=1) Register each host with host name:

a) Add following statement to /etc/hosts file in both nodes. (Master node to submit jobs; node1, node2 are computing node)
(Assumption: 10.1.3.118 -> master, 10.1.3.117 -> node1)

```
10.1.3.118        master
10.1.3.117        node1
```

b) Test connection using host name instead of IP address using ping command from both systems

```
> ping -c 3 node1              // from master node, send 3 packets
> ping -c 3 master
```

2) Create mpiuser:

Same user for all nodes for easy MPI communication (Refer Beowulf setup document for detail understanding of Why common user)

If user id <1000 then it is prevented to display in login screen. Hence, create user with username 'mpiuser', group 'mpiuser' and user id 999, and password 'mpiuser123' using following command

```
> sudo adduser mpiuser --uid 999
```

3) Install NFS:

Install NFS to provide node's file access to master node.

a) Run following command on master node.

```
> Sudo apt-get install nfs-kernel-server
```

b) Run following command on other nodes in order to make it possible to mount a Network File System on the compute nodes. Install nfs-common

```
> sudo apt-get install nfs-common
```

4**) Share directory for MPI jobs:**

Share a directory on master to other computing nodes using NFS

a) Share one directory (i.e. /home/mpiuser) of master node to all computing nodes.

File /etc/exports contain contains a table of local physical file systems on an NFS server that are accessible to NFS clients.

Each file systems on physical table contain options and control-lists. Refer man exports for detail.

i.e. rw - read/write privilege;
sync - (Synchronous) file share request is only replied

After changes has been committed to stable storage.
no_subtree_check - prevent to check/access sub tree.
=> No need to export entire tree
=> Better performance. And security

Note: All files and programs used to run MPI job -> must be placed in this shared directory (i.e. here /home/mpiuser)

For this edit /etc/exports file on master node. Append following line

```
/home/mpiuser *(rw,sync,no_subtree_check)
```

b) Restart nfs daemon using following command on master node
> sudo service nfs-kernel-server restart
In Future, if /etc/export is modified then following command can be used to export directories listed in this file.
> sudo exportfs -a

c) By Default: Firewall is enabled on Ubuntu => which block client's access to NFS shared directory
So, it is required to add rule using UFW (tool for firewall) to allow access from specific subnet.

Run following command on master to allow incoming access from specific subnet
> sudo ufw allow from 10.1.3.0/24

d) 1. Manual mount on each boot on computing node (of NFS shared directory to computing nodes)

This command will allow you to mount master's NFS shared directory on computing node.

TEST: Run following command on computing node to test this.
> sudo mount master:/home/mpiuser /home/mpiuser

If this command fails, restart computing node.
Else try to list content of master node's content (files created on /home/mpiuser on master)
> ls /home/mpiuser

2. Automatic mount on every boot on computing node.
Step 1: On Computing node => Edit /etc/fstab file and add following line
master:/home/mpiuser /home/mpiuser nfs

5) set passwordless SSH for communication between nodes:

For cluster to work, master node need to communicate with computing nodes. Master node can execute commands on computing nodes.
SSH makes secure remote access.

a) Install ssh server on all nodes using following commands.
> sudo apt-get install ssh

b) generate SSH key for all MPU users on all nodes. The SSH key is by default created in the user's home directory (i.e./home/mpiuser for mpiuser login)
So, if we generate SSH key for mpi user on one node, it will be automatically available on all node. Use following command to generate ssh key as mpiuser on master node.
- su mpiuser
- ssh-keygen

c) Run the following commands on the master node as user "mpiuser",
> mpiuser@master:~$ ssh-copy-id localhost

d) This means rthat we should now be able to login on the compute nodes from the master node without having to enter a password
mpiuser@master:~> ssh node1
mpiuser@master:~> echo $HOSTNAME
node1

Note: If "ssh node1" command ask for password, then ssh-key has not been succssefully copied. Try following command to make it password-less ssh connection

```
mpiuser@master:~> ssh-copy-id –i ~/.ssh/id_rsa.pub node1
mpiuser@master:~> ssh node1
```

6) MPICH installation and configuring the process manager.

a) Installation of MPICH

The process manager is needed to spawn and manage parallel jobs on the cluster.

These process managers communicate with MPICH processes using a predefined interface called as PMI (process management interface)

The process manager is included with the MPICH package, so start by **installing MPICH on all nodes with following command**

```
> sudo apt-get install mpich2
```

b) Check MPICH version

```
> mpich2version
```

Here, we use 1.4

c) Configure Process manager

MPD has been the traditional default process manager for MPICH till the 1.2.x release series. Starting the 1.3.x series, Hydra is the default process manager. So, depending on the version of MPICH you are using, you should either use MPD or Hydra for process management.

