Raspberry Pi Cluster Guide

Parts of this guide were developed using an article by Garrett Mills titled Building a Raspberry Pi Cluster on Medium.com. Please see this article for further detailed information. The below guide gets right to the deployment instructions. This guide does deviate greatly at points to follow a more linear path for faster deployment and less back and forth between nodes. Also used is MPICH3 instead of OpenMPI.

A note on microSD card sizes

I will generally used 4GB microSD cards to build my images. Once I am satisfied with my final head node and compute node image I will put them on a larger microSD card such as a 16GB, 32GB, or 64GB card and expand the filesystem using `raspi-config`. This allows for a much smaller image files on your hard drive plus faster read and write times when working with imaging software.

Parts List

- 8 x Raspberry Pi 3 Model B (7 for compute nodes and 1 for head node)
- 8 x MicroSD cards
- 8 x Micro-USB power cables
- 1 x 8-port 10/100/1000 network switch
- 1 x 10-port USB power-supply
- 1 x 128GB USB flash drive

Before starting:

Download and install Raspberry Pi OS Lite. The easiest way to do this is by using the new Raspberry Pi Imager tool provided here.

Use this tool to install the Raspberry Pi OS Lite image directly to your microSD card for your head node. Instructions are provided on the above linked page.

> ### Step 1 - Generic node configuration

Using a keyboard and monitor you can complete the initial configuration.

Default username: `pi`
Default password: `raspberry`

1.1 Initial Configuration

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.

1.1.1 Configure Locale:

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

Start the Raspberry Pi configuration tool:

```
sudo raspi-config
```

Setup System Options:

Select `1 System Options`

- Select `S1 Wireless LAN`
  - Select `US United States`
  - Select `Ok`
  - Enter the SSID
  - Enter the passphrase
  - Select `Ok`

Select `1 System Options`

- Select `S4 Hostname`
  - Select `Ok`
  - Enter `nodeX` for the hostname
  - Press `Enter`

1.1.2 Configure Interfacing Options:

Select `3 Interfacing Options`

- Select `P2 SSH`
  - Select `Yes`
  - Select `Ok`
  - Press `Enter`

1.1.3 Configure Performance Options:

- Select `4 Performance Options`
  - Select `P2 GPU Memory`
  - Enter `16`
  - Press `Enter`

1.1.4 Configure Localisation Options:

- Select `5 Localisation Options`
  - Select Locale `L1 Change Locale`
  - Unselect `en_GB.UTF-8 UTF-8`
  - Select `en_US ISO-8859-1`
  - Press `Enter`
  - Select `en_US`

- Select `5 Localisation Options`
  - Select `L2 Change Timezone`
  - Select `US` (or appropriate country)
  - Select `Central` (or appropriate local timezone)

- Select `5 Localisation Options`
    - Select `L3 Change Keyboard Layout`
    - Use the default selected Keyboard
    - Press `Enter`
    - Select `Other`
    - Select `English (US)`
    - Select `English (US)`
    - Select `The default for the keyboard layout`
    - Select `No compose key`
    - Press `Enter`

1.1.5 Configure Advanced Options:

- Select `6 Advanced Options`
    - Select `A1 Expand Filesystem`
    - Select `Ok`

Tab to `Finish`

Select `No` when asked to reboot

1.2 Configure Wired Network

Setup *eth0* static ip address:

Edit */etc/dhcpcd.conf*:

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

Add to the end of the file:

```
interface eth0
static ip_address=192.168.10.3/24
static domain_name_servers=1.1.1.1
```

Save and exit

1.3 Update the system

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

1.4 Create hosts file

Update */etc/hosts* file by adding the following to the end of the file:

Note: At this point you want 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 8 node cluster including the head node as one of the six. This file will be copied with the image to the compute nodes and will save you a step of developing and deploying the hosts file later. Notice that the last octet of the ip address correlates to the node name. E.g 192.168.10.**100** on node**0**, 192.168.10.**101** on node**1**, etc.

Edit */etc/hosts* file:

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

Modify or add the following lines to the file:

```
127.0.1.1          nodeX

192.168.10.3       nodeX
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
192.168.10.105     node5
192.168.10.106     node6
192.168.10.107     node7
```

1.5 Shutdown Raspberry Pi and create generic node image

Shutdown the Raspberry Pi node:

```
sudo shutdown -h now
```

Using Win32DiskImager read the node image to your hard drive. Insert the SD card adapter with the SD card from the Raspberry Pi node into a USB port on your computer.

Click the blue folder icon in the `Image File` groupbox.

Select a location and type a name in the filename field (i.e. generic_node_image_2021-01-12.img).

