</value> </property> <property> <name>yarn.nodemanager.aux- services.mapreduce.shuffle.class</name> <value>org.apache.hadoop.mapred.ShuffleHandler</ value> </property> </pre>	<p>5 min</p>

<p>4.5. Creating Folders and Permissions</p> <p>Run the following commands:</p> <div style="background-color: #f4a460; padding: 5px; border: 1px solid black;"> <p>*Copy and paste the following</p> <pre> sudo mkdir -p /opt/Hadoop/hadoop_data/hdfs/namenode e sudo mkdir -p /opt/Hadoop/hadoop_data/hdfs/datanode sudo chown hduser:hadoop /opt/Hadoop/hadoop_data/hdfs -R sudo chmod 750 /opt/Hadoop/hadoop_data/hdfs </pre> </div>		5 min
<p>4.6. Start, Stop, and List Running Services</p> <p>Run the following commands to start the services:</p> <div style="background-color: #f4a460; padding: 5px; border: 1px solid black;"> <p>*Copy and paste the following</p> <pre> cd \$HADOOP_INSTALL hdfs namenode -format cd HADOOP_HOME/sbin ./start-dfs.sh ./start-yarn.sh </pre> </div> <p>Run the following command to stop the services:</p> <div style="background-color: #f4a460; padding: 5px; border: 1px solid black;"> <p>*Copy and paste the following</p> <pre> ./stop-dfs.sh ./stop-yarn.sh </pre> </div>	<p>*Note: screenshot displays “datanode1” but you’ll need to stay consistent with our use of namenode by using “namenode”</p> <p style="color: red; font-weight: bold; font-size: 1.2em;">*</p>	5 min

The following command will list all running services as its output:

jps

Ensure you have the same services running as is shown in the screenshot.



Note that at this point we have a single node Hadoop running. If all goes well you will processes running as in section 4.6. At this point we do not have a cluster but have a standalone fully functional Hadoop node. The complete install of Hadoop single node takes ~90 – 120 minutes.

At this point the single node install is complete! One can verify this by running “jps” command as shown above is step 4.6 and verifying the daemons/processing running and/or login into the GUI.

<http://namenode:50070>

How to Hadoop...How to Hadoop...14.04 - ssh: conn...Apache Downloa...hadoop2 - Hadoo...Namenode inform...

namenode:50070/dfshealth.html#tab-overview

Started:Sun Jun 25 18:25:30 CDT 2017

Version:2.7.2, rb165c4fe8a74265c792ce23f546c64604acf0e41

Compiled:2016-01-26T00:08Z by Jenkins from (detached from b165c4f)

Cluster ID:CID-dbee25b4-abb0-44bb-becf-ee49aa9ad404

Block Pool ID:BP-205447443-192.168.1.227-1498433076202

Summary

Security is off.
Safemode is off.
1 files and directories, 0 blocks = 1 total filesystem object(s).
Heap Memory used 22.95 MB of 37.86 MB Heap Memory. Max Heap Memory is 241.75 MB.
Non Heap Memory used 21.84 MB of 22.21 MB Committed Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	27.25 GB
DFS Used:	28 kB (0%)
Non DFS Used:	5.88 GB
DFS Remaining:	21.37 GB (78.42%)
Block Pool Used:	28 kB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)



Connect Power to each pi board

Now we will add two other nodes as datanodes and expands this single node into a three-node cluster.

***Note:** When installing the cluster, install Hadoop as standalone by following sections 2.1 to 4.6 on each node. Basically, each Raspbian PI node will be installed as a standalone node and then we execute section 5 below onwards to make it a Hadoop cluster. Each installation of the two nodes will take an average of 45-50 min. After performing these steps, we connected the nodes physically together and then followed the steps below.

