Raspberry Pi Cluster Setup Guide

Considerations to Consider before starting

If your SD card size will vary you will want to build the head node using the smallest size of SD card. This will ensure that the image for that SD card will ALWAYS be able to be written to a similar sized SD or larger. If you start with a 64GB SD card you will not be able to write the image to a 16GB SD card.

Head Node

> Hardware:
>
> - Raspberry Pi board x 1
> - WiPi USB dongle x 1
> - SD Card 16GB+ x 1
> - Ethernet cable x 1
> - HDMI cable x 1
> - Power cable mini-USB x 1

Compute nodes

> Hardware:
>
> - Raspberry Pi board x 7
> - SD Card 16GB+ x 1
> - Ethernet cable x 1
> - Power cable mini-USB x 1

Additional Hardware

- 10 Port USB hub
- 16 Port gigabit switch

Setup, Installation, and Testing

> ### Step 1 - Install operating systems

Install Raspbian Lite on SD card for head unit(s) and each compute node

Raspbian Lite

Raspbian Install Guides

> ### Step 2 - Configure head node settings

Setup the locale settings to make sure the correct keyboard, language, timezone, etc are set. This will ensure we are able to enter the correct symbols while working on the command line.

Configure Locale:

Log in with username: **pi** and password **raspberry**

```
sudo raspi-config
```

Expand the filesystem (***Option 7***)

- Select ***Yes***

Setup Localization Options (***Option 4***)

- Set Locale (***Option I1***)
  - Unselect ***en_GB.UTF-8***
  - Select ***en_US ISO-8859-1***
  - Select ***en_US***

Under Localization Options:

- Set TimeZone (***Option I2***)
  - Select ***America***
  - Select ***Chicago***

Under Localization Options:

- Set Keyboard Layout (***Option I3***)
  - Use the default selected Keyboard
  - Select ***English (US)***
  - Use the default keyboard Layout
  - Select ***No compose key***
  - Select ***No***

Under Localization Options:

- Set Wi-Fi country (***Option I4***)
  - Select ***US United States***

On the main settings page (not under advanced options):

- Set Hostname (***Option 2***)
  - Set *Hostname* (***Option A2***)
  - Enter ***head***

Under Advanced options:

- Set Memory Split (***Option 7***)
  - Set *Memory Split* (***Option A3***)
  - Enter ***16***

Setup SSH service:

- Select ***Interfacing Options*** (***Option 5***)
  - Select ***SSH*** (***Option P2***)
  - Select ***Yes***
  - Select ***Ok***

Select *Finish* and *Yes* to reboot

> ### Step 3 - Configure head node network

Set a static address for the cluster facing netw ork interface connection *eth0*. Turn on w ireless and setup w ireless connection on netw ork interface connection *wlan0*. Turn on SSH service and then reboot the head node.

Add or edit */etc/network/interface* file to match below :

```
# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

iface eth0 inet manual

allow-hotplug wlan0
iface wlan0 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Setup *eth0*:

Edit */etc/dhcpcd.conf*:

```
sudo nano /etc/dhcpcd.conf
```

Add to the end of the file:

```
interface eth0
static ip_address=192.168.10.5
static domain_name_servers=8.8.8.8
```

Save and exit

Setup *wlan0* by adding w ireless netw ork credentials to */etc/wpa_supplicant/wpa_supplicant.conf*:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Choose secure netw ork settings or unsecure netw ork settings and add to the end of the file:

Secure netw ork settings:

```
network={
ssid="<network name>"
psk="<network password>"
}
```

Unsecure netw ork settings:

```
network={
ssid="<network name>"
key_mgmt=NONE
}
```

Reboot:

```
sudo reboot
```

Step 4 - Update the system

```
sudo apt update && sudo apt upgrade -y
```

Reboot:

