

THE UNIVERSITY OF NOTTINGHAM NINGBO CHINA

Installation and Configuration

Technical Document

Written by

23 June, 2015

Big Data and Visual Analytics

Department of Computer Science

Abstract -----	1
1. Machine Set Up -----	2
1.1 Installation of Memory Bank -----	2
1.2 Installation of Ubuntu -----	2
2. Cloning using Clonezilla -----	12
2.1 Introduction to Clonezilla -----	12
2.2 Setting up Clonezilla for server -----	12
2.3 Setting up Clonezilla for client -----	13
3. Installation and configuration of MongoDB -----	15
3.1 Introduction -----	15
3.2 Installation of mongodb-linux-x86_64-3.0.4 -----	15
3.3 Change the IP address for each computer -----	15
3.4 Change the file system of mongoldb into wiredTiger -----	16
3.5 Configure Sharding Cluster of MongoDB -----	17
4. MPI -----	19
4.1 MPI (Message Passing Interface) Introduction -----	19
4.2 MPI Setting Up Instruction -----	19
4.2.1 Both server and client should do -----	19
4.2.2 Only server should do -----	19
4.2.3 Only clients should do -----	21
4.2.4 MPI compiling and running -----	21

Abstract

In this tutorial, the specific steps of setting up a linux machine and how to use Clonezilla to do cloning are described. Installation and configuration of MongoDB and the way to configure sharding cluster of MongoDB are also discussed here. The instructions of setting up MPI, compiling and running are given at the end.

1. Machine Set Up

1.1 Installation of Memory Bank

1. Definition of memory bank: A logical unit of storage in electronics, which is hardware dependent.



2. Standard of memory bank: 2G.

3. Number of memory bank: [n | n <- [1] ++ [2, 4, 6..n]].

4. Means of installation:

- Two memory banks should be in the same group and the same direction.
- Unlock the locks on the both side of the memory bank, then plug in the new memory bank and lock the locks.
- Make sure the memory banks are installed stably.

1.2 Installation of Ubuntu

1. Definition of Ubuntu: Ubuntu is a Debian-based Linux operating system, with Unity as its default desktop environment. It is based on free software and named after the Southern African philosophy of ubuntu (literally, "human-ness"), which often is translated as "humanity towards others" or "the belief in a universal bond of sharing that connects all humanity".

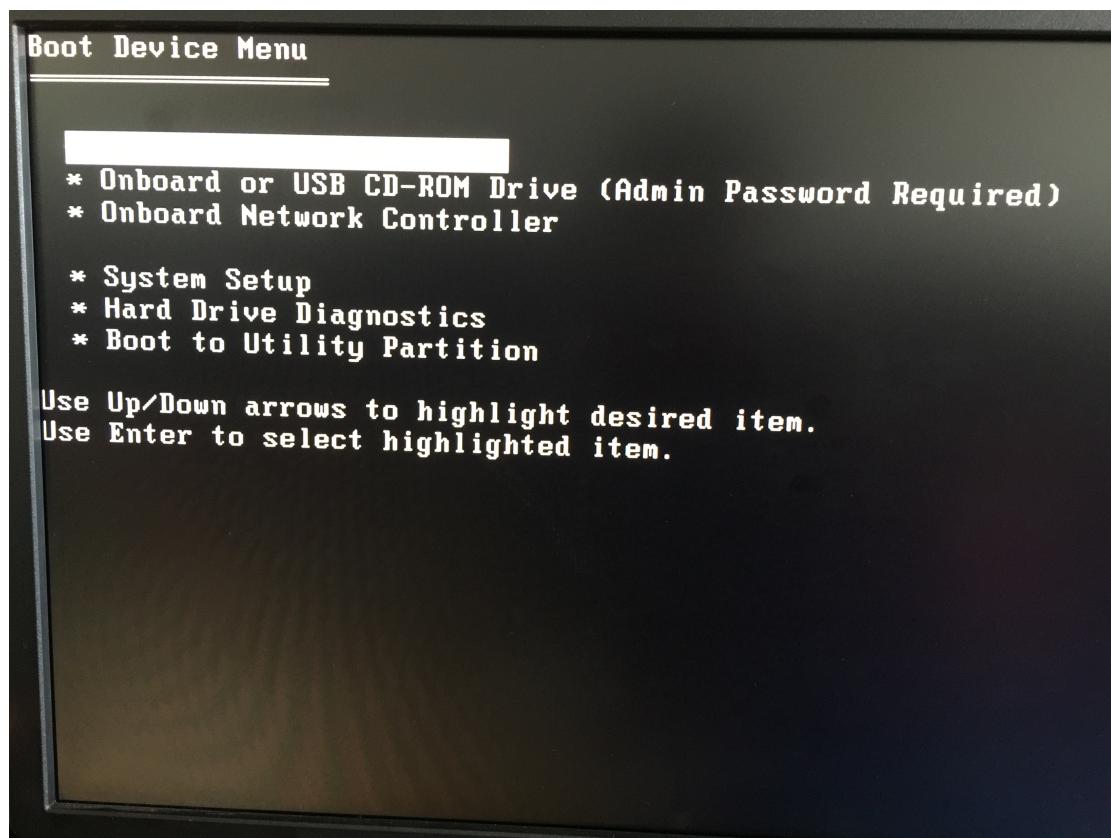
2. Installation steps:

- Insert the CD and power on the computer, push F12 to enter boot system while

the DELL label appears on the screen.