Click the `Open` button and you should see the path and filename in the `Image File` textbox to the left of the blue folder.

Select the device from the `Device` dropdown box.

Click `Read`. Win32DiskImager will now read the microSD card and write the an image to the location and filename you entered in the steps above.

Step 2 - Configure Head Node

2.1 Create another microSD card using the generic node image

This will be the head or master node (*node0*).

Using the Raspberry Pi Imager software click `Choose OS`. Scroll down the list to the bottom option `Use custom`.

Navigate to the image you created and click on it. Then click `Open`.

Click `Choose SD Card` and select your SD card.

Click `Write` to write the image to the microSD card.

2.2 Login to the head node

Using puTTY login to the head node using the ip address `192.168.10.3` or the WiFi ip address from your local router.

2.3 Configure head node

2.3.1 Install NTP

This will ensure that the system time is synced and for the SLURM scheduler and Munge authentication.

```
sudo apt install ntpdate -y
```

2.3.2 Change the hostname

Change hostname for persistence (only changes after restart):

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

Change `nodeX` to `node0`.

Change the hostname for current session (avoids needing to restart until later):

```
sudo hostname node0
```

2.3.3 Change the ip address

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

Change `192.168.10.3/24` to `192.168.10.100/24`.

2.3.4 Change the hosts file

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

Change `127.0.1.1 nodeX` to `127.0.0.1 node0`.

2.4 Create shared folder

This will create a folder structure that will be shared across the nodes using NFS. This will allow all data files, compiled software libraries, and user files to be shared across the cluster from the head node.

Create hpc group:

```
sudo groupadd hpc
```

Add pi user to hpc group:

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

Create hpc directory in root:

```
sudo mkdir /hpc
```

2.5 Connect and Mount Flash Drive

2.5.1 Find the drive identifier

Plug the flash drive into on of the USB ports on the head node. To figure out its device location use the `lsblk` command.

```
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda           8:0    1 115.7G  0 disk
`-sda1        8:1    1 115.7G  0 part
mmcblk0     179:0    0   3.8G  0 disk
|-mmcblk0p1 179:1    0   256M  0 part /boot
`-mmcblk0p2 179:2    0   3.5G  0 part /
```

In this case, the main partition of the flash drive is at `/dev/sda1`.

2.5.2 Format the drive

We're first going to format the flash drive to use the `ext4` filesystem:

```
sudo mkfs.ext4 /dev/sda1
```

Note: The UUID listed when creating the filesystem will be needed for automatically mounting the drive. Take note of the UUID field in the output message similar to what is below.

```
Creating filesystem with 30326780 4k blocks and 7585792 inodes
Filesystem UUID: 512f2ef6-727d-4d54-8580-ca965da3af38
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks): done
Writing superblocks and filesystem accounting information: done
```

2.6 Setup automatic flash drive mounting

Edit `/etc/fstab` to mount the drive on boot:

```
sudo nano /etc/fstab
```

Add the following line to the end of the file:

```
UUID=512f2ef6-727d-4d54-8580-ca965da3af38 /hpc ext4 defaults 0 2
```

Finally, mount the drive:

```
sudo mount -a
```

2.7 Export the NFS share

Now we need to export the mounted drive as a network file system share so the other nodes can access it.

2.7.1 Install the NFS server

```
sudo apt install nfs-kernel-server -y
```

2.7.2 Export the NFS share

Edit `/etc/exports` and add the following line:

```
/hpc 192.168.10.0/24(rw,sync,no_root_squash,no_subtree_check)
```

Run the following command to update the NFS kernel server:

```
sudo exportfs -a
```

Take ownership of */hpc*:

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

Create hpc subdirectories:

```
cd /hpc

mkdir users data lib
```

Set permissions of */hpc* folder:

```
sudo chmod 777 -R /hpc
```

2.8 Move *pi* user to the new home directory

This account is used to move the home directory of the pi user since that users home directory can't be moved while signed in. This is account creation is temporary and will be removed at the end. Additional accounts can be added once configuration is completed to include moving of home directories to the new */hpc/users/* folder.

Create a new user account:

```
sudo adduser --home /hpc/users/tempuser tempuser
```

Set the password for the user and save it. No other settings required.

Give need group permissions:

```
sudo usermod -aG hpc tempuser
sudo usermod -aG sudo tempuser
```

Logout of the *pi* user:

```
exit
```

Login as the new ***tempuser*** account using puTTY.

Move the *pi* user home directory:

```
sudo usermod -m -d /hpc/users/pi pi
```

Close this terminal and login as ***pi*** account again.