```
sudo reboot
```

Step 5 - IP forwarding for nodes to access internet

Setup IP forw arding so that all compute nodes w ill have access to the internet for package installation and to dow nload any needed materials on later use.

Log in w ith username: **pi** and passw ord **raspberry**

Enable IPv4 Forw arding and Disable IPv6:

```
sudo nano /etc/sysctl.conf
```

Add the follow ing lines to the end of the file (this includes the IP forw arding rule from above):

```
# Enable IPv4 forwarding
net.ipv4.ip_forward = 1

# Disable IPv6
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

Save and exit

Update the configuration files:

```
sudo sysctl -p
```

Edit and Save the iptables:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE

sudo bash -c "iptables-save > /etc/iptables.rules"
```

Add settings to */etc/network/interfaces*:

```
sudo nano /etc/network/interfaces
```

Add the follow ing line at the end of the w lan0 section under w pa-conf line to make the changes persistent:

```
pre-up iptables-restore < /etc/iptables.rules
```

Save and exit

Update */etc/hosts_ file*:

Add the follow ing to the end of the file:

Note: At this point you w ant to assign and name all of your nodes that **WILL** be in your cluster and enter them in the hosts file. Below is an example of a 6 node cluster including the head node as one of the six. This file w ill be copied w ith the image to the compute nodes and w ill save you a step of developing and deploying the hosts file later.

```
127.0.1.1    head

192.168.10.3    nodeX
192.168.10.5    head
192.168.10.100    node0
192.168.10.101    node1
192.168.10.102    node2
192.168.10.103    node3
192.168.10.104    node4
```

Reboot:

```
sudo reboot
```

Install MPICH-3.2

Install prerequisite *Fortran* w hich w il be required for compiling MPICH. All other dependencies are already installed.

Step 1 - Install Fortran

```
sudo apt install gfortran
```

Step 2 - Install and Setup MPICH3

Create hpc group:

```
sudo groupadd hpc
```

Add pi user to hpc group:

```
sudo usermod -aG hpc pi
```

Create hpc directory in root:

```
sudo mkdir -p /software/lib

cd /software/lib
```

Take ow nership of /softw are:

```
sudo chown -R pi:hpc /software
```

Create build and install directory inside mpich3 directory:

```
cd /software/lib

mkdir mpich_3.2

cd mpich_3.2

mkdir build install
```

Dow nload mpich3 and untar:

```
wget http://www.mpich.org/static/downloads/3.2/mpich-3.2.tar.gz

tar xvfz mpich-3.2.tar.gz
```

Compile and install mpich3:

```
cd build

/software/lib/mpich_3.2/mpich-3.2/configure --prefix=/software/lib/mpich_3.2/install

make

make install
```

Activate environment variable:

```
export PATH=/software/lib/mpich_3.2/install/bin:$PATH
```

Add path to environment variables for persistance:

```
sudo nano ~/.bashrc
```

Add the follow ing to the end of the file:

```
# MPICH-3.2
export PATH="/software/lib/mpich_3.2/install/bin:$PATH"
```

Step 3 - Create list of nodes for MPI:

This list of nodes w ill need to be updated as you add nodes later. Initially you w ill only have the head node.

Create node list:

```
cd ~
sudo nano nodelist
```

Add the head node ip address to the list:

```
192.168.10.5
```

Note: Anytime you need to add a node to the cluster make sure to add it here as w ell as */etc/hosts* file.

Step 4 - Test MPI

Test 1 - Hostname Test

```
cd ~

mpiexec -f nodelist hostname
```

Should return:

```
head
```

Test 2 - Calculate Pi

```
mpiexec -f nodelist -n 2 /software/lib/mpich_3.2/build/examples/cpi
```

Should return similar:

```
Process 0 of 2 is on head
Process 1 of 2 is on head
pi is approximately 3.1415926544231318, Error is 0.0000000008333387
wall clock time = 0.003250
```

Step 5 - Setup SSH keys

Note: *Must be executed from head node as pi user*

Generate SSH key:

