

Raspberry Pi Cluster Setup

1. Install OS onto microSD card
 - a. Master node options
 - i. Raspbian
 - ii. Raspbian Lite
 - b. Slave nodes → Raspbian Lite
 - c. Raspberry Pi Imager makes this easy
2. 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"
3. 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`
4. Change hostname
 - a. `$ sudo raspi-config`
 - b. "1. System Options"
 - c. "S4. Hostname"
 - d. `<node name><node #>`
 - i. Ex: node0, node1,...
5. Change default password for pi account
 - a. `$ sudo raspi-config`
 - b. "1. System Options"
 - c. "S3. Password"
 - d. Enter new password
6. 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)

7. Enable SSH
 - a. `$ sudo raspi-config`
 - b. "3. Interface Options"
 - c. "P2. SSH"
 - d. Select "Enable" or "Yes"
8. Add "rasberry_pi_cluster" repo
 - a. `$ git clone https://github.com/TheOGChips/rasberry_pi_cluster.git ~/rasberry_pi_cluster`
9. Setup Ethernet connection
 - a. `$ cd rasberry_pi_cluster`
 - b. `$ sudo su`
 - c. `# bash eth_static_ip_setup.sh <node #>`
10. Setup VNC (master node only)
 - a. `$ cd rasberry_pi_cluster`
 - b. `$ sudo su`
 - c. `# bash vnc_setup.sh`
 - d. `# exit`
 - e. `$ source ~/.bash_aliases`
 - f. `$ start-vnc`
 - g. `$ bash vnc_config.sh`
 - h. Note: Optional. Only useful if Raspbian (not Raspbian Lite) image is running on master node.
11. Reboot the Raspberry Pi
 - a. Test SSH connectivity
 - i. `$ ssh pi@<IP address created in step 9>`
 - b. Test Wi-Fi connectivity
 - i. `$ sudo apt update`
 1. Note: if no errors → working
12. Repeat steps 1 through 11 for each node in the cluster
13. On main computer (the one you'll be managing and accessing the cluster from), install and configure ClusterSSH
 - a. `$ bash clusterssh_setup.sh`
14. Using ClusterSSH: install OpenMPI and mpi4py, create aliases for their commands, create and mount an NFS (network file system), and setup trusted SSH communication amongst all nodes in the cluster
 - a. `$ cssh`
 - i. Note: This will open all IP addresses automatically under default (no need to specify IP addresses). There will be one terminal window for each Pi.
 - ii. On master node
 1. `$ bash comm_setup.sh master`
 - iii. On slave nodes
 1. `$ bash comm_setup.sh slave`

15. Setup SLURM
 - a. On master node
 - i. `$ bash slurm_setup/slurm_setup.sh master`
 - ii. Reboot
 1. `# reboot`
 - b. On each slave node
 - i. `$ bash slurm_setup/slurm_setup.sh slave`
 - ii. Testing Munge
 1. If you see an error message...
 1. Double-check that `munge.key` is identical across all nodes
 2. Reboot all nodes
 3. Try `"$ ssh pi@node01 munge -n | unmunge"` again
 4. If there's still an error, try replicating `munge.key` across all nodes and retry again
 - iii. Testing SLURM
 1. If you see another error message...
 1. Reboot all Pis
 2. Try the following again:
 1. `sinfo`
 2. `srun --nodes=<# slave nodes> hostname`- 16. Sources:
 - a. Part 1: <https://glmdev.medium.com/building-a-raspberry-pi-cluster-784f0df9afbd>
 - b. Part 2: <https://glmdev.medium.com/building-a-raspberry-pi-cluster-aaa8d1f3d2ca>
 - c. Part 3: <https://glmdev.medium.com/building-a-raspberry-pi-cluster-f5f2446702e8>