5. Adding *Two More Nodes and Making a Hadoop Cluster

Instructions	Screen Capture and Output	Time Period
<p>5.1 Copy configuration</p> <p>5.2 Delete HDFS Storage and Add Permissions</p> <p>Run the following commands:</p> <div style="border: 1px solid black; background-color: #f4a460; padding: 10px; margin: 10px 0;"> <p>*Copy and paste the following</p> <pre>rm -rf /opt/hadoop/hadoop_data sudo mkdir -p /opt/Hadoop/hadoop_data/hdfs/namenode (not required for nodes 2 and 3) sudo mkdir -p /opt/hadoop/hadoop_data/hdfs/datanode sudo chown hduser:hadoop /opt/hadoop/hadoop_data/hdfs -R sudo chmod 750 /opt/Hadoop/hadoop_data/hdfs</pre> </div> <p>Repeat the above commands for the other nodes.</p>		5 Min

5.3 Edit the /etc/hosts files on all three nodes	Example of Before Change:	10min
<div>*Copy and paste the following</div> <div>sudo nano /etc/hosts</div>	* sudo nano /etc/hosts 127.0.0.1 localhost 172.25.194.179 namenode 172.25.194.189 datanode1 172.25.194.178 datanode2	
<p>Verify that all three host files on all three nodes are the same.</p> <p>Note that the three IP's should match the static ip for each node on specific cluster. An example is shown here.</p>		
5.4 Update node1 slaves file		4 min
<div>*Copy and paste the following</div> <div>\$sudo nano /opt/hadoop/etc/hadoop/slaves</div>	namenode datanode1 datanode2	
<p>All three nodes in our cluster have a datanode configured.</p>		
5.5 Update node1 Masters file	namenode	4 Min
<div>*Copy and paste the following</div> <div>\$sudo nano /opt/hadoop/etc/hadoop/masters</div>		

6 Scala and Spark Install

In addition to Hadoop we installed Spark and Scala. A powerful tool and a scripting language introduced to the Hadoop framework. Spark uses in memory manipulation of data making it number of times faster when processing. This is an additional step to Hadoop cluster install. However simplicity of installation make it a viable option to expand and add additional functionality to the cluster.

Instructions	Screen shots	Time
6.1 Get spark software * Note: use the most recent stable version of spark and scala. <div> <div>*Copy and paste the following</div> <pre>wget https://d3kbcqa49mib13.cloudfront.net/spark-2.1.1-bin-hadoop2.7.tgz</pre> </div>		20 Min
6.2 Untar spark software <div> <div>*Copy and paste the following</div> <pre>sudo tar -zxf spark-2.1.1-bin-hadoop2.7.tgz -C /opt/</pre> </div>		4 Min
6.3 Change the directory name and ownership to hduser <div> <div>*Copy and paste the following</div> <pre>sudo mv spark-2.1.1-bin-hadoop2.7 spark sudo chown -R hduser:hadoop spark</pre> </div>		2 Min
6.4 Add the following lines to spark-env.sh nano /opt/spark/conf/spark-env.sh Note: The idea is that the Hadoop namenode or the master node is also the spark master node.	* <pre>SPARK_MASTER_IP=172.25.194.179 SPARK_WORKER_MEMORY=512m</pre>	5 Min

<p>6.5 Execute master script to start spark</p> <div data-bbox="94 296 714 415"> <p>*Copy and paste the following</p> <pre>./start-master.sh</pre> </div>		6 Min
<p>6.6 Verify spark daemons</p> <div data-bbox="94 541 714 783"> <p>*Copy and paste the following</p> <pre>hduser@namenode:/opt/spark/sbin \$ jps 4273 Jps 4233 Master</pre> </div>		5 Min

Note: The one node Spark Installation is a 30-minute installation effort.

7 Issues and Anomalies

- Attempted using Win32 Disk Imager to clone SD card. However, we ran into an issue with kernel panic. As a result, we used to install Hadoop software on individual nodes.
- * When starting the cluster ran into an anomaly. As a result, sometimes cleanup is required. All hduser files under should be deleted.

<p>*Copy and paste the following for cleanup commands</p> <pre>sudo rm -rf /opt/Hadoop sudo mkdir -p /opt/Hadoop/hadoop_data/hdfs/namenode sudo mkdir -p /opt/Hadoop/hadoop_data/hdfs/datanode sudo chown hduser:hadoop /opt/Hadoop/hadoop_data/hdfs -R sudo chmod 750 /opt/Hadoop/hadoop_data/hdfs</pre>
--

- Resolved an issue with IP change by editing the /etc/dhcpd.conf file. This assisted us in restarting the cluster without IP changing or configuring static IPs

8 Cluster Verification and Examples

Cluster verification methods and examples: can be used to verify which Hadoop daemons are running on each node. If the cluster is running properly the following list should appear on each node.