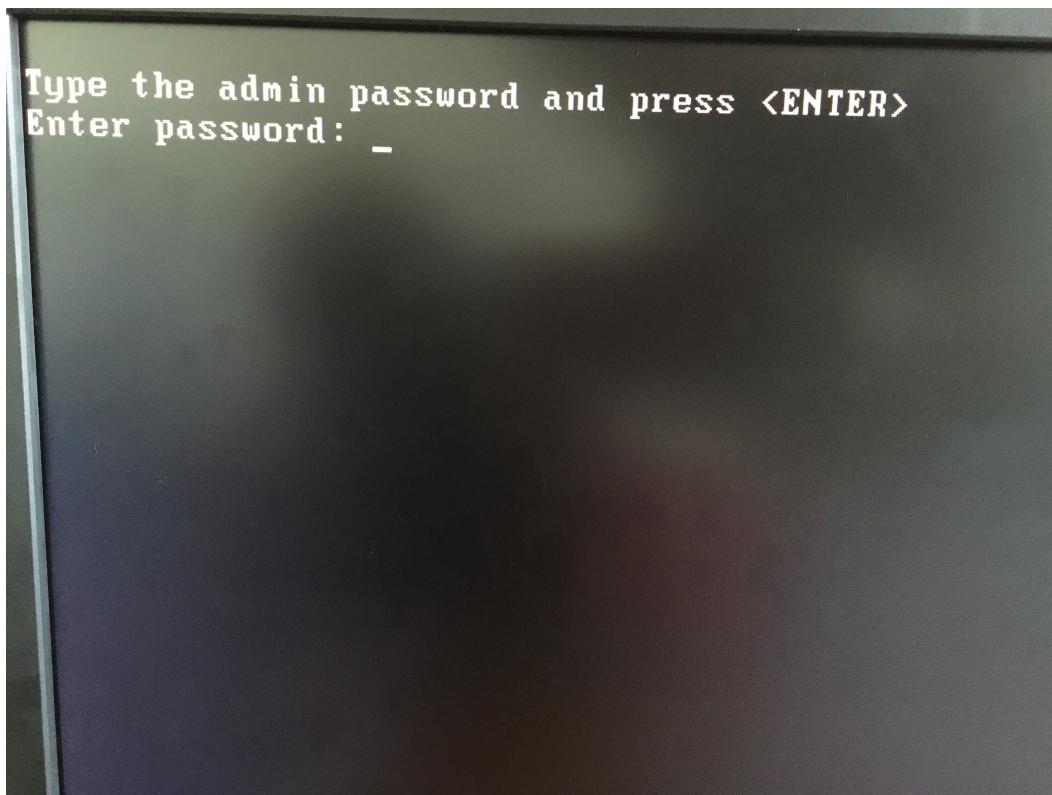
b) Boot Device Menu:

Select **Onboard or USB CD-ROM Drive (Admin Password Required)**.



c) Log in the system:

Password: **m1cky**.



d) Language selection:

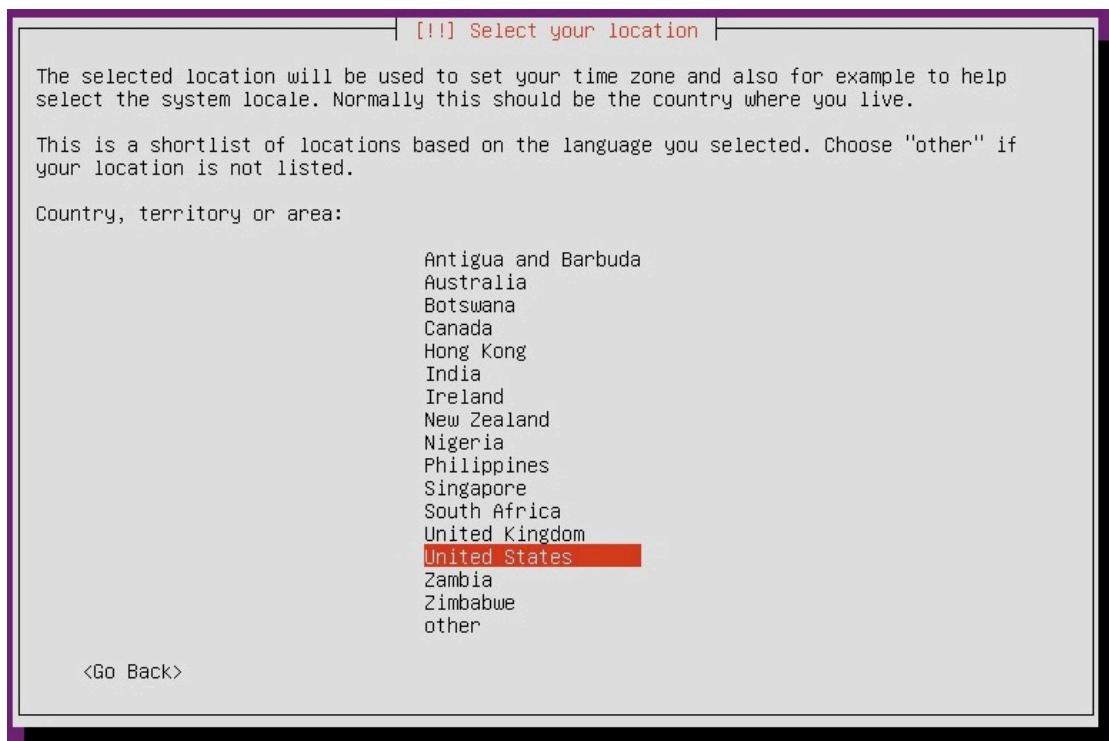
Select **English**.

The screenshot shows a window titled "[!!] Select a language". The text inside says: "Choose the language to be used for the installation process. The selected language will also be the default language for the installed system." Below this, there is a list of languages with their corresponding localization names. The "English" option is highlighted with a red rectangle. At the bottom left, there is a "<Go Back>" button, and at the bottom right, a message "**<Tab> moves; <Space> selects; <Enter> activates buttons**".

C	- No localization
Albanian	- Shqip
Arabic	- عربى
Asturian	- Asturianu
Basque	- Euskara
Belarusian	- Беларуская
Bosnian	- Bosanski
Bulgarian	- Български
Catalan	- Català
Chinese (Simplified)	- 中文(简体)
Chinese (Traditional)	- 中文(繁體)
Croatian	- Hrvatski
Czech	- Čeština
Danish	- Dansk
Dutch	- Nederlands
English	- English
Esperanto	- Esperanto
Estonian	- Eesti
Finnish	- Suomi
French	- Français
Galician	- Galego
German	- Deutsch
Greek	- Ελληνικά

e) Location selection:

Select **United States**.

