BUILDING A BEOWULF CLUSTER

Terminologies

- ***Node:*** A node refers to a virtual machine (VM) within the cluster setup that typically operates without a graphical user interface (GUI). Nodes are primarily managed via the command line interface (CLI). In this setup, we'll be installing Ubuntu Live Server on each node to provide essential services and functionality.
- ***Master Node:*** The master node is a virtual machine within the cluster setup that includes a graphical user interface (GUI). It serves as the central control point for managing and orchestrating the activities of all other nodes in the cluster. In this setup, we'll install Ubuntu Desktop version on the master node to provide a user-friendly interface for administrative tasks.
- ***NAT (Network Address Translation):*** NAT is a networking technique used to modify network address information in packet headers while in transit across a routing device. It allows multiple devices within a private network to share a single public IP address. In this setup, NAT will be configured to enable internet access for all nodes while providing network isolation and security.
- ***enp0s3 and enp0s8:*** enp0s3 and enp0s8 are network interface names assigned to Ethernet-based connections in Linux systems. These names follow the modern predictable network interface naming convention. Specifically:
    - **en**: Indicates Ethernet, indicating the interface is for Ethernet-based connections.
    - **p0**: Indicates the physical location of the device, with "p0" indicating it's located on the motherboard.
    - **s3, s8**: Represents the slot number where the network interface is located. For example, "s3" might mean slot 3, and "s8" might mean slot 8.

Cluster Setup Overview

In the cluster setup, we aim to achieve the following objectives:

- All nodes have internet access, facilitated through NAT configuration.
- All nodes can communicate with each other, enabling seamless interaction and data exchange.
- The master node serves as the entry point for cluster management and administration, providing a centralized interface for controlling and monitoring cluster activities.

1) SETTING UP THE ENVIRONMENT

VIRTUALBOX INSTALLATION

Download VirtualBox:

- Visit the official VirtualBox website at https://www.virtualbox.org and download the appropriate version for your operating system.
- Ensure that you download VirtualBox version 7.0.14, as this version is compatible with the requirements of the virtual cluster setup.

Install VirtualBox:

- Follow the installation instructions provided by the VirtualBox installer.
- Once installed, ensure that VirtualBox is running correctly by launching it from your applications menu or desktop shortcut.

MASTERNODE INSTALLATION

Download Masternode Desktop

- Obtain the Masternode Desktop version (version 22.04.1) installation package from the official source.
- You can download it from https://ubuntu.com/download/desktop

Install Masternode Desktop in VirtualBox

- Open VirtualBox and create a new virtual machine.
- Follow the VirtualBox wizard to set up the virtual machine, ensuring to allocate appropriate resources such as CPU, RAM, and disk space.
- During the virtual machine creation process, select the option to install the operating system later.
- Once the virtual machine is created, start it and mount the Masternode Desktop installation ISO.
- Follow the installation instructions provided by the Masternode Desktop installer to complete the installation process.

NODES (node1, node2, node3) INSTALLATION

Download Desktop version

- Obtain the Masternode Desktop version (version 22.04.1) installation package from the official source.
- You can download it from https://ubuntu.com/download/desktop

Create Virtual Machines for Nodes:

- Open VirtualBox and create a new virtual machine for each node (node1, node2, node3).
- Follow the VirtualBox wizard to set up each virtual machine, allocating appropriate resources as needed.
- During the virtual machine creation process, select the option to install the operating system later.

Install Ubuntu Server in Virtual Machines:

- Start each virtual machine and mount the Ubuntu Server ISO.
- Follow the installation instructions provided by the Ubuntu Server installer for each virtual machine.
- Ensure to configure networking settings, hostname, and user accounts according to your requirements during the installation process.

POST-INSTALLATION TASKS

Additional Software Installation:

- Install any additional software or packages required for your specific use case on each node, such as MPI libraries.
- Follow the respective installation instructions provided by the software vendors or communities.

START THE VIRTUAL MACHINE

- Once the virtual machine is created, start it from the virtualization software.
- Open Terminal
- Update Package Lists: Run the following command to update the package lists from the repositories:

- **sudo apt update**

  This command will fetch the latest information about available packages from the Ubuntu repositories.
  - Enter Password
  - Wait for Update to Complete
  - Proceed with Installation
  - Optional, Upgrade Installed Packages: After updating the package lists, you can also upgrade any installed packages to their latest versions using the following command:
- **sudo apt upgrade**

CONFIGURATION OF NETWORK INTERFACES FOR THE VIRTUAL MACHINES

- ***Create NAT Adapter for Internet Connectivity (enp0s3):***

  Open VirtualBox and navigate to File => Tools => Network Manager => NAT Networks => Create.

  Name the NAT interface 'local-cluster-nat.' This interface will connect directly to the internet, enabling internet access for all virtual machine's NAT interfaces (enp0s3).
- ***Connect Virtual Machines to Internet NAT Interface:***

  For each virtual machine, navigate to Settings => Network.

  Enable Adapter 2 and set it to be Attached to NAT Network.

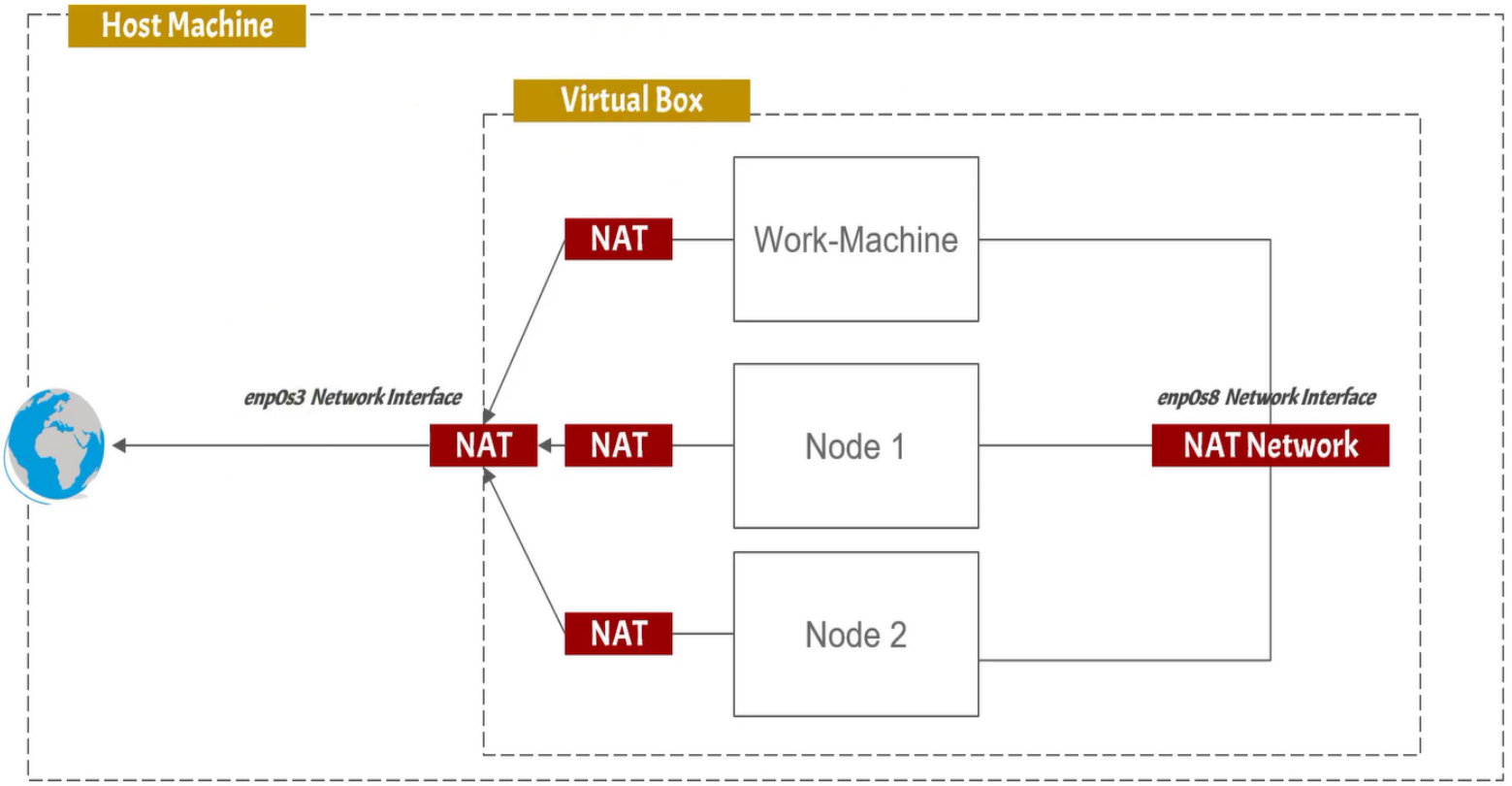
  Choose the NAT Network named 'local-cluster-nat' from the dropdown menu.