8.1 jps Command

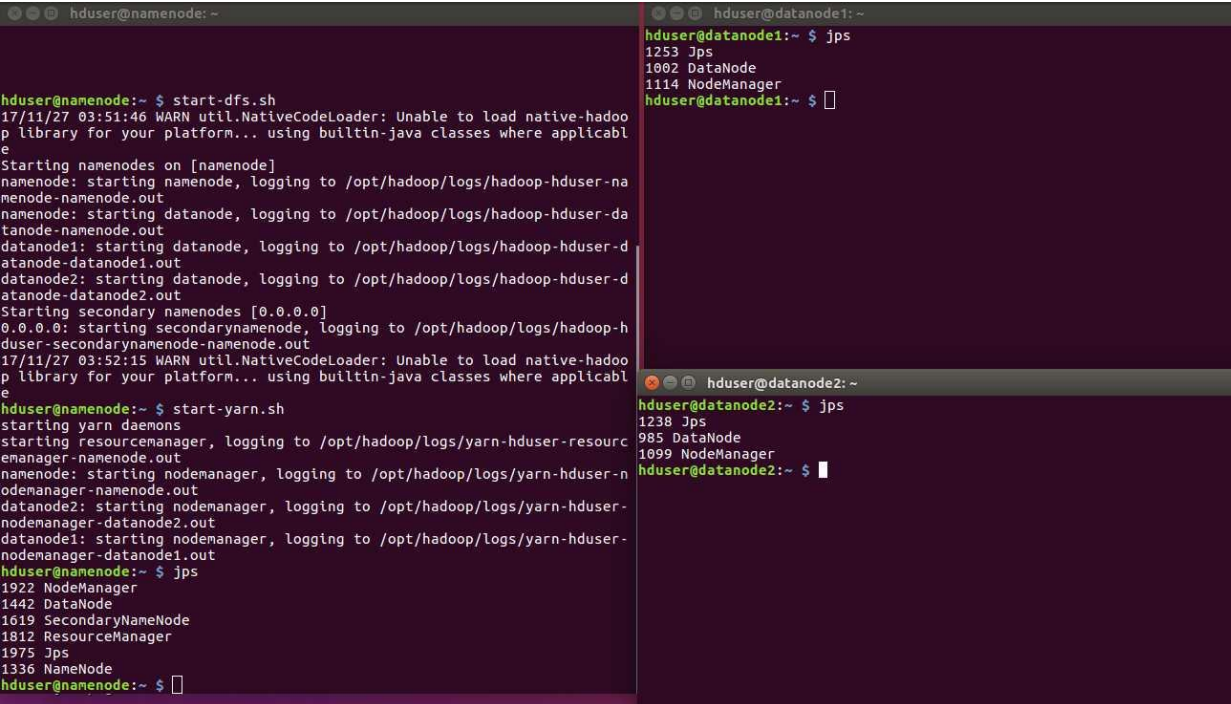
Namenode:

\$jps

```
2169 Jps
1232 NameNode
1310 SecondaryNameNode
1350 DataNode
1863 NodeManager
1750 ResourceManager
```

Datanode

```
2169 Jps
1357 DataNode
1864 NodeManager
```



The screenshot shows three terminal windows. The top-left window is the Namenode terminal, showing the execution of `start-dfs.sh` and `start-yarn.sh`, followed by the output of `jps`. The top-right window is the Datanode1 terminal, showing the output of `jps`. The bottom window is the Datanode2 terminal, also showing the output of `jps`. A red asterisk is placed next to the Namenode terminal's `jps` output.

```
hduser@namenode:~$ start-dfs.sh
17/11/27 03:51:46 WARN util.NativeCodeLoader: Unable to load native-hadoop
p library for your platform... using builtin-java classes where applicabl
e
Starting namenodes on [namenode]
namenode: starting namenode, logging to /opt/hadoop/logs/hadoop-hduser-na
menode-namenode.out
namenode: starting datanode, logging to /opt/hadoop/logs/hadoop-hduser-da
tanode-namenode.out
datanode1: starting datanode, logging to /opt/hadoop/logs/hadoop-hduser-d
atanode-datanode1.out
datanode2: starting datanode, logging to /opt/hadoop/logs/hadoop-hduser-d
atanode-datanode2.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /opt/hadoop/logs/hadoop-h
duser-secondarynamenode-namenode.out
17/11/27 03:52:15 WARN util.NativeCodeLoader: Unable to load native-hadoop
p library for your platform... using builtin-java classes where applicabl
e
hduser@namenode:~$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /opt/hadoop/logs/yarn-hduser-resourc
emanager-namenode.out
namenode: starting nodemanager, logging to /opt/hadoop/logs/yarn-hduser-n
odemanager-namenode.out
datanode2: starting nodemanager, logging to /opt/hadoop/logs/yarn-hduser-
nodemanager-datanode2.out
datanode1: starting nodemanager, logging to /opt/hadoop/logs/yarn-hduser-
nodemanager-datanode1.out
hduser@namenode:~$ jps
1922 NodeManager
1442 DataNode
1619 SecondaryNameNode
1812 ResourceManager
1975 Jps
1336 NameNode
hduser@namenode:~$ *
```

