คู่มือการติดตั้ง
Raspberry Pi Cluster

การพัฒนา HPC Cluster ขนาดเล็กด้วย Raspberry Pi สำหรับการศึกษา

Small-scale HPC Cluster with Raspberry Pi : An Educational Kit

(Draft Version – 14 Feb 2022)

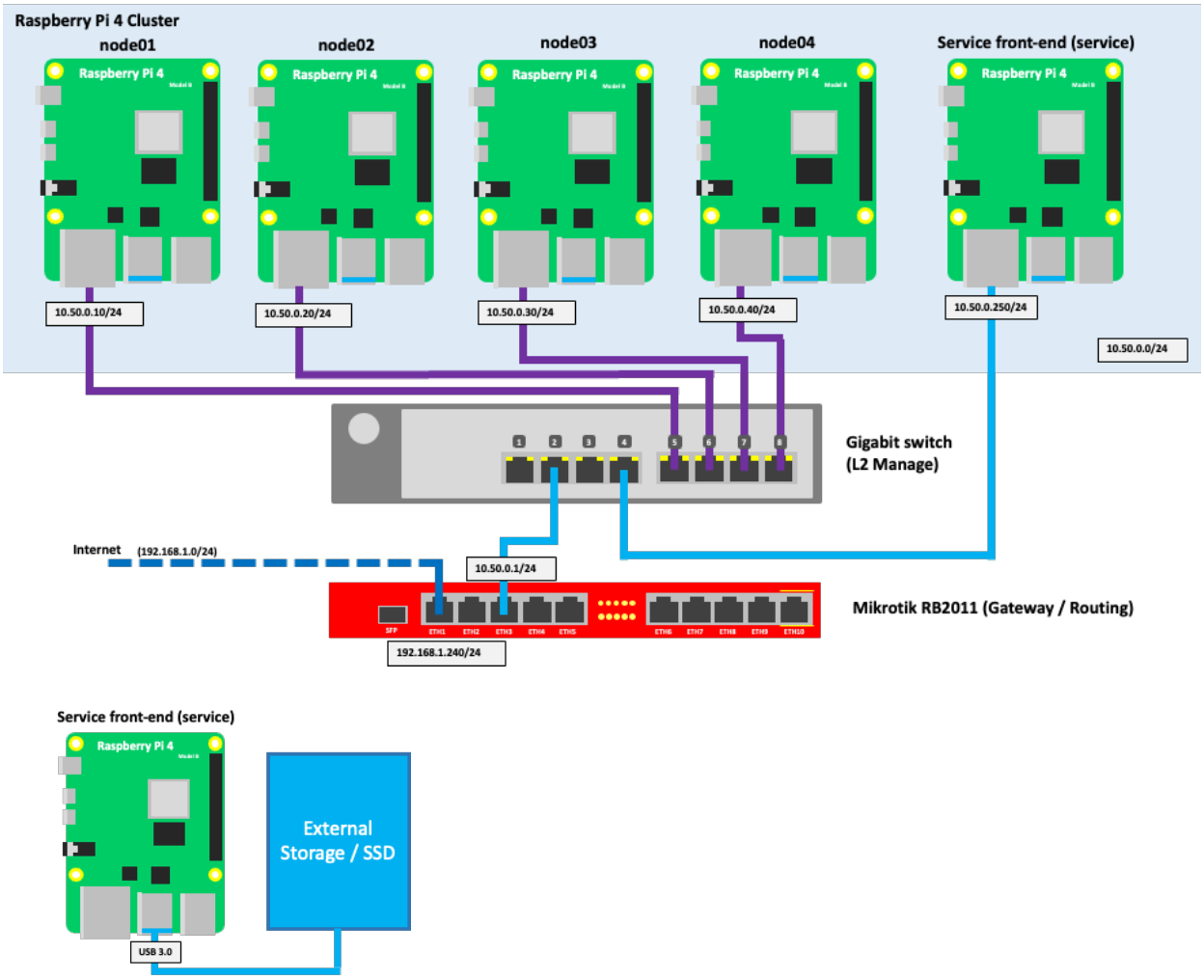
การติดตั้งและซอฟต์แวร์ที่เกี่ยวข้องกับ Cluster