Remove the *tempuser* account and directories:

```
sudo userdel -r tempuser
```

2.9 Install SLURM Controller Packages

```
sudo apt install slurm-wlm -y
```

2.9.1 Slurm Configuration

Change to the Slurm configuration folder and copy over the provided configuration file:

```
cd /etc/slurm-llnl

sudo cp /usr/share/doc/slurm-client/examples/slurm.conf.simple.gz .

sudo gzip -d slurm.conf.simple.gz

sudo mv slurm.conf.simple slurm.conf
```

2.9.2 Set the control machine info

Open the Slurm configuration file:

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

Set the *SlurmctldHost* line to the ip address of the head node:

```
SlurmctldHost=node0
```

2.9.3 Set the cluster name

Set the *ClusterName* field:

```
ClusterName=swosucluster
```

2.9.4 Add the nodes

Remove any lines beginning with `NodeName=` or `PartitionName=`.

Add the following to the end of the file:

```
NodeName=node0 NodeAddr=192.168.10.100 CPUs=4 State=UNKNOWN
NodeName=node1 NodeAddr=192.168.10.101 CPUs=4 State=UNKNOWN
NodeName=node2 NodeAddr=192.168.10.102 CPUs=4 State=UNKNOWN
NodeName=node3 NodeAddr=192.168.10.103 CPUs=4 State=UNKNOWN
NodeName=node4 NodeAddr=192.168.10.104 CPUs=4 State=UNKNOWN
NodeName=node5 NodeAddr=192.168.10.105 CPUs=4 State=UNKNOWN
NodeName=node6 NodeAddr=192.168.10.106 CPUs=4 State=UNKNOWN
NodeName=node7 NodeAddr=192.168.10.107 CPUs=4 State=UNKNOWN
```

2.9.5 Create a partition

```
PartitionName=swosucluster Nodes=node[0-7] Default=YES MaxTime=INFINITE State=UP
```

2.9.6 Configure cgroups support

Create */etc/slurm-llnl/cgroup.conf* file and add the following lines:

```
CgroupMountpoint="/sys/fs/cgroup"
CgroupAutomount=yes
CgroupReleaseAgentDir="/etc/slurm-llnl/cgroup"
AllowedDevicesFile="/etc/slurm-llnl/cgroup_allowed_devices_file.conf"
ConstrainCores=no
TaskAffinity=no
ConstrainRAMSpace=yes
ConstrainSwapSpace=no
ConstrainDevices=no
AllowedRamSpace=100
AllowedSwapSpace=0
MaxRAMPercent=100
MaxSwapPercent=100
MinRAMSpace=30
```

Whitelist system devices by creating the file */etc/slurm-llnl/cgroup_allowed_devices_file.conf*:

```
/dev/null
/dev/urandom
/dev/zero
/dev/sda*
/dev/cpu/*/*
/dev/pts/*
/hpc*
```

2.10 Copy the Configuration Files to Shared Storage

```
sudo cp slurm.conf cgroup.conf cgroup_allowed_devices_file.conf /hpc

sudo cp /etc/munge/munge.key /hpc
```

2.11 Enable and Start SLURM Control Service

Munge:

```
sudo systemctl enable munge

sudo systemctl start munge
```

The SLURM daemon:

```
sudo systemctl enable slurmd

sudo systemctl start slurmd
```

The control daemon:

```
sudo systemctl enable slurmctld

sudo systemctl start slurmctld
```

2.12 Reboot

```
sudo reboot
```

2.13 Install MPICH

Install prerequisite *Fortran* which wil be required for compiling MPICH. All other dependencies are already installed.

1. Install Fortran

```
sudo apt install gfortran -y
```

2. Create build and install directory inside mpich3 directory:

```
cd /hpc/lib

mkdir mpich_3.3.2

cd mpich_3.3.2

mkdir build install
```

3. Download mpich3 and unzip:

```
wget http://www.mpich.org/static/downloads/3.3.2/mpich-3.3.2.tar.gz

tar xfz mpich-3.3.2.tar.gz
```

4. Compile and install MPICH3:

```
cd build

/hpc/lib/mpich_3.3.2/mpich-3.3.2/configure --prefix=/hpc/lib/mpich_3.3.2/install

make

make install
```

5. Activate environment variable:

```
export PATH=/hpc/lib/mpich_3.3.2/install/bin:$PATH
```

6. Add path to environment variables for persistance:

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

Add the following to the end of the file:

```
# MPICH-3.3.2
export PATH="/hpc/lib/mpich_3.3.2/install/bin:$PATH"
```

Save and exit.