```
cd ~
```

```
ssh-keygen -t rsa -C "<username>@swosubta" -f ~/.ssh/id_rsa
```

Press 'Enter' for passphrase

Press 'Enter' for same passphrase

Transfer the key to the authorized_keys file:

```
cat ~/.ssh/id_rsa.pub > ~/.ssh/authorized_keys
```

Prepare for cloning

Shutdow n the head node:

```
sudo shutdown -h now
```

Save SD Image

At this point you w ill w ant to save an image of the head node. This w ill give you a fall back point if you make mistakes moving forw ard. You w ill also use this image to begin your node image.

Using the same guide as described in the beginning you w ill w ant to reverse the process of w riting an image to the SD and *read* an image from the SD and save that image to your PC. Now you have saved your SD like a checkpoint.

Sample name for SD image:

```
compute_node_mpi_stage_2017_01_03
```

Create Node image

The overview of this process:

1. Save image of *head node*.
2. On a new SD card w rite the *head node* image you just saved.
3. Boot the second SD you just created from the head node and make the follow ing changes for "Creating a Generic Node Image".
4. Save image of new ly created *generic compute node*.

At this point you have a copy of both the *head node* and *generic comput node* at the MPI stage. This is a checkpoint that you can fall back to if there are errors after this point.

This w ill be a repeatable process w hen completed. You w ill setup an initial *compute node* image using your saved *head node* image. You w ill go in and change specific settings to *generic settings*. Doing this w ill allow you to alw ays access your *generic compute node* image at the same IP address and hostname. You w ill then be able to set up the compute node image to a specific IP address and hostname. Follow ing this process w ill allow for prompt and efficient deployment of a cluster.

Raspbian Install Guides

Create Generic Node image

This w ill be a repeatable process w hen completed. You w ill setup an initial *compute node* image using your saved *head node* image. You w ill go in and change specific settings to *generic settings*. Doing this w ill allow you to alw ays access your *generic compute node* image at the same IP address and hostname. You w ill then be able to set up the compute node image to a specific IP address and hostname. Follow ing this process w ill allow for prompt and efficient deployment of a cluster.

> **Step 1 - Boot image and login**

Log in w ith username: **pi** and passw ord **raspberry**

> **Step 2 - Enter a generic ip address**

```
sudo nano /etc/dhcpcd.conf
```

Change the *eth0* ip address from:

```
static ip_address=192.168.10.5
```

To:

```
static ip_address=192.168.10.3
```

Also add to the end of the file:

```
static routers=192.168.10.5
```

Save and exit

> **Step 3 - Enter a generic hostname**

```
sudo nano /etc/hostname
```

Change:

```
head
```

To:

```
nodeX
```

Save and exit

> **Step 4 - Edit hosts file**

```
sudo nano /etc/hosts
```

Change:

```
127.0.1.1               head
```

To:

```
127.0.1.1               nodeX
```

Save and exit

> **Step 5 - Remove wireless connection information**

Edit *interfaces* file:

```
sudo nano /etc/network/interfaces
```

Remove:

```
allow-hotplug wlan0
iface wlan0 inet manual
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

pre-up iptables-restore < /etc/iptables.rules
```

Edit *wpa_supplicant.conf*:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Remove this section if you have a secure network:

```
network={
ssid="<network name>"
psk="<network password>"
}
```

Remove this section if you have an unsecure network:

```
network={
ssid="<network name>"
key_mgmt=NONE
}
```

> **Step 6 - Shutdown and create a new image of the SD**

```
sudo shutdown -h now
```

Now you will go back to WinDiskImager32 and save the image as a node image. This is a generic node image that you can quickly deploy and use to set up your cluster with.

Sample name for SD image:

```
compute_node_mpi_stage_2017_01_03
```

Setup Generic Node image

Raspbian Install Guides

> ### Step 1 - Copy generic node image created earlier to an SD card using WinDiskImager32.
>
> ### Step 2 - Boot and login to your system

Log in with username: **pi** and password **raspberry**

> ### Step 3 - Adjust */etc/hostname* file