ขั้นตอนที่ 0 : อุปกรณ์ฮาร์ดแวร์ที่เกี่ยวข้อง

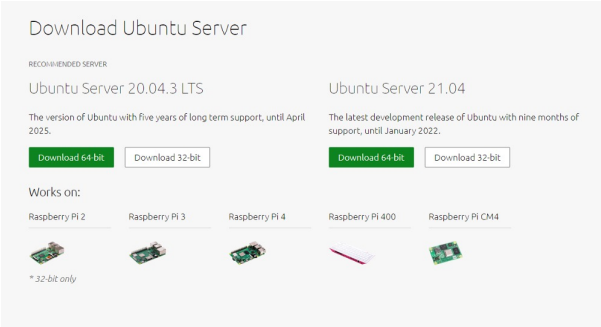
No.	รายการ	จำนวน
1.	Raspberry Pi 4 Model B (8 GB)	2
2.	Raspberry Pi 4 Model B (4 GB)	3
3.	MicroSD Card (32 GB)	5
4.	USB Type C power cables	4
5.	USB Type C power adapter	1
6.	6 Port USB power station adapter	1
7.	SSD Storage (120 GB) + USB 3.0 Enclosure	1
8.	8 Port 10/100/1000 Gigabit Switch	1
9.	Router Mikrotik RB2011 (Optional)	1

ขั้นตอนที่ 1 : การติดตั้งฮาร์ดแวร์

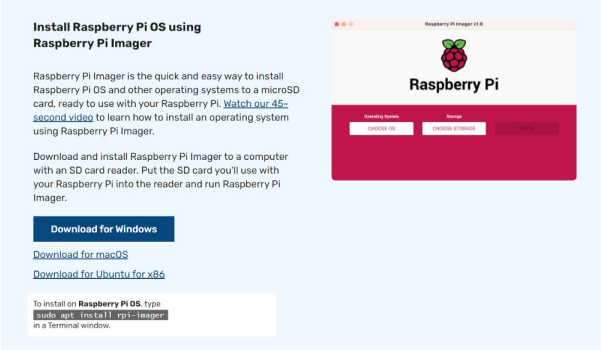
การออกแบบและการเชื่อมต่อ



ขั้นตอนที่ 2 : การติดตั้งระบบปฏิบัติการ (OS) ให้กับบอร์ด Raspberry Pi



1. ดาวน์โหลดไฟล์ระบบปฏิบัติการ Ubuntu Server 20.04 LTS (arm) 64 bit ได้จาก https://ubuntu.com/download/raspberry-pi โดยเมื่อทำการดาวน์โหลดเสร็จสิ้นจะได้ไฟล์ที่ชื่อว่า ubuntu-20.04.x-reinstalled-server-arm64+raspi.iso เป็นอันเรียบร้อย



2. ต่อไปทำดาวน์โหลดและติดตั้งโปรแกรม Raspberry Pi Imager ได้จาก Raspberry Pi Imager – Raspberry Pi ซึ่งเป็นโปรแกรมในการเขียน BOOT ลงไปใน MicroSD Card

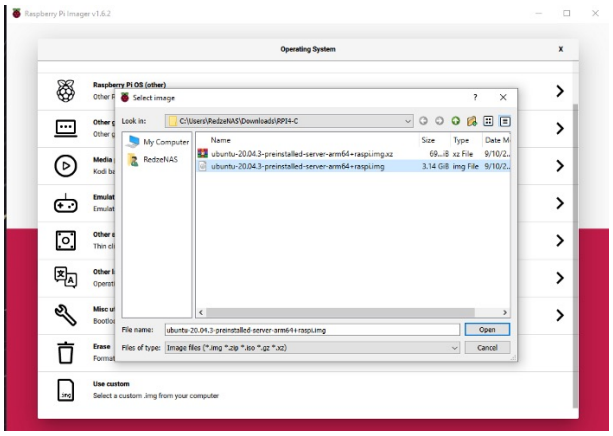
3. นำ MicroSD Card ใส่ใน USB Card reader แล้วนำไปเสียบที่ช่อง USB บนคอมพิวเตอร์หรือแล็ปท็อป