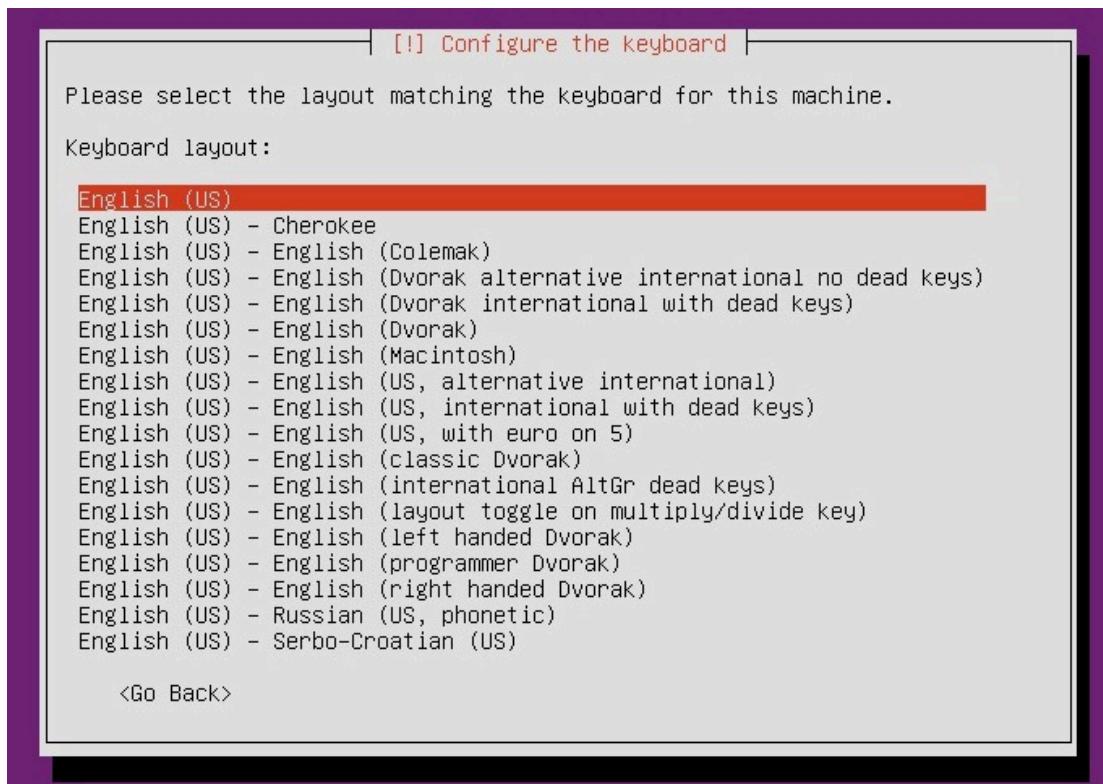


f) Select **Install Ubuntu Server**

g) Keyboard detection:

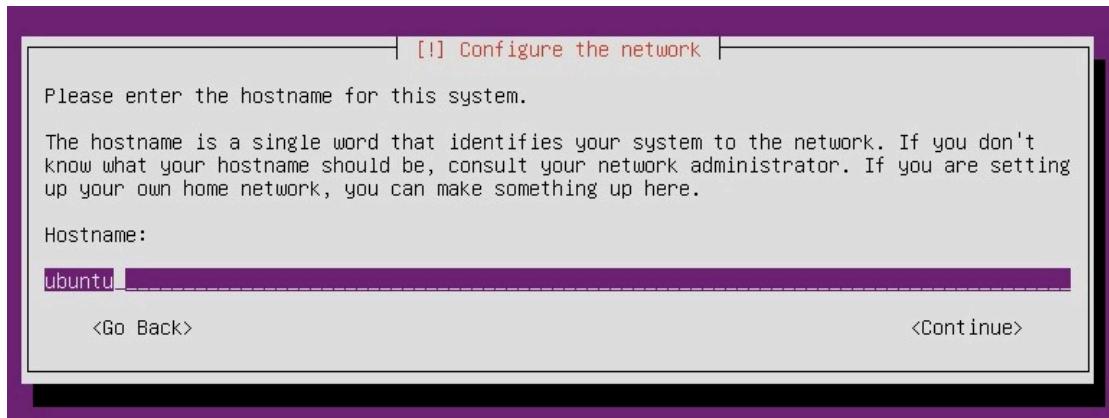
Select **no** and **US**.





h) Network configuration:

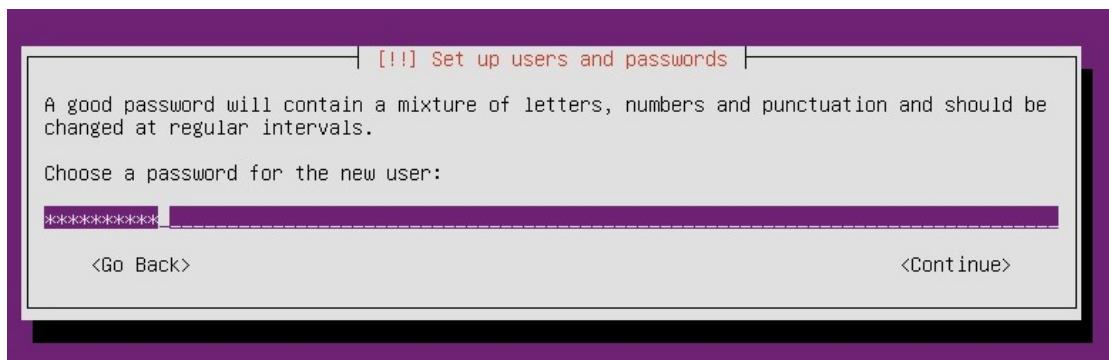
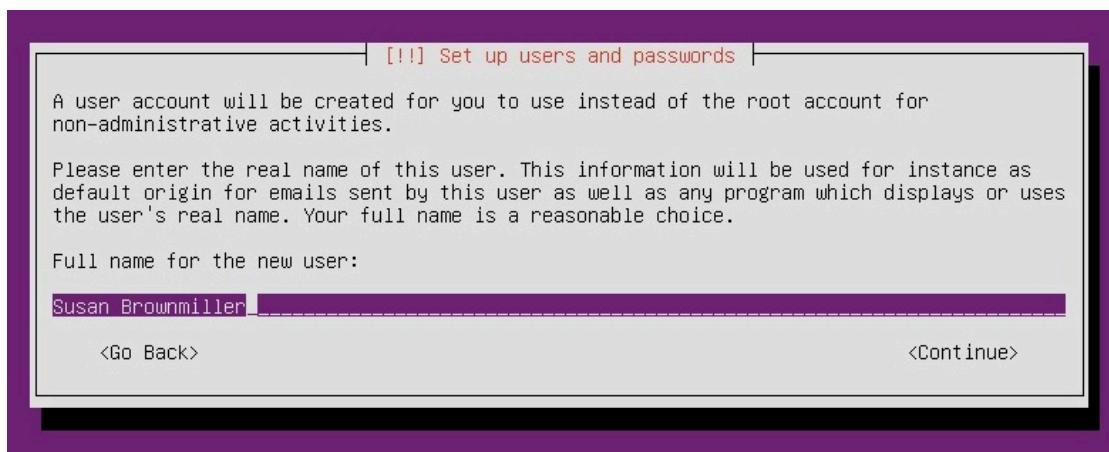
Hostname: **bigdata1, bigdata2...**



i) User and password set up:

Username: **big data**

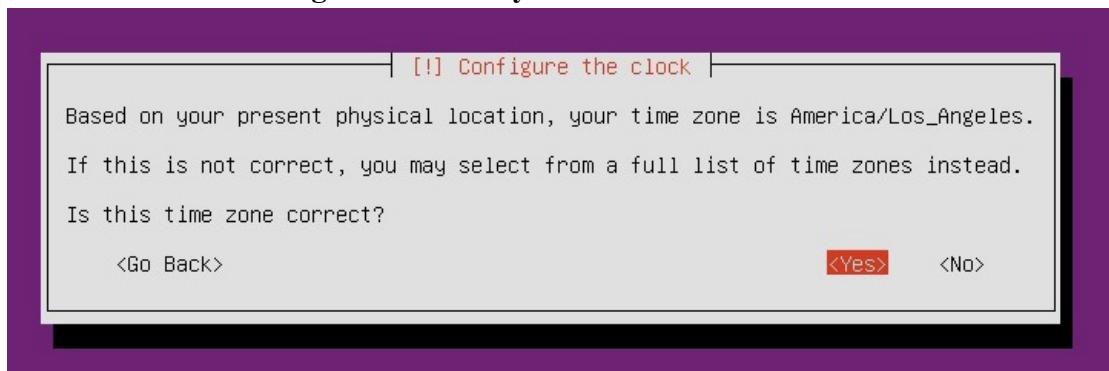
Password: **idic2014**





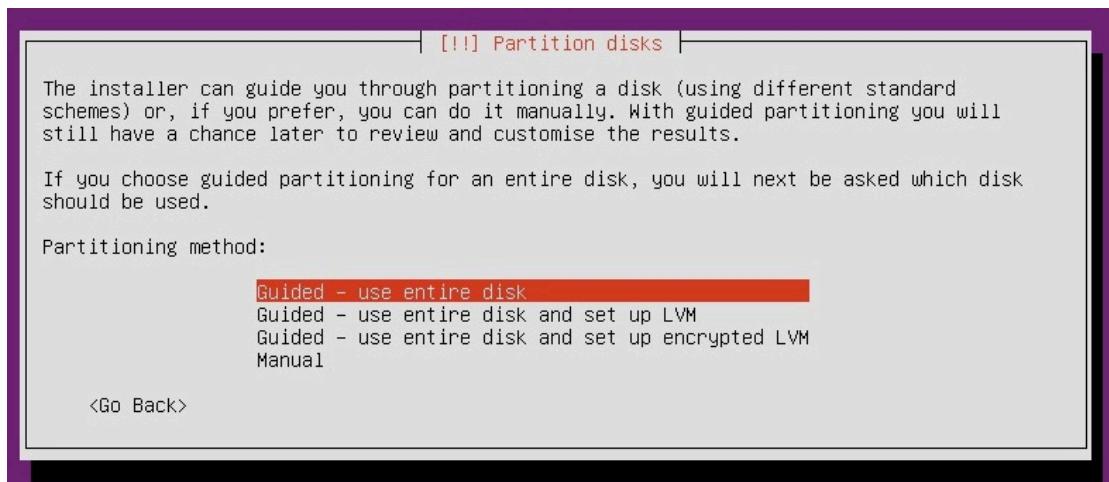
j) Clock configuration:

Select **China/Shanghai** and select yes.



k) Partition disks:

Select **Guided-use entire disk** and yes.



[!!] Partition disks

If you continue, the changes listed below will be written to the disks. Otherwise, you will be able to make further changes manually.

The partition tables of the following devices are changed:
SCSI1 (0,0,0) (sda)

The following partitions are going to be formatted:
partition #1 of SCSI1 (0,0,0) (sda) as ext4
partition #5 of SCSI1 (0,0,0) (sda) as swap

Write the changes to disks?

<Yes>

<No>

l) Mirror of Ubuntu archive:

Select **Continue** without filling in anything.

[!] Choose a mirror of the Ubuntu archive

If you need to use a HTTP proxy to access the outside world, enter the proxy information here. Otherwise, leave this blank.

The proxy information should be given in the standard form of "http://[[user][:pass]@]host[:port]/".

HTTP proxy information (blank for none):

<Go Back>

<Continue>

m) Configuring tasksel:

Select **No automatic updates**.

[!] Configuring x11-common

Applying updates on a frequent basis is an important part of keeping your system secure.

By default, updates need to be applied manually using package management tools. Alternatively, you can choose to have this system automatically download and install security updates, or you can choose to manage this system over the web as part of a group of systems using Canonical's Landscape service.

How do you want to manage upgrades on this system?

No automatic updates
Install security updates automatically
Manage system with Landscape

n) Software selection:

Select **OpenSSH server, DNS server and Manual package selection**.

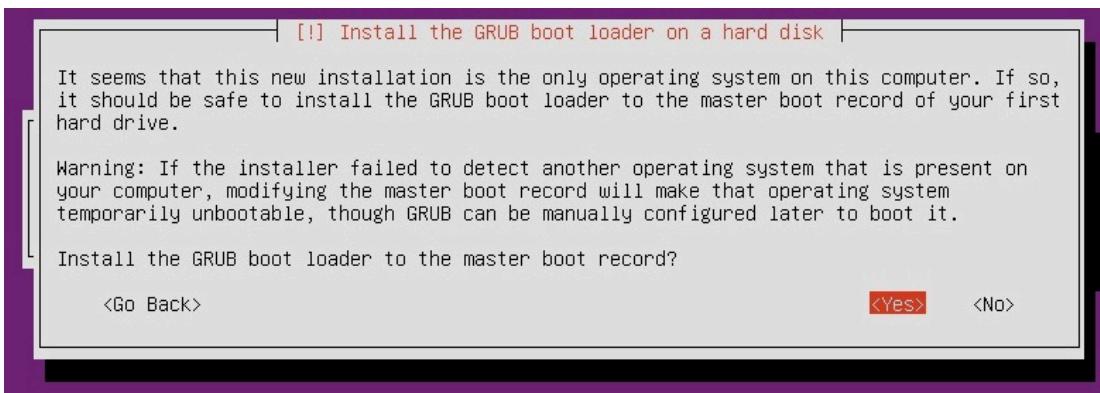
Notice: use **Space** to configure do not use **Enter**.

o) “Actions undo package resolver search options views help”:

Push ‘**q**’ and configure to **quit**.