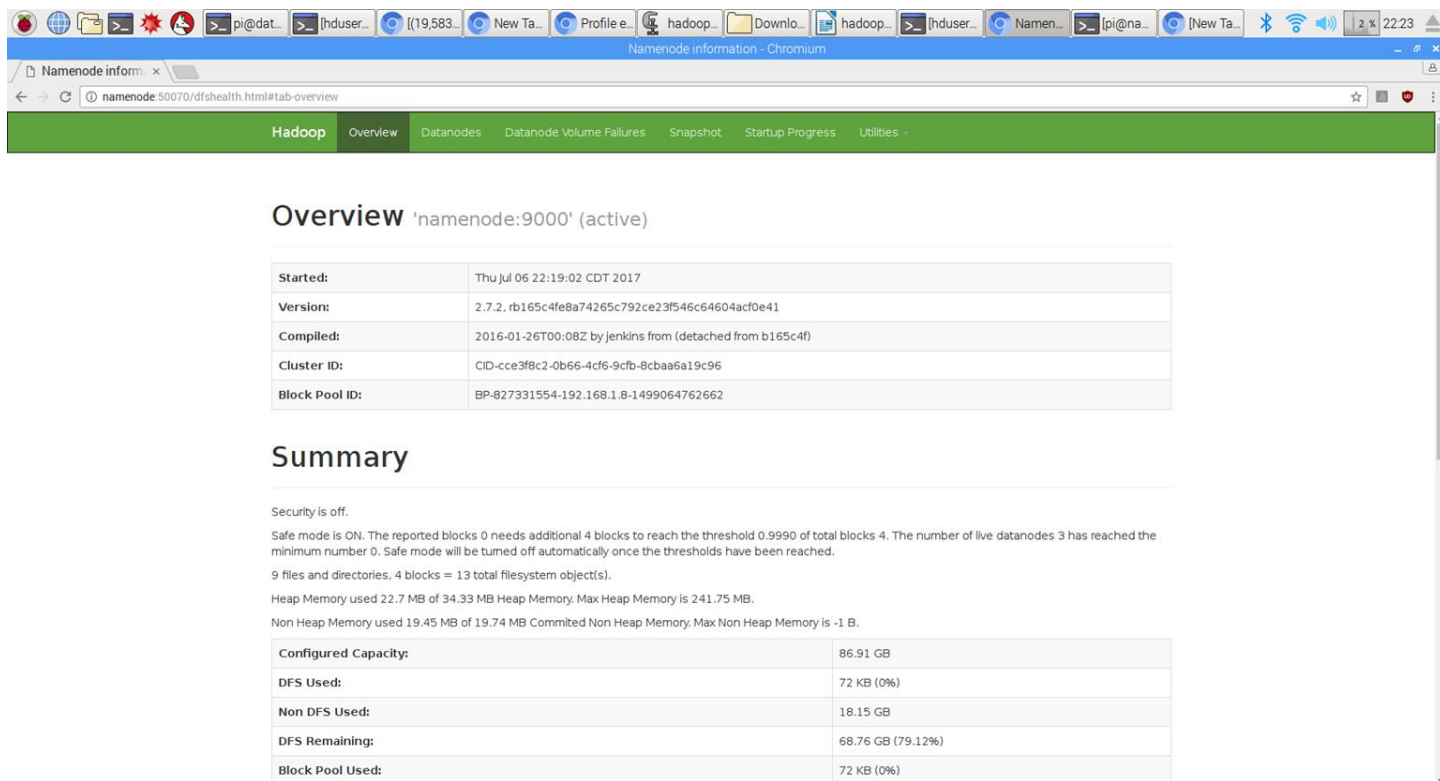
```
hduser@datanode1:~$ jps
1253 Jps
1002 DataNode
1114 NodeManager
hduser@datanode1:~$
```

```
hduser@datanode2:~$ jps
1238 Jps
985 DataNode
1099 NodeManager
hduser@datanode2:~$
```

Default ports for Hadoop and its application cluster. By logging into the GUI we can verify that cluster is up and running.