  This configuration ensures that all virtual machines can access the internet via the NAT adapter.

CONFIGURE NAT NETWORK FOR INTERNAL COMMUNICATION (enp0s8):

VirtualBox provides a NAT Network interface (enp0s8) for internal communication between virtual machines. Modify the enp0s8 interface on each virtual machine to assign static IP addresses and disable DHCP. This ensures consistent network settings across the cluster.

Disable DHCP on NAT Network Interface: DHCP (Dynamic Host Configuration Protocol) assigns IP addresses dynamically. Since we want static IP addresses, DHCP will be disabled on the NAT network interface (enp0s8). This ensures that each virtual machine retains its assigned static IP address within the cluster.



2) SETTING STATIC IP ADDRESSES

Configure each virtual machine with a static IP address within the same subnet to facilitate communication between them. Ensure that each virtual machine has a unique static IP address within the network to avoid conflicts. By following these steps, you'll establish connectivity between the virtual machines and configure network interfaces for both internet access and internal communication within the cluster**.**

To assign static IP addresses to the enp0s8 interface on each virtual machine, follow these steps:

- ***Create a New Interface Configuration File****:* Open a terminal and use a text editor like Vim to create a new interface configuration file:
    - **sudo vim /etc/network/interfaces**

  Add the following configuration for the enp0s8 interface:

  auto enp0s8

  iface enp0s8 inet static

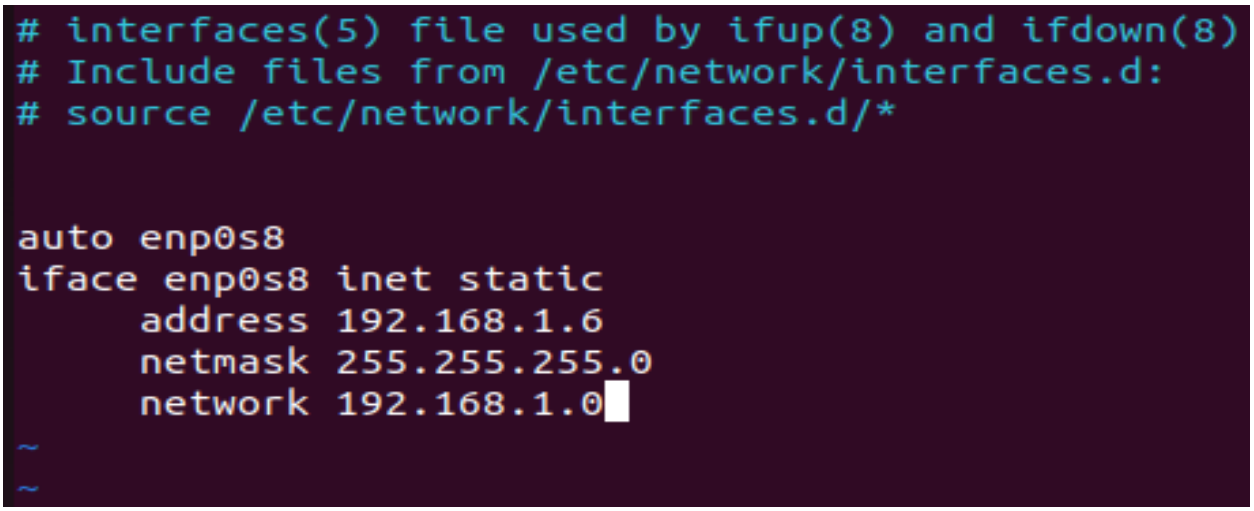
  address 192.168.1.xxx

  netmask 255.255.255.0

  network 192.168.1.0

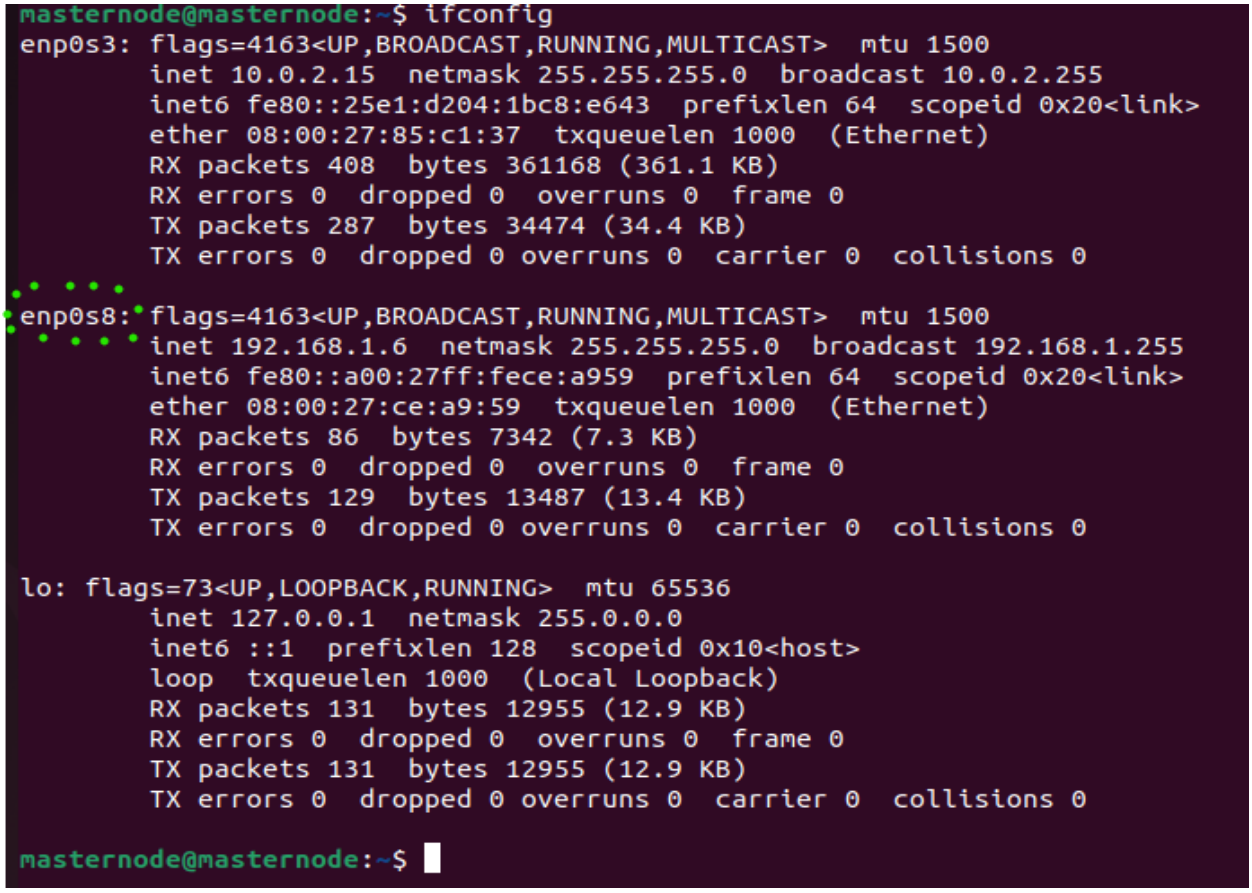
  broadcast 192.168.1.255

  Replace ‘xxx’ with the desired static ip address, save and exit the text editor.



- ***Install Network Tools****:* If the net-tools package is not already installed, you can install it using the following command:
    - **sudo apt-get install net-tools**
- ***Apply the Configuration:*** Install the ifupdown package to manage network interfaces and apply the new configuration:
    - **sudo apt install ifupdown**
    - **sudo ifup enp0s8**
- ***Confirm Configuration****:* Use the ***ifconfig*** command to confirm that the static IP address has been assigned to the enp0s8 interface:
    - **ifconfig**

    This command will display network interface information, including the assigned IP address for enp0s8.



By following these steps, you'll assign static IP addresses to the enp0s8 interface on each virtual machine, facilitating communication within the cluster.

2.1) SETTING HOSTNAMES

To set the hostname and configure the '**/etc/hosts'** file for each virtual machine in the cluster, follow these steps:

- ***Set Hostname**:* Open a terminal and use the **hostnamectl** command to set the hostname of the virtual machine:
    - **sudo hostnamectl set-hostname 'name_of_the_VM'**
      Replace 'name of the VM' with the desired hostname for the virtual machine.

```
masternode@masternode:~$ sudo hostnamectl set-hostname masternode
[sudo] password for masternode:
```

- ***Edit '/etc/hosts' File**:* Open the **/etc/hosts** file in a text editor:
    - **sudo vim /etc/hosts**

  Add an entry for the IP address and hostname of the virtual machine:
    - **192.168.1.xxx 'name of the VM'**

  Replace '192.168.1.xxx' with the static IP address assigned to the virtual machine, and 'name of the VM' with the hostname set in the previous step.

```
127.0.0.1       localhost
127.0.1.1       masternode-VirtualBox
192.168.1.6     masternode

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
~
```

- ***Verify Hostname**:* Use the hostname command to verify that the hostname has been set correctly:
    - **hostname**

```
masternode@masternode:~$ hostname
masternode
masternode@masternode:~$
```

MAKE MASTERNODE THE ENTRY POINT

- Open the **/etc/hosts** file in a text editor on the masternode:
    - **sudo vim /etc/hosts**
- Add entries for the IP addresses and hostnames of the master node and other nodes in the cluster:
    - **192.168.1.xxx 'name_of_node1'**
      192.168.1.yyy 'name_of_node2'
      192.168.1.zzz 'name_of_node3'
- Replace '192.168.1.xxx,' '192.168.1.yyy,' and '192.168.1.zzz' with the static IP addresses assigned to each node, and 'name_of_node1,' 'name_of_node2,' and 'name_of_node3' with the corresponding hostnames.

```
127.0.0.1       localhost
127.0.1.1       masternode-VirtualBox
192.168.1.6     masternode
192.168.1.7     node1
192.168.1.8     node2
192.168.1.9     node3

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

- Upon making the masternode the entry point, repeat step '4.1' on the other nodes (i.e. node1, node2, node3). This means adding masternode and the remaining nodes to the host file.

CHECK CONNECTIVITY

- Use the **ping** command to verify if nodes can connect to each other:
    - **ping 'name of node'**

  Replace 'name of node' with the hostname of the node you want to check connectivity to.

By following these steps, you'll set the hostname and configure the **/etc/hosts** file for each virtual machine in the cluster, ensuring proper connectivity and communication between nodes.

```
masternode@masternode:~$ ping node1
PING node1 (192.168.1.7) 56(84) bytes of data.
64 bytes from node1 (192.168.1.7): icmp_seq=1 ttl=64 time=1.93 ms
64 bytes from node1 (192.168.1.7): icmp_seq=2 ttl=64 time=1.08 ms
64 bytes from node1 (192.168.1.7): icmp_seq=3 ttl=64 time=0.889 ms
64 bytes from node1 (192.168.1.7): icmp_seq=4 ttl=64 time=0.809 ms
^C
--- node1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.809/1.178/1.933/0.446 ms
masternode@masternode:~$ ping node2
PING node2 (192.168.1.8) 56(84) bytes of data.
64 bytes from node2 (192.168.1.8): icmp_seq=1 ttl=64 time=0.611 ms
64 bytes from node2 (192.168.1.8): icmp_seq=2 ttl=64 time=0.875 ms
64 bytes from node2 (192.168.1.8): icmp_seq=3 ttl=64 time=0.782 ms
64 bytes from node2 (192.168.1.8): icmp_seq=4 ttl=64 time=0.687 ms
^C
--- node2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3026ms
rtt min/avg/max/mdev = 0.611/0.738/0.875/0.099 ms
masternode@masternode:~$ ping node3
PING node3 (192.168.1.9) 56(84) bytes of data.
64 bytes from node3 (192.168.1.9): icmp_seq=1 ttl=64 time=2.04 ms
64 bytes from node3 (192.168.1.9): icmp_seq=2 ttl=64 time=0.742 ms
64 bytes from node3 (192.168.1.9): icmp_seq=3 ttl=64 time=0.749 ms
64 bytes from node3 (192.168.1.9): icmp_seq=4 ttl=64 time=0.596 ms
64 bytes from node3 (192.168.1.9): icmp_seq=5 ttl=64 time=0.785 ms
^Z
[3]+  Stopped                 ping node3
masternode@masternode:~$
```

3 DEFINING A USER FOR RUNNING MPI JOBS

It's essential to set up a dedicated user for running MPI (Message Passing Interface) jobs on your cluster. While the necessity of this step might not be immediately clear, several reasons justify its importance:

- **Consistency and Convenience**
- **SSH Communication**
- **NFS Directory Access**
- **Permissions and Security**

To create the MPI user, use the following command:

- **sudo adduser mpiuser --uid 999**

This command creates a new user named "mpiuser" with a user ID of 999. It's crucial to maintain consistency in the username and UID across all nodes in the cluster.

When prompted, enter a password for the MPI user. Using the same password across all nodes simplifies management.

The command also creates a home directory (/home/mpiuser) for the MPI user. This directory serves as the execution environment for MPI jobs on the cluster.

