### Installing Operating System Image:

If you bought the microSD cards with the NOOB image installed on it, then you can skip these instructions. These instructions are for if you do not have an operating system installed on your microSD card or if you encounter a problem during any of the following steps and need to restart. A good resource that may help you with majority of the steps can be found at: <https://www.raspberrypi.org/documentation/installation/installing-images/README.md>.

For the following instructions, it is created for the image Raspbian Jessie.

Optional helpful software for Mac users: <http://ivanx.com/raspberrypi/>

Once the tool has been downloaded, unzip the folder and the tool will be ready to run. When installing the image onto the SD card, you can utilize this tool which will give you an estimate of the remaining time till the card is completed. It provides step by step aid on what to do throughout the process.

After the image has been installed onto the microSD cards and the raspberries have been booted up, you need to configure the settings. So boot up one raspberry pi and this one will be the master node which has the WiFi adapter.

1. The first step is to change the password from the default on, so type **passwd** into the terminal and enter the password of your choice. By default, the raspberry password is raspberry.
2. Afterwards, enter **sudo raspi-config** so that you can expand the file system which means you can fully utilize the entire size of the microSD card.
   1. Navigate to Expand File System and click [ENTER].
   2. After the expansion is complete, it will return an [OK].
3. You will also want to set the keyboard, time configuration, and wifi.
   1. Go to Menu -> Preferences -> Raspberry Pi Configuration-> Localisation to set the appropriate Locale, Timezone and the Keyboard and then click [OK].
4. Afterwards, you can go ahead and reboot.
5. Once the raspberry has been rebooted, you should connect to the WiFi if you are not connected yet.
6. Since the standard downloaded image doesn’t include the updated packages and releases, it would be best to update the system. In terminal, you can run **sudo apt-get update**. If you want to upgrade to the most updated system, you can run **sudo apt-get upgrade**. WARNING: The upgrade can take on average approximately 40 minutes to complete.
7. Just like how the image does not contain all the updated packages and releases, they do not possess all the libraries that are available to the raspberry pi. Therefore, you will need to install several libraries that will be needed later on. Enter the following command into the terminal: **sudo apt-get install emacs openmpi-bin libopenmpi-dev python-mpi4py vim -y**Note: The -y flag indicates that you agree to any of the warnings that the system may indicate such as how much space these library installs will require.
8. In order to enable easy transitions between several raspberry pis, you need to enable ssh without passwords. Therefore, you need to create a ssh key using the command **ssh-keygen** in the terminal.
   1. [ENTER] for “Enter the file in which to save the key” which tells the system where to place this key in. It is fine to utilize the default folder directory.
   2. [ENTER] for the “Enter passphrase (empty for no passphrase)” which tells the system that you do not want to set a password that will need to be used to access this key.
   3. [ENTER] again to confirm that you do not want to enter a passphrase.
   4. Afterwards, the system will generate the key and return the key’s randomart image which indicates that they key was successfully created.
9. Setting up the static IP address for the cluster
   1. In order to set a static IP address, you need to edit one of the networking files. Enter the following commands into the terminal:  
      **sudo nano /etc/network/interfaces**
      1. This will be open a file that needs to be edited. Change ***iface eth0 inet dhcp*** to ***iface eth0 inet static*** which indicates that you want this raspberry to use a static IP address.
      2. Following that line, you need to specify some information about the static IP address you want the raspberry to use. Therefore, you want to append the following information underneath the iface eth0 inet static line  
          ***address 10.0.0.10  
          netmask 255.255.255.0  
          network 10.0.0.0  
          broadcast 10.0.0.255***Notes:*address* – The address you want to give your Pi, this can be any IP in the network range, but it's usually advisable to go higher rather than lower, or you could end up logging different devices to the same IP! I've selected 192.168.1.81, as we're already registered to that address (denoted by 'inet addr'), but this can be any IP address from the range 192.168.1.1 to 192.168.1.255.  
         *netmask* – [NEED TO ADD INFORMATION]  
         *network* – The router IP address, this is the 'Destination' Address was found earlier. You can also grab this off your router, it will say on the side somewhere.  
         *broadcast* – [NEED TO ADD INFORMATION]
      3. Now that the information is in the file, you want to save the file before exiting. [CTRL] + X will exit the file and then enter yes to save the changes.
   2. Now that you have set the static IP configuration, you want to remove any existing DHCP leases. In order to do that, use the following command in the terminal: **sudo rm /var/lib/dhcp/\***
   3. After you have removed the existing leases, you can reboot the system.