Set up Hydra

In order to setup Hydra, we need to create one file on the master node.

This file contains all the host names of the compute nodes. This file can be created anywhere, but preferred to create in mpiuser's home directory.

i) Change file timestamps for synchronization

```
>cd ~
>touch hosts
```

ii) In order to be able to send out jobs to the other nodes in the network, **add the host names of all compute nodes to the `hosts` file.** Master node also can be added to hosts file if you like to use master also as working node.

Hosts file should be present to node, which will be used to start job on cluster, typically a master node.

iii) Configure

d) Download examples, and Running jobs on the cluster

The MPICH2 package comes with a few example applications that you can run on your cluster. Download examples using following commands:

```
> mkdir examples
> sudo apt-get install build-dep mpich2
> wget http://www.mpich.org/static/downloads/1.4.1/mpich2-1.4.1.tar.gz
```

```
> tar -xvzf mpich2-1.4.1.tar.gz
> cd mpich2-1.4.1/
> ./configure
> make
> cd examples/
```

Executables will be available in examples folder. Other examples can be built like

```
> make yellow
> make pmande1
```

Once compiled , place executable file in a directory /home/mpiuser/bin (note: create bin directory)

Run the job

```
> su mpiuser
```

(on 3 nodes using Hydra, file hosts specify name of file contain host name of to run the job)

```
> mpiexec –f hosts -n 3 /home/mpiuser/bin/cpi
```

Compile and execute mpi code

```
> mpicc -o mpi1.c -o mpi1
```

Execute

```
>mpirun -np 4 -hosts /etc/hosts ./hello
```

cluster run issue

1) if username doesnot displayed as mpiuser@master in prompt, system may not interact each other . Give error : cannot locate hostname my-desktop (machine hostname)

Solution: modify hostname to /etc/hostname as master in master node and node1 to node1 node.

2) Permission error for mpi execution

Solution: change firewall setting given below as given below in 1) setup for MPI code execution

3) s
4) Every execution of mpirun or mpiexec ask password for each computenode. So, do SSH configuration and ssh-copy-id as per steps given in above Step-f5

MPI Code Execution

1) setup

The firewall is by default enabled onUbuntu. The firewall will block access when a client tries to access an NFS shared directory. So you need to add a rule with UFW (a tool for managing the firewall) to allow access from a specific subnet.

Run the following command to allow incoming access from a specific subnet, i.e

```
mpiuser@master:~$ sudo ufw allow from 10.1.3.114/24
```

2) mount master:/home/mpiuser on the compute nodes using following command:

>SUdo mount master:/home/mpiuser /home/mpiuser

Note: to mount everytime when compute node is booted, add following line to /etc/fstab file of all computing nodes

master:/home/mpiuser /home/mpiuser nfs

3) Location of MPI Job files

Store all cluster job file to /home/mpiuser of master node

4) compilation

mpiuser@master:~$>mpicc <your-mpi-C-filename> -o <objfilename>

e.g. $> mpicc MPI1.c -o MPI1.o

5) execution

(examples avialble in MPICH2 pacakge)

Note: hosts file in current direcotry store hostname of all computing nodes only

```
mpiuser@master:~$> mpiexec  -f hosts -n 3  /home/mpiuser/bin/cpi
```

```
mpiuser@master:~$> mpiexec -f hosts -n 4 MPI1.o
or
mpiuser@master:~$> mpiexec -np 4 MPI1.o -machinefile ../my_hosts
   or
mpiuser@master:~$> mpirun -n 4 MPI1.o -machinefile ../my_hosts
```

openmp+mpi

==========

1) create Openmp+MPI file and store it :/home as mm_OMP_MPI.

```
#include <stdio.h>
#include "mpi.h"
#include <omp.h>

int main(int argc, char *argv[]) {
  int numprocs, rank, namelen;
  char processor_name[MPI_MAX_PROCESSOR_NAME];
  int iam = 0, np = 1;

  MPI_Init(&argc, &argv);
  MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);
  MPI_Get_processor_name(processor_name, &namelen);

  #pragma omp parallel default(shared) private(iam, np)
  {
    np = omp_get_num_threads();
    iam = omp_get_thread_num();
    printf("Hello from thread %d out of %d from process %d out
of %d on %s\n",
           iam, np, rank, numprocs, processor_name);
  }

  MPI_Finalize();
}
```

2) Compilation

```
    mpiuser@master:~$    mpicc mm_OMP_MPI.c -openmp  -o
mpicc_omp_mpi.o
    mpiuser@master:~$    export OMP_NUM_THREADS=4
    mpiuser@master:~$    mpirun -n 4 -machimename machinefile
../my_hosts -o mpicc_omp_mpi.o
```