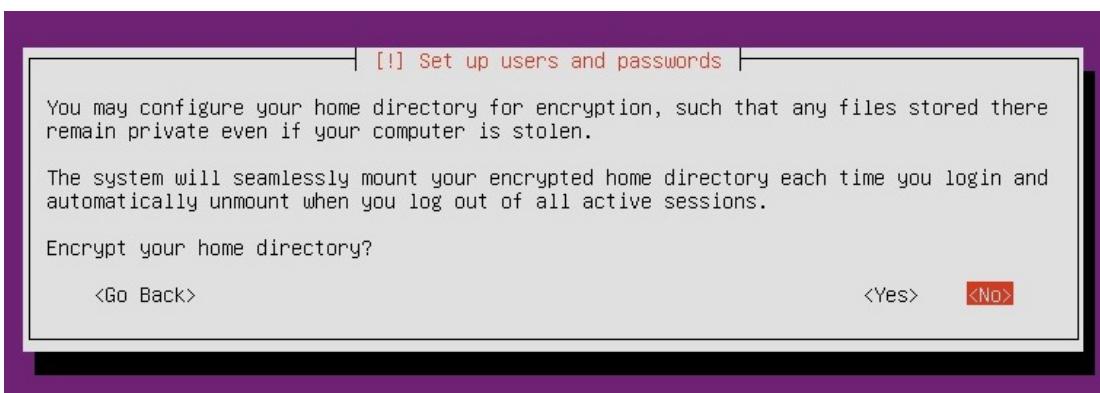
p) Install the GRUB boot loader on the hard disk:

Select yes.



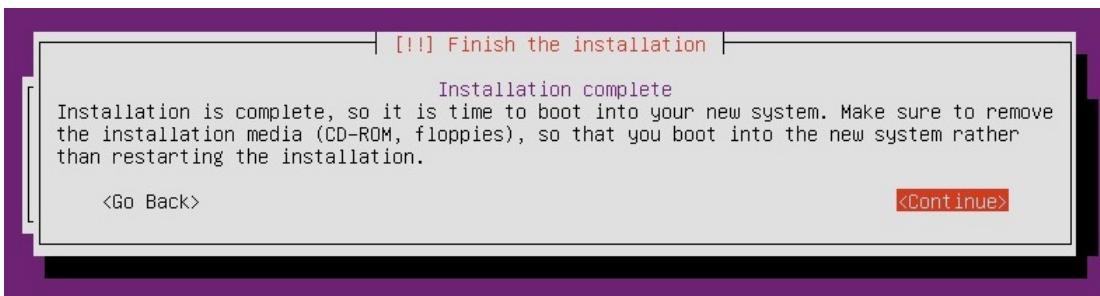
q) "Encrypt home directory":

Select no.



r) "Finish installation":

Select continue.



s) Finish all steps and shut down the computer:

input **sudo shutdown -h now** -> shut down the computer immediately(-h represents sleep).

```
* Documentation: https://help.ubuntu.com/

System information as of Tue Jun 23 10:45:46 CST 2015

[ 1597.790597] systemd-logind[1339]: Failed to start unit us
[ 1597.790664] systemd-logind[1339]: Failed to start user se
bigdata@shard1:~$ sudo shutdown -h now
```

input **sudo shutdown -r now** -> restart the computer immediately(-r represents reboot).

```
shard1 login: bigdata
Password:

Login incorrect
shard1.bigdata.com login: bigdata
Password:
Last login: Fri Jun 19 20:29:52 CST 2015 on ttym
Welcome to Ubuntu 14.10 (GNU/Linux 3.16.0-38-generic x86_64)

* Documentation: https://help.ubuntu.com/

System information as of Tue Jun 23 10:45:46 CST 2015

[ 1597.790597] systemd-logind[1339]: Failed to start unit user@1000.service: Unknown u
[ 1597.790664] systemd-logind[1339]: Failed to start user service: Unknown unit: user
bigdata@shard1:~$ sudo shutdown -r now
```

2. Cloning using Clonezilla

2.1 Introduction to Clonezilla

1. Definition of Clonezilla: Clonezilla is a partition and disk imaging/cloning program similar to True Image® or Norton Ghost®. It helps you to do system deployment, bare metal backup and recovery. Two types of Clonezilla are available, Clonezilla live and Clonezilla SE (server edition). Clonezilla live is suitable for single machine backup and restore. While Clonezilla SE is for massive deployment, it can clone many (40 plus!) computers simultaneously. Clonezilla saves and restores only used blocks in the hard-disk. This increases the clone efficiency. With some high-end hardware in a 42-node cluster, a multicast restoring at rate 8 GB/min was reported.

2. Website for downloading: <http://www.clonezilla.org/downloads.php>

Clonezilla Live Download

Live branch	Extra info	Other notes
alternative stable releases - 20150608-vivid	checksums , changelog , known issue , release note	Ubuntu-based, ?
stable releases - 2.4.2-10	checksums , changelog , known issue , release note	Debian-based, ?
alternative testing releases - 20150617-wily 20150617-vivid	checksums , changelog , known issue , release note	Ubuntu-based, ?
testing releases - 2.4.2-11	checksums , changelog , known issue , release note	Debian-based, ?

3. Installation using CD/DVD: Download an ISO file for CD/DVD. Then you can burn the ISO file to a CD/DVD with any burning program, such K3b on GNU/Linux or InfraRecorder on MS Windows, and remember to choose "Burn Image" to burn the ISO file on the CD. The CD can then be used to boot the machine you want to image or clone.

Notice: You need to clone at least two CDs: One for server, the other for clients.

2.2 Setting up Clonezilla for server

1. Installation steps:

- Insert the Clonezilla disk
- Reboot the computer and press F12 when it starts.
- When seeing the picture below

Select **CD/DVD ROM** and press **Enter**.

- Language selection:

Select **en_US.UTF-8 English**.

- Configuring console-data:

Select **Don't touch keymap** for the policy for handling keymaps.