```
node3@node3-VirtualBox:~$ sudo adduser mpiuser --uid 999
Adding user `mpiuser' ...
Adding new group `mpiuser' (999) ...
Adding new user `mpiuser' (999) with group `mpiuser' ...
Creating home directory `/home/mpiuser' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for mpiuser
Enter the new value, or press ENTER for the default
        Full Name []: angel
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
node3@node3-VirtualBox:~$
```

Note: It's possible to use a different UID for the MPI user as long as it remains consistent across all nodes. However, it's recommended to keep it below 1000 to prevent the user from appearing on the login screen in desktop versions of Ubuntu.

4) INSTALL AND SETUP THE NETWORK FILE SYSTEM

Files and programs used for MPI jobs (jobs that are run in parallel on the cluster) need to be available to all nodes, so we give all nodes access to a part of the file system on the master node. Network File System (NFS) enables you to mount part of a remote file system so you can access it as if it is a local directory.

To install and set up the Network File System (NFS) on your cluster, follow these step-by-step instructions:

1. Install NFS on the Master Node

- Log in to the master node.
- Run the following command to install the NFS server package:
    - **sudo apt-get install nfs-kernel-server**

```
masternode@masternode:~$ sudo apt-get install nfs-kernel-server
[sudo] password for masternode:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nfs-kernel-server is already the newest version (1:2.6.1-1ubuntu1.2).
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 466 not upgraded.
masternode@masternode:~$
```

2. Install NFS Common on Compute Nodes

- Log in to each compute node.
- Run the following command to install the NFS common package:
    - **sudo apt-get install nfs-common**

```
node3@node3-VirtualBox:~$ sudo apt-get install nfs-common
[sudo] password for node3:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nfs-common is already the newest version (1:2.6.1-1ubuntu1.2).
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 470 not upgraded.
node3@node3-VirtualBox:~$
```

3. Check Ownership of Shared Directory

- Verify that the shared directory (/home/mpiuser) is owned by the MPI user:
    - **ls -l /home/ | grep mpiuser**

```
masternode@masternode:~$ ls -l /home/ |grep mpiuser
drwxr-xr-x  2 root       root       4096 May  5 01:23 mpiuser
masternode@masternode:~$
```

4. Share the Directory via NFS

- Edit the **/etc/exports** file on the master node:
    - **sudo vim /etc/exports**
- Add the following line to the file to share the **/home/mpiuser** directory:
    - **/home/mpiuser *(rw,sync,no_subtree_check)**

Save and exit the text editor.

```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/home/mpiuser *(rw,sync,no_subtree_check)
~
```

5. Restart NFS Service:

- Restart the NFS kernel server service:
    - **sudo service nfs-kernel-server restart**

6. Export Directories Listed in /etc/exports

- Run the following command to export the directories listed in /etc/exports:
    - **sudo exportfs -a**

```
masternode@masternode:~$ sudo vim /etc/exports
masternode@masternode:~$ sudo service nfs-kernel-server restart
masternode@masternode:~$ sudo exportfs -a
masternode@masternode:~$
```

7. **Test NFS Share**
   - From a compute node, run the following command to test if the NFS share is accessible:
     - **showmount -e masternode**

This should print the path **/home/mpiuser**.

```
node3@node3-VirtualBox:~$ showmount -e masternode
Export list for masternode:
/home/mpiuser *
node3@node3-VirtualBox:~$
```

```
node1@node1:~$ showmount -e masternode
Export list for masternode:
/home/mpiuser *
node1@node1:~$
```

8. **Allow Access from Specific Subnet**:
   - Run the following command on the master node to allow incoming access from a specific subnet:
     - **sudo ufw allow from 192.168.1.0/24**

```
masternode@masternode:~$ sudo ufw allow from 192.168.1.0/24
[sudo] password for masternode:
Rules updated
masternode@masternode:~$
```

If a different network was used, replace **'192.168.1.0/24'** with the subnet of that network.

9. **Mount NFS Share on Compute Nodes**:
   - On each compute node, run the following command to mount the NFS share:
     - **sudo mount masternode:/home/mpiuser /home/mpiuser**

```
masternode@masternode:~$ sudo mount masternode:/home/mpiuser /home/mpiuser
masternode@masternode:~$
```

10. **Automatically Mount NFS Share on Boot**:
    - Edit the **/etc/fstab** file on each compute node:
        - **sudo vim /etc/fstab**
    - Add the following line to the file:
        - **masternode:/home/mpiuser /home/mpiuser nfs**

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/ubuntu-vg/ubuntu-lv during curtin installation
/dev/disk/by-id/dm-uuid-LVM-wri9PW8MJOgjYDhYjOPG6uskhwDkwzDi964qyk7Jo1pssIvJdTCzpEVzQ6XzWSoa / ext4
defaults 0 1
# /boot was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/229d6c83-700a-4bc8-858b-e83210149e7b /boot ext4 defaults 0 1
/swap.img       none    swap    sw      0       0
masternode:/home/mpiuser /home/mpiuser nfs
```

11. **Reboot Compute Nodes**:
    - Reboot each compute node.
12. **Check Access to Shared Data**
    - To verify access to the shared data on each compute node is to create a test file on the master node's **/home/mpiuser** directory and check if it appears on each compute node.
    - **Create a Test File on the Master Node:** Log in to the master node. Run the following command to create a test file in the /home/mpiuser directory:
        - **sudo touch /home/mpiuser/test_file.txt**

```
masternode@masternode:~$ touch /home/mpiuser/test_file.txt
touch: cannot touch '/home/mpiuser/test_file.txt': Permission denied
masternode@masternode:~$ sudo touch /home/mpiuser/test_file.txt
[sudo] password for masternode:
masternode@masternode:~$
```

13. **Check if the Test File Appears on Compute Nodes**
    - Log in to each compute node then navigate to the **/home/mpiuser** directory.
    - Run the following command to list the contents of the directory:
        - **su mpiuser**
        - **ls /home/mpiuser**

```
node1@node1:~$ su mpiuser
Password:
mpiuser@node1:/home/node1$ ls /home/mpiuser
test_file.txt
mpiuser@node1:/home/node1$
```

```
node2@node2:~$ su mpiuser
Password:
mpiuser@node2:/home/node2$ ls /home/mpiuser
test_file.txt
mpiuser@node2:/home/node2$
```