```
sudo nano /etc/hostname
```

Change:

```
nodeX
```

To:

```
node0
```

Save and exit

Note: This number will increment by one each time you add a node and must be unique on your cluster.

> ### Step 4 - Adjust */etc/dhcpcd.conf*

```
sudo nano /etc/dhcpcd.conf
```

Change the *eth0* ip address from:

```
static ip_address=192.168.10.3
```

To:

```
static ip_address=192.168.10.100
```

Save and exit

> ### Step 5 - Edit hosts file

```
sudo nano /etc/hosts
```

Change:

```
127.0.1.1               nodeX
```

To:

```
127.0.1.1               node0
```

Save and exit

Deploy Head Node SSH Key

Issue the follow ing command for each node:

```
rsync -a --rsync-path="sudo rsync" ~/.ssh/authorized_keys pi@nodeX:~/.ssh/authorized_keys
```

Note: At this point you w ill just do this once to develop a compute node image w ith Slurm installed. After that is complete you w ill create a new generic image of the compute node. Once that is complete you can use that image to finish deploying your compute nodes for the rest of your cluster.

Install NTP

NTP is used to keep the cluster time close together using outside NTP servers to sync w ith the head node. All computer nodes w ill sync w ith the head node.

Reference:
http://raspberrypi.tomasgreno.cz/ntp-client-and-server.html
http://w w w.pool.ntp.org/zone/north-america

> **Head Node**
>
> Install NTP:

```
sudo apt install ntp
```

Edit the */etc/ntp.conf*:

```
sudo nano /etc/ntp.conf
```

Change:

```
server 0.debian.pool.ntp.org iburst
server 1.debian.pool.ntp.org iburst
server 2.debian.pool.ntp.org iburst
server 3.debian.pool.ntp.org iburst
```

To:

```
server 0.north-america.pool.ntp.org
server 1.north-america.pool.ntp.org
server 2.north-america.pool.ntp.org
server 3.north-america.pool.ntp.org
```

Restart NTP:

```
sudo /etc/init.d/ntp restart
```

> **Compute Node**

Set Head Node as NTP server.

Edit */etc/ntp.conf*:

Under `restrict ::1` add:

```
restrict 192.168.10.0 mask 255.255.255.0
```

Change:

```
#broadcast 192.168.123.255
```

To:

```
broadcast 192.168.10.255
```

Restart NTP service:

```
sudo /etc/init.d/ntp restart
```

Install Slurm on Head Node

> ### Step 1 - Install Slurm

```
sudo apt install slurm-wlm slurmctld
```

> ### Step 2 - Add configuration file

Create the new Slurm configuration file */etc/slurm-llnl/slurm.conf*:

```
sudo nano /etc/slurm-llnl/slurm.conf
```

Add the following to the file and save:

```
# slurm.conf file generated by configurator easy.html.
# Put this file on all nodes of your cluster.
# See the slurm.conf man page for more information.
#
ControlMachine=head
ControlAddr=192.168.10.5
#
#MailProg=/bin/mail
MpiDefault=none
#MpiParams=ports=#-#
ProctrackType=proctrack/pgid
ReturnToService=2
SlurmctldPidFile=/var/run/slurm-llnl/slurmctld.pid
#SlurmctldPort=6817
SlurmdPidFile=/var/run/slurm-llnl/slurmd.pid
#SlurmdPort=6818
SlurmdSpoolDir=/var/lib/slurm/slurmd
SlurmUser=slurm
#SlurmdUser=root
StateSaveLocation=/var/lib/slurm/slurmctld
SwitchType=switch/none
TaskPlugin=task/none
#
#
# TIMERS
#KillWait=30
#MinJobAge=300
#SlurmctldTimeout=120
#SlurmdTimeout=300
#
#
# SCHEDULING
FastSchedule=1
SchedulerType=sched/backfill
#SchedulerPort=7321
SelectType=select/linear
#
#
# LOGGING AND ACCOUNTING
AccountingStorageType=accounting_storage/none
ClusterName=raspi2
#JobAcctGatherFrequency=30
JobAcctGatherType=jobacct_gather/none
#SlurmctldDebug=3
SlurmctldLogFile=/var/log/slurm/slurmctld.log
#SlurmdDebug=3
SlurmdLogFile=/var/log/slurm/slurmd.log
#
#
# COMPUTE NODES
NodeName=node[0-6] Procs=1 RealMemory=768 State=UNKNOWN