- Start Clonezilla:

Select mode **Start Clonezilla Start Clonezilla** and press **Enter**

- g) Clonezilla-Opensource Clone Systems:
 - (i) Select mode **Device-device work directly from a disk or partition to a disk or partition** and press **Enter**
 - (ii) Select **Expert Expert mode: Choose your own options** and press **Enter**
 - (iii) Select **Disk to remote disk local disk to remote disk clone** and press **Enter**
- h) Clonezilla-Opensource Clone Systems | Mode:
Select **disk to remote disk** and Press **Enter**
- i) Clonezilla on the fly advanced extra parameters | Mode: disk to remote disk:
 - (i) Keep the advanced **default value** and Press **Enter**.
 - (ii) Select **Skip checking/repairing source file system** and Press **Enter**.
 - (iii) Select **Use the partition table from the source disk** and Press **Enter**.
- j) Waiting for the target machine to connect...
(Now for setting up clonezilla on the clients, please see 3.3 for details.)
- k) Choose mode:
Select **cmd Enter command line prompt** and Press **Enter**.
- l) Enter network mask for eth0:
Enter **255.255.255.0** and Press **Enter**.
- m) Reboot the computer

2.3 Setting up Clonezilla for client

1. Installation steps:

- a) Insert CD for the client computer:

When the instruction prompts waiting for the target machine to connect, change to the client computer, insert the CD for that computer

- b) Start up the computer from CD:

Start up the computer and press **F12**(for DELL) to enter **boot options**.

In the boot device menu:

Boot Device Menu

- 1. Normal
- 2. Diskette Drive
- 3. Hard-Disk Driver C :
- 4. IDE CD-ROM Device

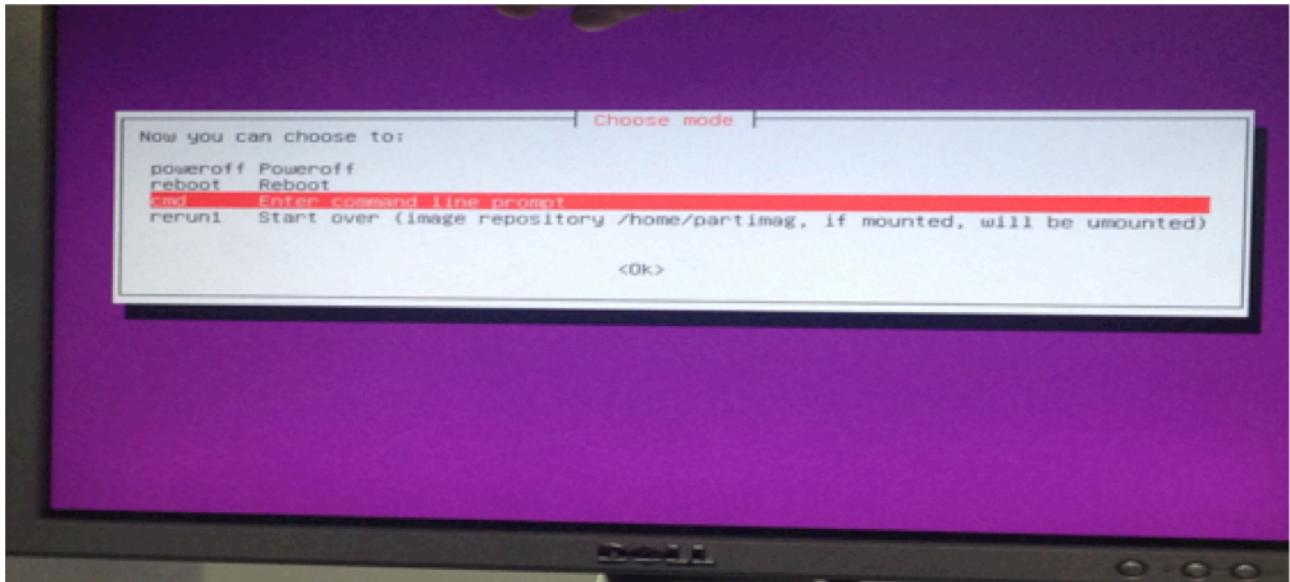
Select **4. IDE CD-ROM Device** to boot from CD

- c) Start Clonezilla on the client:

Select mode **Enter_shell Enter command line prompt** (which is different from the server) for the prompt **Start Clonezilla or enter login shell** (command line)?

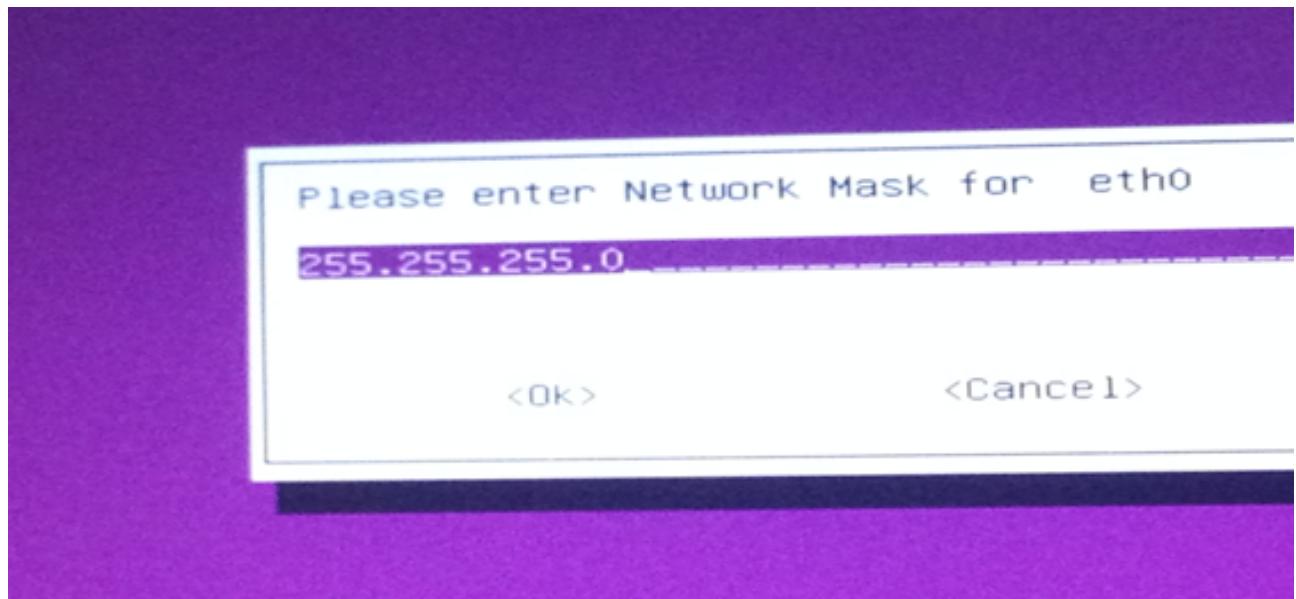
- d) Choose mode:

Select **cmd Enter command line prompt** and input: **sudo su -**



e) Network configuration:

- (i) Choose the mode to setup the network for this network card: eth0: Select **Static Use static IP address**
(ii) Set up network mask: Enter Network Mask for eth0: **255.255.255.0**



- (iii) Set up Default Gateway: Enter default gateway: **192.168.120.254**
(iv) Set up Nameserver: Enter Nameserver(s): **8.8.8.8**
(v) Input: **\$ ocs-onthefly -s 192.168.120.1 -t sda**
Notice: 192.168.120.1 is the IP address of the server computer.
sda is the disk partition which is going to be cloned.
- f) Reboot the computer:
Select the mode **Reboot** to reboot the computer.

3. Installation and configuration of MongoDB

3.1 Introduction

1. Definition of Sharding: Sharding is a method for storing data across multiple machines. MongoDB uses sharding to support deployments with very large data sets and high throughput operations.

2. This tutorial has been tested with following software versions:

Linux: Ubuntu14.10

MongoDB: mongodb-linux-x86_64-3.0.4

3.2 Installation of mongodb-linux-x86_64-3.0.4

1. Installation steps:

a) Download the binary files for the desired release of MongoDB:

Download the binaries from <https://www.mongodb.org/downloads>.

```
$ curl -O https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-3.0.4.tgz
```

b) Extract the files from the downloaded archive:

```
$ tar -zvxf mongodb-linux-x86_64-3.0.4.tgz
```

c) Copy the extracted archive to the target directory:

```
$ mkdir -p mongodb
```

```
$ cp -R -n mongodb-linux-x86_64-3.0.4/ mongodb
```

d) Ensure the location of the binaries is in the PATH variable:

```
$ export PATH=<mongodb-install-directory>/bin:$PATH
```

Replace <mongodb-install-directory> with the path to the extracted MongoDB archive.

3.3 Change the IP address for each computer

1. Channge teps:

a) Using vim change the IP address:

```
$ sudo vi /etc/network/interfaces
```

Change the document like this:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
```

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
    iface eth0 inet static
        address 10.6.2.2
        gateway 10.6.2.254
        netmask 255.255.255.0
        dns-nameservers 8.8.8.8
        broadcast 10.6.2.255
```

Notice:

1. Address: change IP address to whatever you like but make sure there is no same IP address in the same switch.
 2. Gateway: the first three must the same with IP address, the last one use 254.
 3. Netmask: 255.255.255.0
 4. Dns-nameservers: optional
 5. Broadcast: the first three must the same with address, the last one use 255.
- b) Save and exit the file, if want the config work, need reboot.

3.4 Change the file system of mongoldb into wiredTiger

1. Change steps:

- a) Create the direction and change the owner of the direction:

```
$ sudo mkdir -p /m01/mondata
$ sudo chown mongodb:mongodb /m01/mondata
$ sudo echo "storageEngine = wiredTiger" >> /etc/mongod.conf
```

- b) Modify mongod.conf:

```
$ sudo vi /etc/mongod.conf
```

After modification, the file should look like this:

```
storageEngine = wiredTiger
dbpath=/m01/mondata
```

- c) Modify /etc/security/limits.conf:

```
$ sudo vi /etc/security/limits.conf
```

add two lines below:

```
* soft nofile 64000
* hard nofile 64000
```

- d) Modify /etc/rc.local:

add the code below:

```
$ touch /var/lock/subsys/local
```

```
if test -f /sys/kernel/mm/transparent_hugepage/enabled; then  
    echo never > /sys/kernel/mm/transparent_hugepage/enabled  
fi  
if test -f /sys/kernel/mm/transparent_hugepage/defrag; then  
    echo never > /sys/kernel/mm/transparent_hugepage/defrag  
fi
```

e) Modify /etc/init/mongod.conf:

```
$ vi /etc/init/mongod.conf
```

add the code below after pre-start script:

```
if [ -f /sys/kernel/mm/transparent_hugepage/enabled ]; then  
    echo never > /sys/kernel/mm/transparent_hugepage/enabled  
fi  
if [ -f /sys/kernel/mm/transparent_hugepage/defrag ]; then  
    echo never > /sys/kernel/mm/transparent_hugepage/defrag  
fi
```

f) Check if it has succeeded with the instruction below:

```
$ echo "db.serverStatus()"| mongo|grep wiredTiger
```

It is successful if the command line shows like this:

```
“name” : “wiredTiger”  
“wiredTiger” : {
```

3.5 Configure Sharding Cluster of MongoDB

Notice:

Here we need 3 computers as server, 2 as Query Router, n as Shard.

Take 8 computers for example:

10.6.2.1	server1
10.6.2.2	server2
10.6.2.3	server3
10.6.2.4	router1
10.6.2.5	router2
10.6.2.6	shard1
10.6.2.7	shard2
10.6.2.8	shard3

1. Steps:

a) Configure 3 servers:

(i) Create direction in 3 servers:

```
$ mkdir -p /m01/mongo-metadata
```

(ii) Change the owner of the direction:

```
$ chown mongodb:mongodb /m01/mongo-metadata
```

(iii) Run the instruction in 3 servers (If it is not successful, check the permission):

server1:

```
$ mongod --configsvr --dbpath /m01/mongo-metadata --bind_ip 10.6.2.1  
--port 35001
```

server2:

```
$ mongod --configsvr --dbpath /m01/mongo-metadata --bind_ip 10.6.2.1  
--port 35001
```

server3:

```
$ mongod --configsvr --dbpath /m01/mongo-metadata --bind_ip 10.6.2.1  
--port 35001
```

b) Configure 2 query routers:

(i) Make sure that the service of mongod has been colsed (use \$kill -15).

(ii) Run the instructions in 2 routers.

```
$ su
```

```
$ mongos —configdb 10.6.2.1:35001,10.6.2.2:35001,10.6.2.3:35001 #ip  
of servers
```

c) Add shards to the cluster:

(i) Make sure that the service of mongod has been colsed (use \$kill -15).