7. Create list of nodes for MPI:

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

Create node list:

```
cd ~
nano nodelist
```

Add the head node ip address to the list:

```
192.168.10.100
```

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

8. Test MPI

Test 1 - Hostname Test

Enter on command line:

```
cd ~

mpiexec -f nodelist hostname
```

Output:

```
node0
```

Test 2 - Calculate Pi

Enter on command line:

```
mpiexec -f nodelist -n 2 /hpc/lib/mpich_3.3.2/build/examples/cpi
```

Output:

```
Process 0 of 2 is on node0
Process 1 of 2 is on node0
pi is approximately 3.1415926544231318, Error is 0.0000000008333387
wall clock time = 0.003250
```

Step 3 - Configure Compute Node

In this step we will configure a single compute node using the previously generated generic image. With a new microSD card write the "generic image" using Raspberry Pi Imager and the "Custom Image" option.

3.1 Connect and configure the generic compute node image

Using puTTY connect to the head node. Using SSH connect from the head node to the generic image node running on a compute node.

```
ssh pi@nodeX
```

Yes to accept the key.

Use the default `raspberry` password.

3.2 Install NFS client

```
sudo apt install nfs-common -y
```

3.2.1 Add hpc group and assign *pi* user to group

```
sudo groupadd hpc

sudo usermod -aG hpc pi
```

3.2.2 Create mount folder

```
sudo mkdir /hpc

sudo chown -R pi:hpc /hpc

sudo chmod -R 777 /hpc
```

3.2.3 Setup automatic mounting

Edit the `/etc/fstab` file:

```
sudo nano /etc/fstab
```

Add the following to the end of the file:

```
192.168.10.100:/hpc     /hpc   nfs   defaults 0 0
```

Now mount the share:

```
sudo mount -a
```

3.3 Install SLURM client

```
sudo apt install slurmd slurm-client -y
```

3.3.1 Copy the configuration files

```
sudo cp /hpc/munge.key /etc/munge/munge.key
sudo cp /hpc/slurm.conf /etc/slurm-llnl/slurm.conf
sudo cp /hpc/cgroup* /etc/slurm-llnl
```

3.4 Configure Munge

3.4.1 Enable and start Munge

```
sudo systemctl enable munge

sudo systemctl start munge
```

3.4.2 Test Munge

This is run on the client/compute node:

```
ssh pi@nodeX munge -n | unmunge
```

3.5 Start the SLURM Daemon

The start command will through an error. This will be corrected once the node is renamed to its final hostname once deployed.

```
sudo systemctl enable slurmd

sudo systemctl start slurmd
```

3.6 Shutdown the node

```
sudo shutdown -h now
```

Step 4 - Generate Generic Compute Node Image

Use Win32DiskImager to Read the image to a file. Name the file something similar to "generic_compute_node_2021_01_17.img". Click the `Read` button and wait for the program to read the microSD card and write the image file.

Step 5 - Deploy Compute Nodes

In this step we will configure a single compute node using the previously generated generic image. With a new microSD card write the "generic image" using Raspberry Pi Imager and the "Custom Image" option.

5.1 Connect and configure the generic compute node image

Using puTTY connect to the head node. Using SSH connect from the head node to the generic image node running on a compute node.

Note: On the below instructions `node1` will increment with the deployment of each new node. As will the ip address 192.168.10.**101**.

```
ssh pi@nodeX
```

Yes to accept the key if asked.

Use the default `raspberry` password.

5.1.1 Change the hostname

Edit `/etc/hostname` file:

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

Change `nodeX` to `node1`.

5.1.2 Change the ip address

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

Change `192.168.10.3/24` to `192.168.10.101/24`.

5.1.3 Change the hosts file

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

Change `127.0.1.1 nodeX` to `127.0.0.1 node1`.

5.1.4 Restart for changes to take effect

```
sudo reboot
```

5.2 Generate SSH key on head node

```
ssh-keygen
```

Press `Enter` on each selection.

5.3 Copy the SSH key to each node:

```
ssh-copy-id pi@node1
```

Enter the default `raspberry` password when prompted.

Replace `node1` with each nodes number and repeat the process to distribute the key to the entire cluster.

Section 6 - Troubleshooting

6.1 Node not in idle state:

Refresh the state of the node by running the below command with the desired nodename. Nodename node1 is used for the exampel below:

```
sudo scontrol update nodename=node1 state=resume
```

Section 7 - References

Much of this guide was combined from my own configuration but also from Garrett Mills guide listed below.

- Building a Raspberry Pi Cluster: Part 1 - The Basics
- Building a Raspberry Pi Cluster: Part 2 - Some Simple Jobs
- Building a Raspberry Pi Cluster: Part 3 - OpenMPI, Python, and Parallel Jobs