8.2 GUI Verification Method

<http://namenode:50070>



The screenshot shows a web browser window displaying the Hadoop NameNode web interface. The browser's address bar shows the URL `namenode:50070/dfshealth.html#tab-overview`. The page has a green header bar with the "Hadoop" logo and navigation tabs: "Overview", "Datanodes", "Datanode Volume Failures", "Snapshot", "Startup Progress", and "Utilities". The "Overview" tab is selected, showing the "Overview 'namenode:9000' (active)" page.

The Overview page contains a table with the following information:

Started:	Thu Jul 06 22:19:02 CDT 2017
Version:	2.7.2, rb165c4fe8a74265c792ce23f546c64604acf0e41
Compiled:	2016-01-26T00:08Z by jenkins from (detached from b165c4f)
Cluster ID:	CID-cce3f8c2-0b66-4cf6-9cfb-8c8aa6a19c96
Block Pool ID:	BP-827331554-192.168.1.8-1499064762662

Below the table, there is a "Summary" section. It contains the following text:

Security is off.

Safe mode is ON. The reported blocks 0 needs additional 4 blocks to reach the threshold 0.9990 of total blocks 4. The number of live datanodes 3 has reached the minimum number 0. Safe mode will be turned off automatically once the thresholds have been reached.

9 files and directories, 4 blocks = 13 total filesystem object(s).

Heap Memory used 22.7 MB of 34.33 MB Heap Memory. Max Heap Memory is 241.75 MB.

Non Heap Memory used 19.45 MB of 19.74 MB Committed Non Heap Memory. Max Non Heap Memory is -1 B.

Below the summary text, there is a table showing the configured capacity and usage:

Configured Capacity:	86.91 GB
DFS Used:	72 KB (0%)
Non DFS Used:	18.15 GB
DFS Remaining:	68.76 GB (79.12%)
Block Pool Used:	72 KB (0%)

<http://namenode:8088>

8.3 Run `hdfs dfsadmin -report`

This command gives a comprehensive health check on all nodes. Pay attention to the decommissioning status which should be Normal if the cluster is running ok. *

***Copy and paste the following**

```
hduser@namenode:/home/pi $ hdfs
dfsadmin -report
```

```
17/07/06 22:51:15 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Safe mode is OFF
Configured Capacity: 93319421952 (86.91 GB)
Present Capacity: 73823592448 (68.75 GB)
DFS Remaining: 73823506432 (68.75 GB)
DFS Used: 86016 (84 KB)
DFS Used%: 0.00%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
```

Missing blocks (with replication factor 1): 0

Live datanodes (3):

Name: 192.168.1.9:50010 (datanode1)

Hostname: datanode1

Decommission Status : Normal

Configured Capacity: 31106473984 (28.97 GB)

DFS Used: 28672 (28 KB)

Non DFS Used: 6151610368 (5.73 GB)

DFS Remaining: 24954834944 (23.24 GB)

DFS Used%: 0.00%

DFS Remaining%: 80.22%

Configured Cache Capacity: 0 (0 B)

Cache Used: 0 (0 B)

Cache Remaining: 0 (0 B)

Cache Used%: 100.00%

Cache Remaining%: 0.00%

Xceivers: 1

Last contact: Thu Jul 06 22:51:19 CDT 2017

Name: 192.168.1.8:50010 (namenode)

Hostname: namenode

Decommission Status : Normal

Configured Capacity: 31106473984 (28.97 GB)

DFS Used: 28672 (28 KB)

Non DFS Used: 6629531648 (6.17 GB)

DFS Remaining: 24476913664 (22.80 GB)

DFS Used%: 0.00%

DFS Remaining%: 78.69%

Configured Cache Capacity: 0 (0 B)

Cache Used: 0 (0 B)

Cache Remaining: 0 (0 B)