PartitionName=raspi2 Default=YES  Nodes=node[0-6] State=UP MaxTime=INFINITE
```

Check if Slurm controller is running:

```
scontrol show daemons
```

Should return:

```
slurmctld slurmd
```

> ### Step 3 - Create Munge key

```
sudo /usr/sbin/create-munge-key
```

Agree to overw rite.

> ### Step 4 - Finish installs and start services

```
sudo systemctl enable slurmctld.service
sudo ln -s /var/lib/slurm-llnl /var/lib/slurm
sudo systemctl start slurmctld.service
sudo systemctl enable munge.service
```

Verify Slurm controller is running:

```
sudo systemctl status slurmctld.service
```

Will return feedback to the screen. Verify *Active* line states: ***active (running)***.

Verify Munge is running:

```
sudo systemctl status munge.service
```

Will return feedback to the screen. Verify *Active* line states: ***active (running)***.

> ### Step 5 - Add user to Slurm group

```
sudo adduser pi slurm
```

> ### Step 6 - Add and take ownership of Slurm log folder

```
sudo mkdir -p /var/log/slurm/accounting

sudo chown -R slurm:slurm /var/log/slurm
```

```
sinfo
```

Install Slurm on Compute Node

> ### Step 1 - Copy Slurm configuration and Munge files from *Head Node*

On *head node*:

```
rsync -a --rsync-path="sudo rsync" /etc/munge/munge.key pi@nodeX:/etc/slurm-llnl/slurm.conf
```

```
rsync -a --rsync-path="sudo rsync" /etc/slurm-llnl/slurm.conf pi@nodeX:/etc/slurm-llnl/slurm.conf
```

> ### Step 2 - Install Slurm daemon

Execute on *node0*:

SSH into *node0*:

```
ssh pi@node0
```

```
sudo apt install slurmd slurm-client
sudo ln -s /var/lib/slurm-llnl /var/lib/slurm
```

Finish install and start Slurm and Munge:

```
sudo systemctl enable slurmd.service
sudo systemctl restart slurmd.service
sudo systemctl enable munge.service
sudo systemctl restart munge.service
```

Verify Slurm daemon is running:

```
sudo systemctl status slurmd.service
```

Will return feedback to the screen. Verify *Active* line states: ***active (running)***.

Verify Munge is running:

```
sudo systemctl status munge.service
```

Will return feedback to the screen. Verify *Active* line states: ***active (running)***.

> ### Step 3 - Add user to Slurm group

```
sudo adduser pi slurm
```

> ### Step 4 - Add and take ownership of Slurm log folder

```
sudo mkdir -p /var/log/slurm/accounting

sudo chown -R slurm:slurm /var/log/slurm
```

Execute on *head node*:

```
sudo scontrol reconfigure

sudo scontrol update nodename="node[0-6]" state=resume
-If this command throws an invalid nodename error:
try updating each node individually with the command:
sudo scontrol update NodeName="nodeX" state=resume

sinfo
```

This should show all nodes in an idle state.

Deploying the Rest of the Cluster

By now you have developed a head node image that contains both MPI and Slurm. You have also developed a compute node image that contains both MPI and Slurm as well. Now you should go back to the instructions for "Create Node Image" to save both images and then use the compute node image to finish deploying your cluster. Saving these images at each stage gives you different configurations that you can easily deploy in the future and also allows you to have a checkpoint in case something goes wrong. You can write the saved node image to your SD and start from that point rather then starting from the beginning.

Add an ethernet adapter

> ### Add *eth0*:
>
> Edit */etc/network/interfaces* file:

```
sudo nano /etc/network/interfaces
```

Add below eth0 section:

```
auto eth1
iface eth1 inet manual
```

Change or add iptables rule to end of file:

```
pre-up iptables-restore < /etc/iptables_wired.rules
```

Create iptables rules file:

```
sudo nano /etc/iptables_wired.rules
```

```
# Generated by iptables-save v1.6.0 on Wed Sep 20 04□42 2017
*nat
:PREROUTING ACCEPT [3:228]
:INPUT ACCEPT [3:228]
:OUTPUT ACCEPT [3:228]
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -o eth0 -j MASQUERADE
-A POSTROUTING -o eth1 -j MASQUERADE
COMMIT
# Completed on Wed Sep 20 04□42 2017
```

> ### Disable *wlan0*:

Edit */etc/wpa_supplicant/wpa_supplicant.conf* file:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Comment out the `network={ connection information }` section (all lines)

Disable eth1 adapter:

```
sudo ifconfig eth1 down
```

Reboot:

```
sudo reboot
```

Now all traffic for the cluster is routed through eth0 and out eth1 to the internet. Any returning traffic or downloads come in via eth1 and through eth0 to the cluster unless its meant for the head node.

Troubleshooting Section:

> ### Received SIGHUP or SIGTERM from Nano
>
> Enter the command:

```
bash
```

> ### NETWORK UNREACHABLE:
>
> When experiencing network connectivity problems with compute nodes:

1. Flush the iptables in Memory

```
sudo iptables --flush
```

1. Delete the rules file

```
sudo rm -rf /etc/iptables.rules
```

1. Rebuild the rules and file
   Repeat the IP tables section of the guide, starting with the commands:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
```

1. Save the iptables.rules file:

```
sudo bash -c "iptables-save > /etc/iptables.rules"
```

1. Check for the iptable rules in the */etc/network/interfaces file*:

Make sure that the line below is present and not commented out:

```
pre-up iptables-restore < /etc/iptables.rules
```

If it is missing then add it to the end of the file. Save and exit.

MPI ISSUES

If mpiexec command fails to execute, stalls, or displays an error message about an unreadable path file:

- Mpich3 could be i
- the w rong directory
- Make sure the export path correlates to the actual install path for MPICH3
- Reinstalling MPICH3 and setting up the proper environment variables can fix many problems, re-evaluate the MPICH3 install instructions and verify all settings before attempting a reinstall.

SSH ISSUES

If the Pi is displaying SSH errors w hen running the mpiexec command:
Check the problematic node's authorized_keys file, and compare it w ith the head node's authorized_keys file.

Check the file by going to the SSH directory:

```
cd ~/.ssh
```

Now check the file information for *authorized_keys* file:

```
ls -ls
```

The filesize is listed after the ow ner and group names.

These file should be identical in length, if not redistribute the head node's authorized_keys file to the compute node using the follow ing command:

```
rsync -a --rsync-path="sudo rsync" ~/.ssh/authorized_keys pi@nodeX:~/.ssh/authorized_keys
```

COMMANDS TO CHECK SERVICE STATUSES

These commands do the same thing, just w ith a different syntax:

```
sudo systemctl [start,stop,restart,status] <service name>
```

```
sudo service <service name> [start,stop,restart,status]
```

```
sudo /etc/init.d/<service name> [start,stop,restart,status]
```

ENABLING/DISABLING NETWORK INTERFACE CONNECTIONS

Disable the specified connection

```
sudo ifdown <connection name>
```

Enable the specified connection

```
sudo ifup <connection name>
```

SLURM ISSUES

Make sure the slurm.conf file is identical across all nodes.