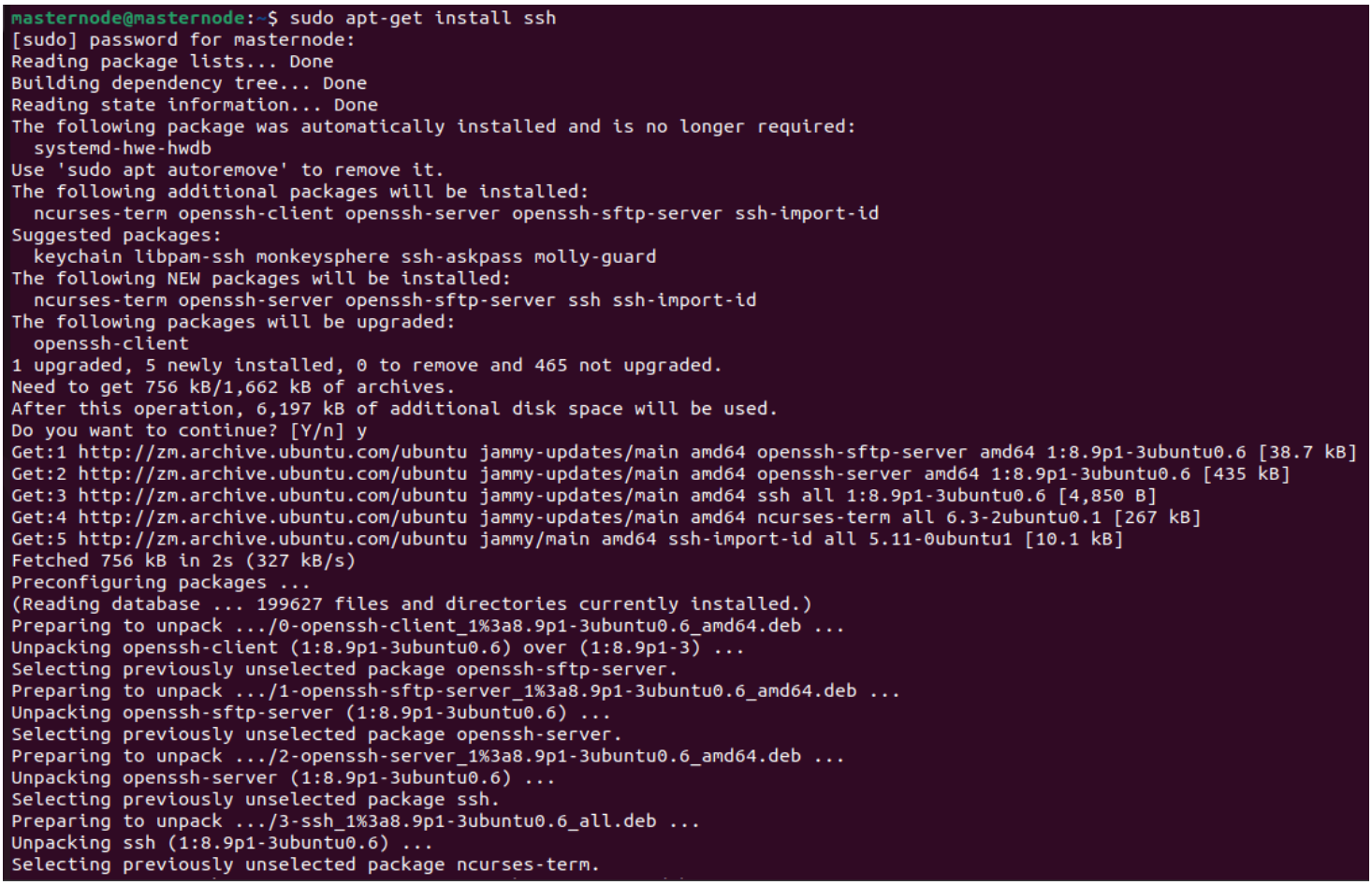
5) SETUP PASSWORDLESS SSH FOR COMMUNICATION BETWEEN THE NODES

In order for the cluster to function effectively, it's essential that communication flows smoothly between the master node and the compute nodes, and vice versa. Typically, Secure Shell (SSH) serves as the go-to method for secure remote connectivity among computers. By establishing passwordless SSH connections between the nodes, the master node gains the capability to execute commands on the compute nodes. This capability is vital for initiating and managing MPI daemons on the compute nodes.

Here are the steps to set up passwordless SSH communication between nodes in your cluster:

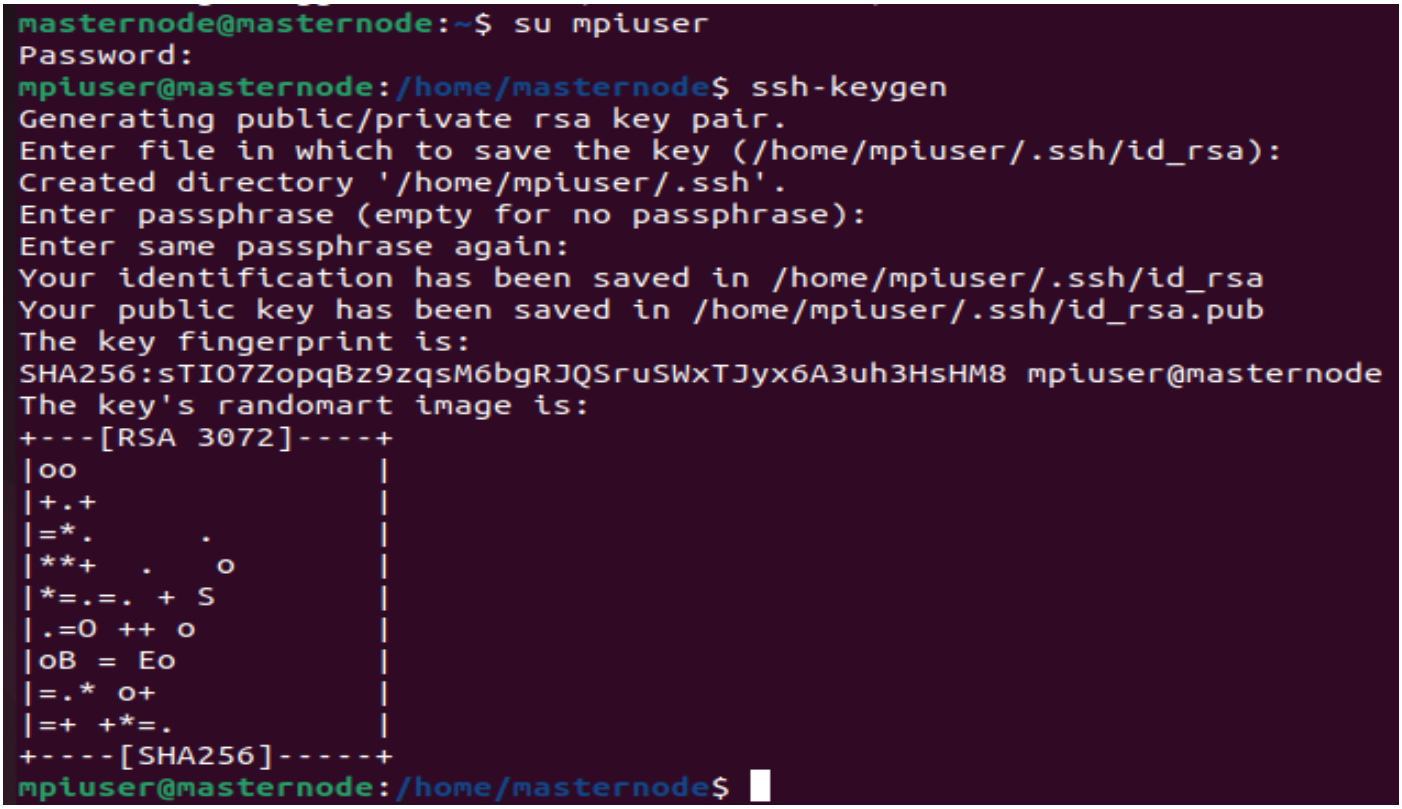
1. **Install SSH Server**:

- Run the following command on all nodes to install the SSH server:
  - **sudo apt-get install ssh**



```
masternode@masternode:~$ sudo apt-get install ssh
[sudo] password for masternode:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  ncurses-term openssh-client openssh-server openssh-sftp-server ssh-import-id
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh ssh-import-id
The following packages will be upgraded:
  openssh-client
1 upgraded, 5 newly installed, 0 to remove and 465 not upgraded.
Need to get 756 kB/1,662 kB of archives.
After this operation, 6,197 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://zm.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-server amd64 1:8.9p1-3ubuntu0.6 [38.7 kB]
Get:2 http://zm.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-server amd64 1:8.9p1-3ubuntu0.6 [435 kB]
Get:3 http://zm.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ssh all 1:8.9p1-3ubuntu0.6 [4,850 B]
Get:4 http://zm.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ncurses-term all 6.3-2ubuntu0.1 [267 kB]
Get:5 http://zm.archive.ubuntu.com/ubuntu jammy/main amd64 ssh-import-id all 5.11-0ubuntu1 [10.1 kB]
Fetched 756 kB in 2s (327 kB/s)
Preconfiguring packages ...
(Reading database ... 199627 files and directories currently installed.)
Preparing to unpack .../0-openssh-client_1%3a8.9p1-3ubuntu0.6_amd64.deb ...
Unpacking openssh-client (1:8.9p1-3ubuntu0.6) over (1:8.9p1-3) ...
Selecting previously unselected package openssh-sftp-server.
Preparing to unpack .../1-openssh-sftp-server_1%3a8.9p1-3ubuntu0.6_amd64.deb ...
Unpacking openssh-sftp-server (1:8.9p1-3ubuntu0.6) ...
Selecting previously unselected package openssh-server.
Preparing to unpack .../2-openssh-server_1%3a8.9p1-3ubuntu0.6_amd64.deb ...
Unpacking openssh-server (1:8.9p1-3ubuntu0.6) ...
Selecting previously unselected package ssh.
Preparing to unpack .../3-ssh_1%3a8.9p1-3ubuntu0.6_all.deb ...
Unpacking ssh (1:8.9p1-3ubuntu0.6) ...
Selecting previously unselected package ncurses-term.
```

2. **Generate SSH Key:**
   - Log in as the MPI user (mpiuser) on any node (preferably the master node):
     - **su mpiuser**
   - Generate an SSH key:
     - **ssh-keygen**
   - When prompted for a passphrase, leave it empty for passwordless SSH.

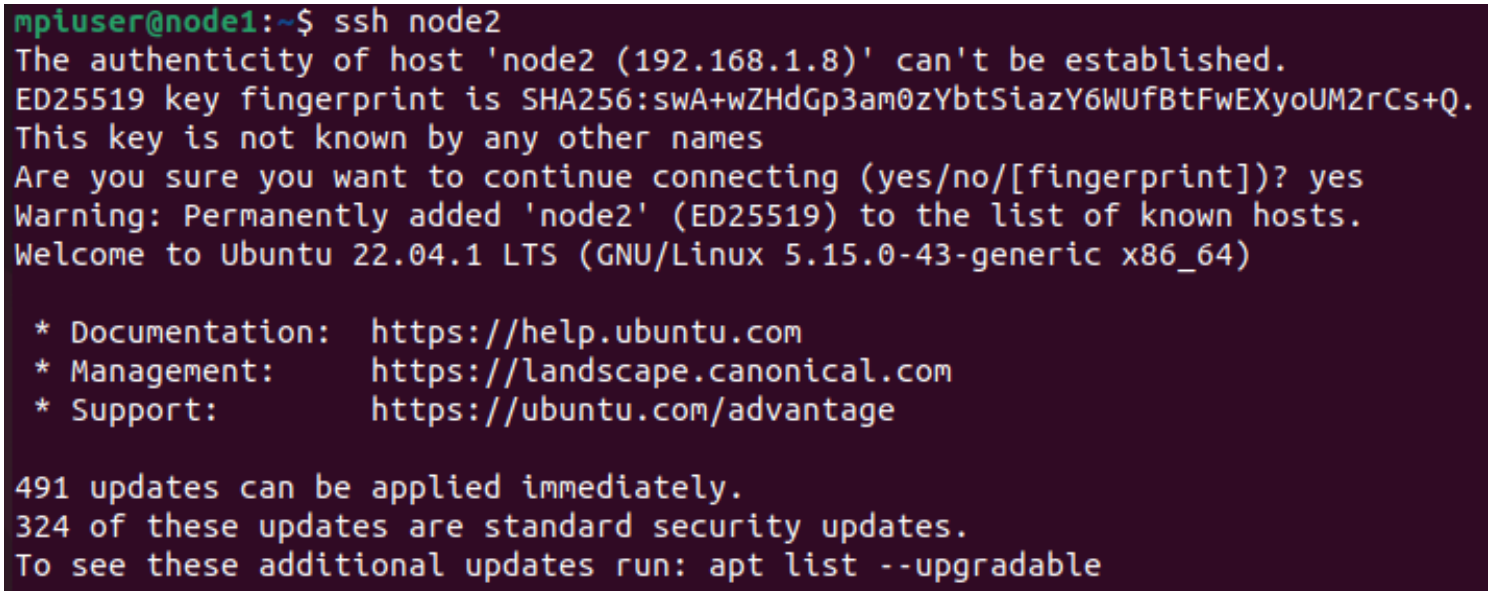


3. **Copy SSH Key to Localhost**:
   - Use the **ssh-copy-id** command to copy the SSH public key to localhost (master node). This command will add the key to **/home/mpiuser/.ssh/authorized_keys** and thus enable passwordless login . This key is visible to all the nodes.
     - **ssh-copy-id localhost**

```
mpiuser@masternode:/home/masternode$ ssh-copy-id localhost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/mpiuser/.s
sh/id_rsa.pub"
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:mvlyz9ui3gLJPCJoJckvsAyEqPhtS4VA458MbuTqimA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
 out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are promp
ted now it is to install the new keys
mpiuser@localhost's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'localhost'"
and check to make sure that only the key(s) you wanted were added.

mpiuser@masternode:/home/masternode$
```

4. **Verify SSH Access**:
   - Test SSH access from the master node to a compute node:
     - **ssh node1**
   - You should now be logged in to **node1** without entering a password.
   - Repeat this step to verify SSH access to all other nodes (**node2**, and **node3**).

```
mpiuser@masternode:/home/masternode$ ssh node1
The authenticity of host 'node1 (192.168.1.7)' can't be established.
ED25519 key fingerprint is SHA256:bHF3lIGcKIKLROSimDgiv37HWworVejcAPzna5mfMRM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Rhythmbox rmanently added 'node1' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-43-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

485 updates can be applied immediately.
322 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
```