   4. Once the system is rebooted, you should check to see if your changes have affected the settings. If your changes were successful, you should see that the inet addr, Bcast, and the Mask in the eth0 to be the values that you have set.
10. Since this first raspberry pi will need to be the access point for all the other raspberry pis to access the internet and cat videos, you have to enable the port forwarding so that the other raspberries can utilize the first raspberry pi as a gateway to the Internet.
    1. You need to edit the system control file. Enter the following commands into the terminal:  
       **sudo nano /etc/sysctl.conf**
    2. You will need to find the line that states **net.ipv4.ip\_forward=1** and uncomment it (remove the ‘#’ sign from the beginning of the line). Afterwards, you want to save the file before exiting. [CTRL] + X will exit the file and then enter yes to save the changes.
    3. In order to check that the appropriate changes have been made, you want to run the command **sudo sysctl -p** which should return the line that you uncommented. It should return **net.ipv4.ip\_forward=1**.
    4. If only that line is returned, then you have turned on port forwarding correctly. Now you want to restart your network.
       1. Restart your network with the following command:   
          **sudo /etc/init.d/networking restart**
11. Now in order to have some examples that you can use as a starting block, you want to clone the Github repository. You can clone the repository using the following command:  
    **git clone** [**https://github.com/sdsc/sandbox-cluster-guide.git**](https://github.com/sdsc/sandbox-cluster-guide.git)
    1. Now you want to utilize one of the files in the repository to make sure your IP tables are set correctly.
       1. Change into the sandbox-cluster-guide directory by   
          **cd sandbox-cluster-guide**
       2. Afterwards, you want to checkout the beta-workshop branch by **git checkout beta-workshop**
       3. Change into the config/pinode-0/network directory by   
          **cd config/pinode-0/network**
       4. Afterwards, you want to restore your IP tables using the following command:  
          **sudo iptables-restore < ./iptables**
12. In order to keep everything in sync throughout all the pis, you want to set a home directory that will be shared amongst all the raspberries. Therefore, you will need an NFS server.
    1. First, you will need to install the libraries that will be needed on the first node. Install the libraries using the following command:  
       **sudo apt-get install nfs-kernel-server rpcbind isc-dhcp-server -y**
    2. Afterwards, you need to reorganize several directories so that the system knows which ones to prioritize first. You can do this will the following for loop command:   
       **for i in 2 3 4 5  
       do  
        sudo mv /etc/rc$i.d/S01nfs-kernel-server /etc/rc$i.d/S02nfs-kernel-server  
        sudo ln -s /etc/init.d/rpcbind /etc/rc$i.d/S01rpcbind  
       done**
    3. Afterwards, you will need to tell the NFS server which directory you want to share. This is done by editing the /etc/exports file.
       1. To edit the exports file, you can use **sudo nano /etc/exports**
       2. Add the following line to the end of the file:   
          **/home 10.0.0.0/23(rw,sync)**  
          Where 10.0.0.0 is the static IP address that you entered for this first raspberry pi and it specifies that it should have both read and write enabled and to keep the /home directory synchronized.
       3. Afterwards, you want to save the file before exiting. [CTRL] + X will exit the file and then enter yes to save the changes.
       4. Now you would want to restart the raspberry pi so that the settings are enabled correctly.
13. Now that the settings for one of the raspberry pis have been set up, you can now boot up another raspberry pi and repeat steps 1 through 9.
    1. For step 9, you want to provide it with a unique static IP address so it will need to be different from the one you set for the first raspberry pi. You should enter the following information into the /etc/network/interfaces file:  
        **iface etho inet static  
        address 10.0.0.11  
        netmask 255.255.255.0  
        network 10.0.0.0  
        broadcast 10.0.0.255  
        gateway 10.0.0.10**Note:  
       *gateway* – [NEED TO ADD INFORMATION] You want to set the gateway to the static IP address of the first raspberry pi.  
       You want to assign the static IP address to be something that you will be able to remember, so it is helpful to assign a consecutive static IP address.
    2. Since the first raspberry pi was set up with a NFS server, the others have be told that they want to mount this directory like an external device. In order to set up automatic mounting, you will need to making the following changes:
       1. You need to edit the fstab file by entering **sudo nano /etc/fstab**
       2. Enter the following information to the bottom of the file:

**10.0.0.10:/home /home nfs defaults,noatime,noauto,x-systemd.automount 0 0**

where the first section specifies which device’s drive to mount. Then it specifies where it is to be mounted to and how, which is by using NFS. Afterwards, it tells it to do an automatic mount at each bootup.

* + 1. Afterwards, you want to save the file before exiting. [CTRL] + X will exit the file and then enter yes to save the changes.
    2. Now you would want to restart the raspberry pi so that the settings are enabled correctly.