

Raspberry Pi Cluster Setup

1. Setup geographic info
 - a. `$ sudo raspi-config`
 - b. "5. Localization Options"
 - i. "L1. Locale"
 1. Deselect "en_GB.UTF-8 UTF-8"
 2. Select "en_US.UTF-8 UTF-8"
 3. Select "en_US.UTF-8" as default
 - ii. "L2. Timezone"
 1. "US"
 2. "Central"
 - iii. "L3. Keyboard"
 1. Select "HP Pavilion ZT1100" or "Generic 104-key PC"
 2. Select "Other"
 3. Select "English (US)"
 4. Select "English (US)"
 5. Select "No AltGr key"
 6. Select "No compose key"
2. Setup Wi-Fi connection
 - a. `$ sudo raspi-config`
 - b. "1. System Options"
 - c. "S1 Wireless LAN"
 - d. Select "US United States"
 - e. Enter Wi-Fi information
 - f. Exit raspi-config
3. Change hostname
 - a. `$ sudo raspi-config`
 - b. "1. System Options"
 - c. "S4. Hostname"
 - d. `clusterphuq<node #>`
4. Change default password for pi account
 - a. `$ sudo raspi-config`
 - b. "1. System Options"
 - c. "S3. Password"
 - d. Enter new password
5. Increase amount of RAM available
 - a. `$ sudo raspi-config`
 - b. "4. Performance Options"
 - c. "P2. GPU Memory"
 - d. Set to either 16 or 32 (probably 16)

6. Enable SSH
 - a. `$ sudo raspi-config`
 - b. "3. Interface Options"
 - c. "P2. SSH"
 - d. Select "Enable" or "Yes"
7. Setup Ethernet connection
 - a. `$ sudo su`
 - b. `# echo "interface eth0" >> /etc/dhcpd.conf`
 - c. `# echo "static ip_address=10.10.10.<node #>/16" >> /etc/dhcpd.conf`
- ~~8. Setup VNC (master node only) → optional~~
 - ~~a. `$ sudo su`~~
 - ~~b. `# bash vnc_setup.sh`~~
 - ~~c. `# exit`~~
 - ~~d. `$ source ~/.bash_aliases`~~
 - ~~e. `$ start-vnc`~~
 - ~~f. `$ bash vnc_config.sh`~~
9. Reboot the Raspberry Pi
 - a. Test SSH connectivity
 - b. Test Wi-Fi connectivity
10. Repeat steps 1 through 9 for each node in the cluster
11. On main computer (the one you'll be managing the cluster from), install ClusterSSH
 - a. `$ sudo apt install clusterssh`
12. Run ClusterSSH to create the config file
 - a. `$ cssh`
 - b. filepath: `~/.clusterssh/config`
13. Change the options in the config file
 - a. `"#comms=" → "comms=ssh"`
 - i. `$ sed -i "s;#comms=;comms=ssh;" ~/.clusterssh/config`
 - b. `"#user=" → "user=pi"`
 - i. `$ sed -i "s;#user=;user=pi;" ~/.clusterssh/config`
14. Add the following options in the config file
 - a. `"clusters = default"`
 - b. `"default = <IP address 1> <IP address 2> ..."`

15. Update and install MPI across all nodes using ClusterSSH
 - a. `$ cssh`
 - i. Note: This will open all IP addresses automatically under default, no need to specify IP addresses.
 - ii. `$ sudo apt update`
 - iii. `$ sudo apt upgrade`
 - iv. `$ sudo apt install openmpi-bin openmpi-common libopenmpi3 libopenmpi-dev`
16. Generate an SSH key on all nodes
 - a. `$ cssh`
 - i. `$ ssh-keygen -t rsa`
 - ii. Press "Enter" for the next 3 questions
17. On each slave node, copy the node's SSH key to master node
 - a. `$ cssh`
 - i. `$ ssh-copy-id <master node's IP address>`
18. On the master node, copy it's SSH key to all slave nodes
 - a. `$ cssh`
 - i. `$ ssh-copy-id <slave node 1's IP address>`
 - ii. `$ ssh-copy-id <slave node 2's IP address>`
 - iii. ...
19. Create aliases for different numbers of nodes on master node
 - a. `$ echo "alias mpiexec-n2='mpiexec -n 2 -host 10.10.10.0,10.10.10.1'" >> ~/.bash_aliases`
 - b. `$ echo "alias mpiexec-n3='mpiexec -n 3 -host 10.10.10.0,10.10.10.1,10.10.10.2'" >> ~/.bash_aliases`
 - c. ...
20. Setup network file system
 - a. Format external drive to ext4 file system
 - i. Option 1 → partition manager program (GUI)
 - ii. Option 2 → `$ sudo mkfs.ext4 /dev/sd<letter><#>`
 - b. On all nodes
 - i. Create mount directory for network file system
 1. `$ sudo mkdir <mount directory>`
 2. `$ sudo chown pi.users -R <mount directory>`
 3. `$ sudo chmod 777 -R <mount directory>`
 - ii. Create environment variable for mount directory
 1. `$ sudo nano /etc/environment`
 2. Add the following line:
 1. `NFS=<mount directory>`
 3. `$ source /etc/environment`
 4. Mount directory can now be used using `$NFS` as shorthand
 - c. On master node
 - i. Plug in external drive
 - ii. Find /dev location
 1. `$ lsblk`