When running the service status command, read the error messages that are displayed: ***these messages are vital in order to troubleshoot current problems***.

PROBLEMATIC NODES

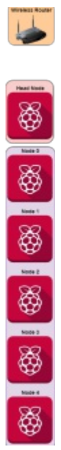
On many occasions, certain nodes fail to w ork because of a softw are/hardw are malfunction. This can be fixed by removing and reinstalling the softw are. Hardw are problems can be fixed by reformatting the node's SD card, and rew riting it w ith a functional node image. Also check each Ethernet cable for w eaknesses, and verify that each node in the cluster is properly connected.

-For Pi 3 Clusters: The head node is connected via Wi-Fi, and each compute node uses the head node's w ireless connection to dow nload files.

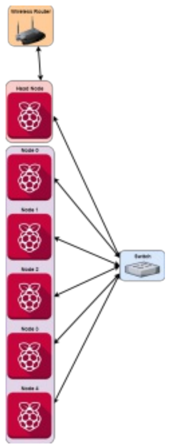
-For Pi 2 Clusters: A Wi-Pi adapter is a tested solution for establishing a w ireless connection w ith a Raspberry Pi model 2. Using other w ireless adapters could result in incompatible drivers or other various issues. The head node can also be connected to the Internet via an Ethernet cable.

Network Diagrams

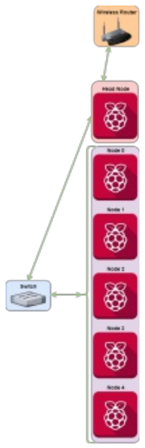
Base Equipment Layer (Pictured Below)



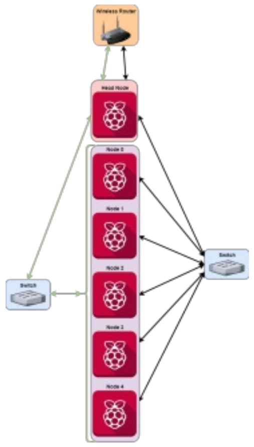
Physical Layer (Pictured Below)



Logical Layer (Pictured Below)



Physical and Logical Layers (Pictured Below)



References

https://w w w.modmypi.com/blog/how -to-give-your-raspberry-pi-a-static-ip-address-update

https://w w w.raspberrypi.org/forums/view topic.php?f=28&t=44609

http://w ew orkw eplay.com/play/automatically-connect-a-raspberry-pi-to-a-w ifi-netw ork/

https://w w w.raspberrypi.org/forums/view topic.php?f=36&t=162096

https://w w w.raspberrypi.org/forums/view topic.php?t=118804&p=808453

https://w w w.raspberrypi.org/forums/view topic.php?f=28&t=37575

http://unix.stackexchange.com/questions/88100/importing-data-from-a-text-file-to-a-bash-script

http://w w w.tldp.org/LDP/abs/html/arrays.html

http://how -to.w ikia.com/w iki/How _to_read_command_line_arguments_in_a_bash_script

http://ryanstutorials.net/bash-scripting-tutorial/bash-variables.php

http://stackoverflow .com/questions/19996089/use-ssh-to-start-a-background-process-on-a-remote-server-and-exit-session

http://stackoverflow .com/questions/29142/getting-ssh-to-execute-a-command-in-the-background-on-target-machine

http://w w w.igeekstudio.com/blog/setting-up-ganglia-and-hadoop-on-raspberry-pi

http://ccm.net/faq/2540-linux-create-your-ow n-command

https://github.com/XavierBerger/RPi-Monitor

https://w w w.raspberrypi.org/blog/visualising-core-load-on-the-pi-2/

https://github.com/davidsblog/rCPU

https://raseshmori.wordpress.com/2012/10/14/install-hadoop-nextgen-yarn-multi-node-cluster/

https://www.packtpub.com/hardware-and-creative/raspberry-pi-super-cluster