(ii) Create direction in 3 shards:

```
$ mkdir -p /m01/mongo-data
```

(iii) Change the owner of the direction:

```
$ chown mongodb:mongodb /m01/mongo-data
```

(iv) Run the instruction in 3 shards (If it is not successful, check the permission):

shard1:

```
$ mongod --dbpath /m01/mongo-data --bind_ip 10.6.2.6 --port 35001  
--storageEngine wiredTiger
```

shard2:

```
$ mongod --dbpath /m01/mongo-data --bind_ip 10.6.2.7 --port 35001  
--storageEngine wiredTiger
```

shard3:

```
$ mongod --dbpath /m01/mongo-data --bind_ip 10.6.2.8 --port 35001  
--storageEngine wiredTiger
```

(v) Add shards to the cluster with the instructions below:

```
$ sh.addShard( "10.6.2.6:35001" )
```

```
$ sh.addShard( "10.6.2.6:35001" )
```

```
$ sh.
```

4. MPI

4.1 MPI (Message Passing Interface) Introduction

Message Passing Interface (MPI) is a standardized and portable message-passing system designed by a group of researchers from academia and industry to function on a wide variety of parallel computers. The standard defines the syntax and semantics of a core of library routines useful to a wide range of users writing portable message-passing programs in different computer programming languages such as Fortran, C, C++ and Java. There are several well-tested and efficient implementations of MPI, including some that are free or in the public domain. These fostered the development of a parallel software industry, and encouraged development of portable and scalable large-scale parallel applications.

In 2015 summer internship, one machine served as a server and 15 machines served as clients. The following is the instruction both used for the server and clients.

4.2 MPI Setting Up Instruction

4.2.1 Both server and client should do

1. Create a shared directory (named mirror)

```
$ mkdir -p /home/bigdata/mirror
```

2. You should change the IP address after installing all packages, then, connect your computer to the switch.

3. You could use command *ifconfig* to check your computer's IP

4.2.2 Only server should do

1. Install NFS server

```
$ sudo apt-get install nfs-kernel-server
```

2. Configure the mirror directory (make it public and could be accessed by all clients after amounting)

```
$ sudo chown nobody:nogroup mirror  
$ sudo vi /etc/exports  
$ add /home/bigdata/mirror *(rw,sync)  
$ sudo exportfs -a
```

3. Start NFS server

```
$ sudo /etc/init.d/nfs-kernel-server start
```

4. Install mpich2

a) Create a directory inside the mirror directory (mpich2):

```
$ mkdir /home/bigdata/mirror/mpich2  
$ cd /home/bigdata/mirror/mpich2
```

b) Download the package:

```
$ sudo wget http://www.mpich.org/static/downloads/3.1/mpich-3.1.tar.gz
```

c) Unzip the package

```
$ sudo tar xvzf mpich-3.1.tar.gz(filename)
```

d) Configuration

```
$ cd mpich-3.1  
$ sudo ./configure --prefix=/home/bigdata/mirror/mpich2/ --disable-f77  
--disable-fc  
$ sudo make  
$ sudo make install
```

e) Add path

```
$ export PATH=/home/bigdata/mirror/mpich2/bin:$PATH  
$ export LD_LIBRARY_PATH=~/mirror/mpich2/lib:$LD_LIBRARY_PATH  
$ export LD_LIBRARY_PATH
```

f) Define MPICH install path

```
$ sudo vi /etc/environment
```

In the /etc/environment file:

Add /home/bigdata/mirror/mpich2/bin to path

5. Set IP address

```
$ sudo vi /etc/network/interfaces
```

Change the content:

iface eth0 inet static(change the dhcp to static)

Add the following few lines:

address 10.6.2.x (x is different for each client)

netmask 255.255.255.0

network 10.6.2.0

broadcast 10.6.2.255

gateway 192.168.1.1 (gateway maybe different on different machines)

6. After adding all these lines, save the file and reboot the computer

7. Add all clients' IP address to the server(after clients have changed their IP address)

```
$ sudo vi /home/bigdata/mirror/hosts
```

adding all clients' IP address to the file, one per line

E.g. 10.6.2.10

10.6.2.11

10.6.2.12

.....

8. Configure the ssh

```
$ cd /home/bigdata/.ssh
```

a) Create a public key

```
$ ssh-keygen -t dsa
```

Notice:

1. Do not use sudo, or it will not work.

2. There will appear several prompts:

- Generating public/private das key pair
- Enter file in which to save the key (/home/bigdata/.ssh/id_dsa): id_dsa
- [just press ENTER for the following two prompts]

3. After this, there will be two files in the .ssh directory:

id_das(private key), id_das.pub(public key)

- b) Copy the id_das.pub to all clients using ssh protocol:

```
$ scp id_das.pub bigdata@10.6.2.x:/home/bigdata/.ssh/
```

Notice:

1. 10.6.2.x is the IP address of the client;
2. Execute this command for all clients, one at a time;
3. This command copies the id_das.pub to the .ssh directory of the clients.

4.2.3 Only clients should do

1. Install NFS client

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

2. Set IP address

```
$ sudo vi /etc/network/interfaces
```

Change the content:

iface eth0 inet static(change the dhcp to static)

Add the following few lines:

address 10.6.2.x (x is different for each client)

netmask 255.255.255.0

network 10.6.2.0

broadcast 10.6.2.255

gateway 192.168.1.1 (gateway maybe different on different machines)

3. Configure the ssh

- a) Authorize the public key from the server

```
$ cd /home/bigdata/.ssh
```

```
$ cat id_das.pub >> authorized_keys
```

- b) ssh to the server:

```
$ ssh bigdata@10.6.2.y (the server IP)
```

(Enter yes)

- c) Test whether the public key works or not(still in the server right now)

```
$ ssh bigdata@10.6.2.x (your client IP)
```

(If you could go into the client without requiring for the password of the client,
it works now!)

- d) Come back the client:

```
$ exit (come back to server)
```

```
$ exit (come back to your client)
```

4. Amount to server

```
$ sudo mount 10.6.2.x:/home/bigdata/mirror ~/mirror
```

```
$ df -h (Check if mounted)
```

(Now, you can access the public folder mirror, it is shared by the server and all
clients)

4.2.4 MPI compiling and running

1. Compiling

```
$ mpicc fileName.c -o filename
```

2. Running

```
$ mpiexec -f hosts -n numberOfCore ./fileName
```

(Here, the *numberOfCore* is the number of cores you want to use among the server and clients)

3. Checking

```
$ htop
```

(If the top 2 lines are 100%, it means the cores on this machine are being used)