- iii. Find external drive's UUID
 - 1. \$ blkid
- iv. Prep master node for automounting external drive
 - 1. \$ sudo nano /etc/fstab
 - 2. Add the following line:
 - 1. UUID=<UUID> <mount directory> ext4 defaults 0 2
- v. Mount external drive
 - 1. \$ sudo mount -a
- vi. Install NFS server
 - 1. \$ sudo apt install nfs-kernel-server
- vii. Export NFS share
 - 1. \$ sudo nano /etc/exports
 - 2. Add the following line:
 - 1. <mount directory> <master node's IP address>(rw, sync, no_root_squash, no_subtree_check)
- viii. Update NFS kernel server
 - 1. \$ sudo exportfs -a
- d. On all slave nodes
 - i. Install NFS client
 - 1. \$ sudo apt install nfs-common
 - ii. Enable automatic mounting
 - 1. \$ sudo nano /etc/fstab
 - 2. Add the following line
 - 1. <master node's IP address>:<mount directory> <mount directory> nfs defaults 0 0
 - 3. \$ sudo mount -a

21. Setup SLURM

- a. On master node
 - i. Switch to superuser
 - 1. \$ sudo su
 - ii. Add hostnames and their IP addresses to /etc/hosts
 - 1. # nano /etc/hosts
 - 1. Add lines according to the following pattern:
 - 1. <IP address of slave node 1> <hostname of slave node 1>
 - 2. <IP address of slave node 2> <hostname of slave node 2>
 - 3. ...
 - iii. Install SLURM controller packages
 - 1. # apt install slurm-wlm
 - iv. SLURM configuration
 - 1. # cd /etc/slurm-llnl
 - 2. # cp /usr/share/doc/slurm-client/examples/slurm.conf.simple.gz .
 - 3. # gzip -d slurm.conf.simple.gz
 - 4. # mv slurm.conf.simple slurm.conf
 - 5. # nano /etc/slurm-llnl/slurm.conf
 - 1. Edit the following lines:

1. SlurmctldHost=<master node hostname>(<master node IP address>)
2. SelectType=select/cons_res
3. SelectTypeParameters=CR_Core
4. ClusterName=<cluster name (missing node number)>
5. For adding compute nodes
 1. Nodename=<master node hostname> NodeAddr=<master node IP address> CPUs=4 State=UNKNOWN
 2. NodeName=<slave node 1 hostname> NodeAddr=<slave node 1 IP address> CPUs=4 State=UNKNOWN
 3. NodeName=<slave node 2 hostname> NodeAddr=<slave node 2 IP address> CPUs=4 State=UNKNOWN
 4. ...
6. PartitionName=<arbitrary name> Nodes=<cluster name>[1-<last slave node number>] Default=YES MaxTime=INFINITE State=UP
6. cgroups support
 1. # touch cgroup.conf
 2. # nano cgroup.conf
 1. Add the following lines:
 1. CgroupMountpoint="/sys/fs/cgroup"
 2. CgroupAutomount=yes
 3. CgroupReleaseAgentDir="/etc/slurm-llnl/cgroup"
 4. AllowedDevicesFile="/etc/slurm-llnl/cgroup_allowed_devices_file.conf"
 5. ConstrainCores=no
 6. TaskAffinity=no
 7. ConstrainRAMSpace=yes
 8. ConstrainSwapSpace=no
 9. ConstrainDevices=no
 10. AllowedRamSpace=100
 11. AllowedSwapSpace=0
 12. MaxRAMPercent=100
 13. MaxSwapPercent=100
 14. MinRAMSpace=30
7. Whitelist system devices
 1. # touch cgroup_allowed_devices_file.conf
 2. # nano cgroup_allowed_devices_file.conf
 1. Add the following lines:
 1. /dev/null
 2. /dev/urandom
 3. /dev/zero
 4. /dev/sda*
 5. /dev/cpu/*/*
 6. /dev/pts/*
 7. <NFS mount dir>*
8. # cp slurm.conf cgroup.conf
cgroup_allowed_devices_file.conf <NFS mount dir>
9. # cp /etc/munge/munge.key <NFS mount dir>
10. Start Munge

```

    1. # systemctl enable munge
    2. # systemctl start munge
11.    Start SLURM
    1. # systemctl enable slurmd
    2. # systemctl start slurmd
    3. # systemctl enable slurmctld
    4. # systemctl start slurmctld
v. Reboot
    1. # reboot
b. On each slave node
i. $ sudo su
ii.    # apt install slurmd slurm-client
iii.   Add hostnames and their IP addresses for all other
        nodes to /etc/hosts
    1. Note: Don't add an entry for the node itself
    2. # nano /etc/hosts
        1. Add lines according to the following pattern:
            1. <IP address of master node> <hostname of master
               node>
            2. <IP address of slave node> <hostname of slave node>
            3. <IP address of slave node> <hostname of slave node>
            4. ...
iv.    Copy master node's configuration to each slave node
    1. # cp <NFS mount dir>/munge.key /etc/munge/munge.key
    2. # cp <NFS mount dir>/slurm.conf
        /etc/slurm-llnl/slurm.conf
    3. # cp <NFS mount dir>/cgroup* /etc/slurm-llnl
v. Enable, start, and test Munge
    1. # systemctl enable munge
    2. # systemctl start munge
    3. # exit
    4. $ ssh pi@node01 munge -n | unmunge
    5. If you see an error message...
        1. Double-check that munge.key is identical across all
           nodes
        2. Reboot all nodes
        3. Try step 4 again
        4. If there's still an error, try replicating munge.key
           across all nodes and retry again
vi.    Enable, start, and test SLURM
    1. $ sudo su
    2. # systemctl enable slurmd
    3. # systemctl start slurmd
    4. # exit
    5. $ sinfo
    6. $ srun --nodes=<#> hostname
    7. If you see another error message, rebooting should fix
        this

```