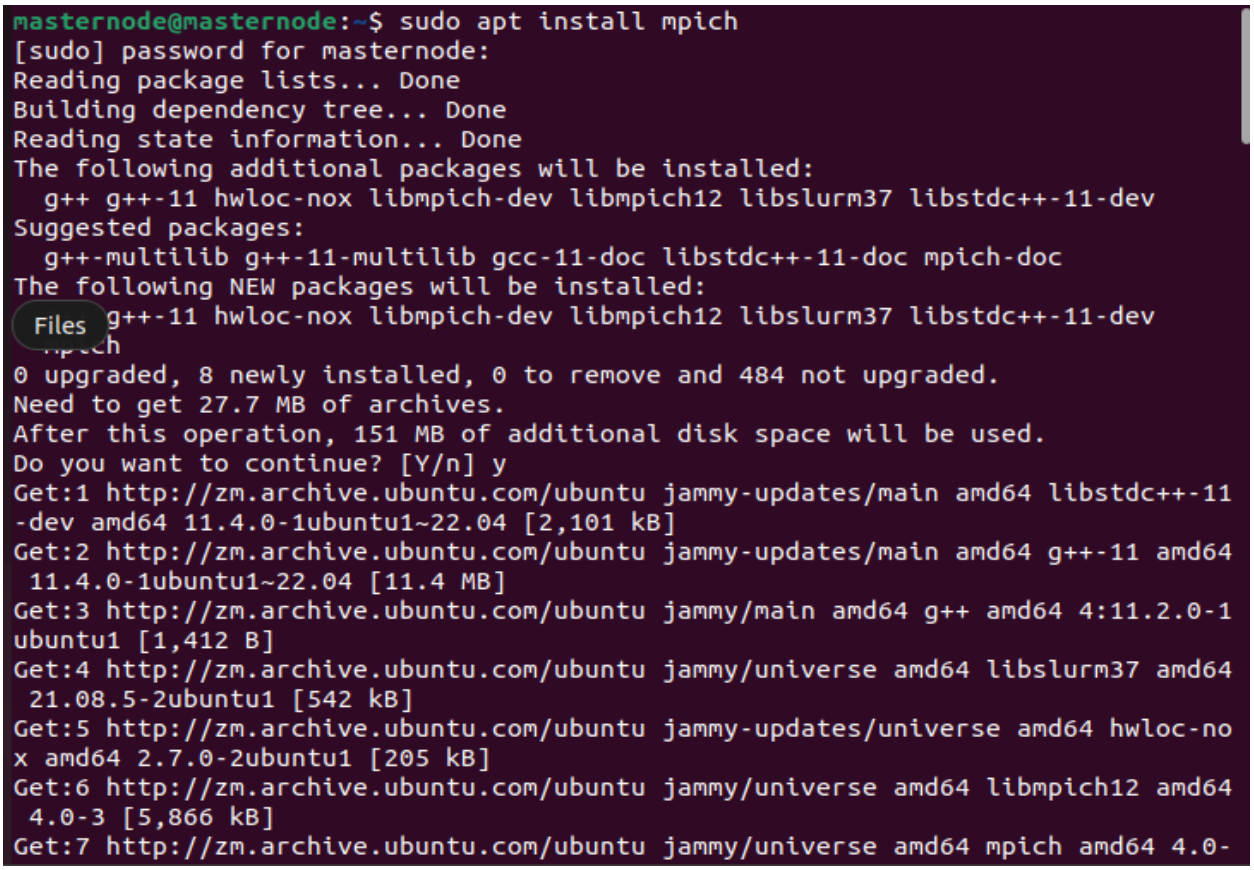


By following these steps, you'll establish passwordless SSH communication between the master node and compute nodes in your cluster using the existing **mpi** user.

6) SETTING UP THE PROCESS MANAGER

This section covers setting up the process manager for a Beowulf cluster. It starts with installing MPICH, then guides users in configuring the appropriate process manager based on their MPICH version. This step is crucial for efficient parallel job management across the cluster nodes, enabling high-performance computing tasks.

1. **Install MPICH**:
    - Run the command on all nodes of the cluster to install MPICH.
        - **sudo apt install mpich**



2. **Check MPICH Version**:
    - Use in the terminal to determine the version of MPICH installed on your system.
        - **Mpichversion**

```
masternode@masternode:~$ mpichversion
MPICH Version:          4.0
MPICH Release date:     Fri Jan 21 10:42:29 CST 2022
MPICH Device:           ch4:ofi
MPICH configure:        --build=x86_64-linux-gnu --prefix=/usr --includedir=${p
refix}/include --mandir=${prefix}/share/man --infodir=${prefix}/share/info --sy
sconfdir=/etc --localstatedir=/var --disable-option-checking --disable-silent-r
ules --libdir=${prefix}/lib/x86_64-linux-gnu --runstatedir=/run --disable-maint
ainer-mode --disable-dependency-tracking --with-libfabric=/usr --with-slurm=/us
```

3. **Select Process Manager**:
    - Depending on the MPICH version:
        - If using MPICH 1.2.x or earlier, MPD is the default process manager.
        - If using MPICH 1.3.x or later, Hydra is the default process manager.
4. **Configure Process Manager:**
    - If MPD is the default process manager:
        - Follow the steps provided for configuring MPD.
    - If Hydra is the default process manager:
        - Follow the steps provided for configuring Hydra.

6.1) SETTING UP HYDRA

1. **Create Hosts File**:
   - Log in to the master node.
   - Change to the MPI user's home directory:
     - **cd ~**
   - Create a file named **hosts**:
     - **touch hosts**
2. **Add Compute Node Hostnames:**
   - Open the hosts file for editing:
     - **vim hosts**

```
masternode@masternode:~$ su mpiuser
Password:
mpiuser@masternode:/home/masternode$ cd ~
mpiuser@masternode:~$ touch hosts
mpiuser@masternode:~$ vim hosts
```

   - Add the hostnames of all compute nodes to the hosts file, each on a new line:
     - **node1**

     node2

     node3

Optionally, include the master node hostname if it will also act as a compute node.

```
masternode
node1
node2
node3
~
~
```

Note: The **hosts** file only needs to be present on the node used to start jobs on the cluster, typically the master node. However, because the home directory is shared among all nodes, all nodes will have access to the **hosts** file.

7) RUNNING JOBS ON THE CLUSTER

7.1) RUNNING MPICH EXAMPLE APPLICATIONS ON THE CLUSTER

1. **The MPICH package came with a few example applications to test the cluster. The following are the commands that were run on the master node:**
   - To download the package from which the example applications were built
     - **$ wget http://www.mpich.org/static/downloads/3.3.2/mpich-3.3.2.tar.gz**

```
masternode@masternode:~$ wget http://www.mpich.org/static/downloads/3.3.2/mpich
-3.3.2.tar.gz
URL transformed to HTTPS due to an HSTS policy
--2024-05-13 04:01:35--  https://www.mpich.org/static/downloads/3.3.2/mpich-3.3
.2.tar.gz
Resolving www.mpich.org (www.mpich.org)... 172.65.90.27, 172.65.90.25, 172.65.9
0.24, ...
Connecting to www.mpich.org (www.mpich.org)|172.65.90.27|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 27311775 (26M) [application/x-gzip]
Saving to: ‘mpich-3.3.2.tar.gz’

mpich-3.3.2.tar.gz  100%[===================>]  26.05M   815KB/s    in 27s

2024-05-13 04:02:03 (995 KB/s) - ‘mpich-3.3.2.tar.gz’ saved [27311775/27311775]

masternode@masternode:~$
```