Cache Used%: 100.00%

Cache Remaining%: 0.00%

Xceivers: 1

Last contact: Thu Jul 06 22:51:17 CDT 2017

Name: 192.168.1.10:50010 (datanode2)

Hostname: datanode2

Decommission Status : Normal

Configured Capacity: 31106473984 (28.97 GB)

DFS Used: 28672 (28 KB)

Non DFS Used: 6714687488 (6.25 GB)

DFS Remaining: 24391757824 (22.72 GB)

DFS Used%: 0.00%

DFS Remaining%: 78.41%

Configured Cache Capacity: 0 (0 B)

Cache Used: 0 (0 B)

Cache Remaining: 0 (0 B)

Cache Used%: 100.00%

Cache Remaining%: 0.00%

Xceivers: 1

Last contact: Thu Jul 06 22:51:19 CDT 2017

8.4 Word Count Example

Copy the file,check HDFS for the file then run wordCount on the file

```
1. $hdfs dfs -copyFromLocal /opt/hadoop/LICENSE.txt /license.txt
2. $./hadoop jar /opt/hadoop/share/hadoop/mapreduce/hadoop-
mapreduce-examples-2.7.2.jar wordcount /license.txt /license
```

One can login to the GUI and under file tab can check a file name license that show wordcounts listed.

hduser@namenode:/home/pi \$

8.5 Spark Example

Run following command and make sure you get the scala prompt.

```
rduser@namenode:~$ spark-shell
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
17/11/27 03:58:32 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
17/11/27 03:59:04 WARN ObjectStore: Failed to get database global_temp, returning NoSuchObjectException
Spark context Web UI available at http://192.168.1.12:4040
Spark context available as 'sc' (master = local[*], app id = local-1511755117085).
Spark session available as 'spark'.
Welcome to

 _   _  _   _ 
| |_| |_| |_|
|_||_|_|_|_|_| version 2.2.0

Using Scala version 2.11.8 (Java HotSpot(TM) Client VM, Java 1.8.0_65)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

9 Glossary

HDFS: Hadoop Distributed File System. The main filesystem architecture on the data storage layer that support Hadoop ecosystem.

YARN: Yet another Resource Negotiator. Introduced in Hadoop 2.0 YARN main function is to provide resource management.

Namenode: The main daemon that manages and Hadoop resources and keeps track of block information. Also known as Master node.

Datanode: Slaves Daemons that run on the datanode. Also known as slaves.

Static IP: static IP is a constant IP that does not change. In the Raspberry IP cluster eth0 interface is provided a static IP so that IP is kept consistent upon reboot.

ResourceManager: Yarn main daemon that tracks and manages resources. In this cluster, it runs in the namenode.

NodeManager: Yarn daemon that runs on each datanode. Works with the resourcemanager to ask for resources as needed.

Mapreduce: MapReduce is a core component of the Apache Hadoop software framework.

Spark: Apache Spark is an open-source cluster-computing framework. Spark provides an interface for programming entire clusters with implicit data parallelism and fault-tolerance and in-memory data processing.

Scala: A general-purpose programming language providing support for functional programming and a strong static type system. Works hand-in-hand with Spark.

jps - Java Virtual Machine Process Status Tool.

sudo – A unix command that allows users created in a superuser group to be able to run root or administrator commands.

10 How to Uninstall Cluster and Restore Hardware to Original State

- Remove all connectivity and separate hardware.
- Format the SD card (remove Hadoop and OS) by using tools like SDFormatter if necessary. If this process is followed Raspbian PI OS should be re-installed. The other option is to remove hadoop software directory and re-install Hadoop as shown below.

To do this execute following commands: *

***Copy and paste the following**

```
sudo rm -rf /opt/hadoop/
```

```
sudo rm -ef /opt/spark/
```

11 References

By Jason Carter (Main document used for Hadoop installation)

<https://medium.com/@jasoncarter/how-to-hadoop-at-home-with-raspberry-pi-part-1-3b71f1b8ac4e>

By Jonas Widriksson

<http://www.widriksson.com/raspberry-pi-hadoop-cluster/>

Primary Spark document

<http://bailiwick.io/2015/07/07/create-your-own-apache-spark-cluster-using-raspberry-pi-2/>

Secondary Spark document

https://www.cloudera.com/documentation/enterprise/5-6-x/topics/spark_first.html

Linux Guide

<http://ryanstutorials.net/linuxtutorial/>