2. **To extract the package:**
   - **$ tar -xvzf mpich-3.3.2.tar.gz**

```
mpich-3.3.2/src/hwloc/utils/hwloc/test-hwloc-diffpatch.input2
mpich-3.3.2/src/hwloc/utils/hwloc/hwloc.7in
mpich-3.3.2/src/hwloc/utils/hwloc/hwloc-info.1in
mpich-3.3.2/src/hwloc/utils/hwloc/hwloc-dump-hwdata.c
mpich-3.3.2/src/hwloc/utils/hwloc/hwloc-patch.1in
mpich-3.3.2/src/hwloc/utils/hwloc/test-hwloc-compress-dir.input.tar.gz
mpich-3.3.2/src/hwloc/utils/hwloc/test-hwloc-compress-dir.output.tar.gz
mpich-3.3.2/src/hwloc/utils/hwloc/hwloc-diff.c
mpich-3.3.2/src/hwloc/utils/hwloc/hwloc-distrib.1in
mpich-3.3.2/src/hwloc/utils/hwloc/test-hwloc-diffpatch.input1
mpich-3.3.2/src/hwloc/utils/hwloc/hwloc-dump-hwdata.1in
mpich-3.3.2/src/hwloc/utils/hwloc/hwloc-calc.1in
mpich-3.3.2/src/hwloc/utils/lstopo/
mpich-3.3.2/src/hwloc/utils/lstopo/lstopo-text.c
mpich-3.3.2/src/hwloc/utils/lstopo/lstopo-draw.c
mpich-3.3.2/src/hwloc/utils/lstopo/Makefile.in
mpich-3.3.2/src/hwloc/utils/lstopo/lstopo.h
mpich-3.3.2/src/hwloc/utils/lstopo/lstopo-ascii.c
mpich-3.3.2/src/hwloc/utils/lstopo/Makefile.am
mpich-3.3.2/src/hwloc/utils/lstopo/lstopo-color.c
mpich-3.3.2/src/hwloc/utils/lstopo/test-lstopo.output
mpich-3.3.2/src/hwloc/utils/lstopo/lstopo-windows.c
mpich-3.3.2/src/hwloc/utils/lstopo/lstopo.c
mpich-3.3.2/src/hwloc/utils/lstopo/test-lstopo.sh.in
mpich-3.3.2/src/hwloc/utils/lstopo/lstopo-fig.c
mpich-3.3.2/src/hwloc/utils/lstopo/lstopo-no-graphics.1in
mpich-3.3.2/src/hwloc/utils/lstopo/lstopo-xml.c
mpich-3.3.2/src/hwloc/utils/lstopo/lstopo-cairo.c
mpich-3.3.2/src/hwloc/utils/lstopo/lstopo.desktop
```

3. **To enter the extracted directory:**
    - **$ cd mpich-3.3.2/**
- To run the ‘configure’ script in the directory that built the example applications:

  $ ./configure

```
masternode@masternode:~$ cd mpich-3.3.2/
masternode@masternode:~/mpich-3.3.2$ ./configure
Configuring MPICH version 3.3.2
Running on system: Linux masternode 5.15.0-43-generic #46-Ubuntu SMP Tue Jul 12
 10:30:17 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
checking for icc... no
checking for pgcc... no
checking for xlc... no
checking for xlC... no
checking for pathcc... no
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
```

- To enter the directory containing the example applications:
  - **$ cd examples/**
- To copy the examples folder to the mpiuser's bin directory:
  - **sudo mkdir -p /home/mpiuser/bin && sudo cp examples.sln /home/mpiuser/bin**

```
masternode@masternode:~/mpich-3.3.2/examples$ sudo mkdir -p /home/mpiuser/bin &
& sudo cp -r examples.sln /home/mpiuser/bin
masternode@masternode:~/mpich-3.3.2/examples$
```

4. **You can verify that the files have been copied to the specified directory by listing the contents of that directory.**
   - Here's how to do it:
     - **ls /home/mpiuser/bin**

```
masternode@masternode:~$ su mpiuser
Password:
mpiuser@masternode:/home/masternode$ ls /home/mpiuser/bin
examples.sln
mpiuser@masternode:/home/masternode$
```

5. To verify that the files have been copied to the **/home/mpiuser/bin** directory on other nodes, you can use SSH to access each node and check the contents of the directory.
   - **ssh node1**
   - Once logged in to each node, run the ls command to check the contents of the **/home/mpiuser/bin** directory:
     - **ls /home/mpiuser/bin**

Repeat steps 1 and 2 for nodes 2 and 3 in your cluster to ensure that the files have been copied to all nodes.

```
mpiuser@masternode:~$ ssh node1
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-43-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

485 updates can be applied immediately.
322 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Sun May 12 16:10:47 2024 from 192.168.1.6
mpiuser@node1:~$ ls /home/mpiuser/bin
examples.sln
```

```
mpiuser@node1:~$ ssh node2
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-43-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

491 updates can be applied immediately.
324 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Sun May 12 13:40:01 2024 from 192.168.1.7
mpiuser@node2:~$ ls /home/mpiuser/bin
examples.sln
mpiuser@node2:~$
```

7.2) RUNNING BIOINFORMATICS TOOLS ON THE CLUSTER

1. **Download ClustalW-MPI source:**
   - The provided website (http://www.clustal.org/clustal2/) might not have the latest version. Search online for the latest stable version and download the source code (e.g., .tar.gz archive) from a reliable source. Use the following website:
     - https://blast.ncbi.nlm.nih.gov/doc/blast-help/downloadblastdata.html
   - Find the link to download the source code of ClustalW-MPI and click on it. This will typically be a compressed archive file (e.g., **.zip** or **.tar.gz**):
     - https://ftp.ncbi.nlm.nih.gov/pub/mmdb/cdd/little_endian/Cdd_NCBI_LE.tar.gz.
2. **Extract the Package:**
   - Once the download is complete, navigate to the directory where the file was downloaded and extract the contents of the archive. You can use the following commands in the terminal:
     - **cd /path/to/downloaded/file/directory**
     - **tar -xvzf filename.tar.gz**

```
masternode@masternode:~/Downloads$ ls
Cdd_NCBI_LE  Cdd_NCBI_LE.tar.gz
masternode@masternode:~/Downloads$ tar -xvzf Cdd_NCBI_LE.tar.gz
Cdd_NCBI.00.aux
Cdd_NCBI.00.freq
Cdd_NCBI.00.loo
Cdd_NCBI.00.pdb
Cdd_NCBI.00.phr
Cdd_NCBI.00.pin
Cdd_NCBI.00.pos
Cdd_NCBI.00.pot
Cdd_NCBI.00.psq
Cdd_NCBI.00.ptf
Cdd_NCBI.00.pto
Cdd_NCBI.00.rps
Cdd_NCBI.01.aux
Cdd_NCBI.01.freq
Cdd_NCBI.01.loo
Cdd_NCBI.01.pdb
Cdd_NCBI.01.phr
Cdd_NCBI.01.pin
Cdd_NCBI.01.pos
Cdd_NCBI.01.pot
Cdd_NCBI.01.psq
Cdd_NCBI.01.ptf
Cdd_NCBI.01.pto
Cdd_NCBI.01.rps
Cdd_NCBI.02.aux
Cdd_NCBI.02.freq
```