4. จากนั้นเปิดโปรแกรม Raspberry Pi Imager แล้วทำการตั้งค่าดังนี้ต่อไปนี้

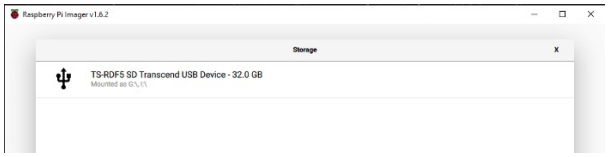
Operating System > CHOOSE OS > เลื่อนลงมาด้านล่างสุด แล้วเลือก **Use custom** > จากนั้นให้ท่านเลือกไฟล์ Image ที่ทำการแตกไฟล์เอาไว้แล้ว (.iso) > เมื่อเลือกเสร็จกดปุ่ม **open**

Storage > **CHOOSE STORAGE** > จะปรากฏ USB Card readers ของท่านขึ้นมาก็ให้ท่านเลือก

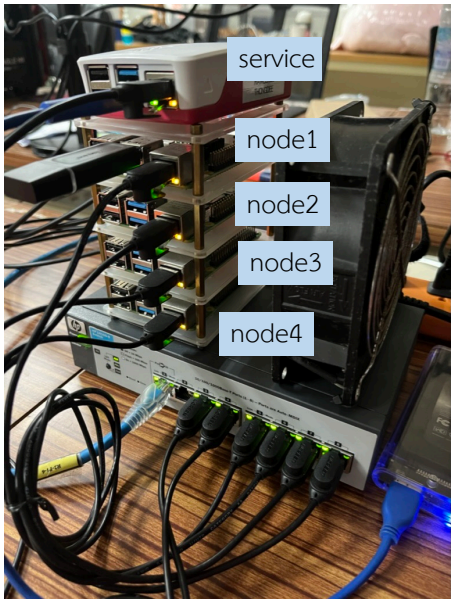
<u>ข้อควรระวัง :</u> กรณีที่มารเสียบ USB Storage อื่นร่วมด้วยกรุณาตรวจสอบก่อนการเลือกด้วยว่า USB Storage นั้นเป็นของ MicroSD Card หรือไม่ เพราะในขั้นตอนถัดไปจะมีการ Format ข้อมูล ข้อมูลอาจจะสูญหายได้



5. เมื่อทำการเลือกทั้ง OS และ Storage เสร็จ แล้วให้ท่านกดปุ่ม **WRITE** จะเป็นการ burn OS image ลง MicroSD card ซึ่งตัวโปรแกรมจะทำการสร้างไฟล์ Boot และไฟล์ config ต่างๆ รวมไปถึงการแบ่ง partition ตัว MicroSD card ของเราด้วย โดยในขั้นตอนนี้ให้ท่านรอสักครู่นึง — จนเสร็จสิ้น



ขั้นตอนที่ 3 : การติดตั้งและตั้งค่าเครือข่าย (Networking) ให้กับบอร์ด Raspberry Pi



รายละเอียดการกำหนด IP Address

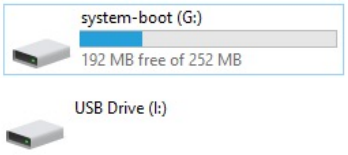
Hostname	IP Address	Netmask	Gateway	Interface
node1	10.50.0.10	255.255.255.0	10.50.0.1	eth0
node2	10.50.0.20	255.255.255.0	10.50.0.1	eth0
node3	10.50.0.30	255.255.255.0	10.50.0.1	eth0
node4	10.50.0.40	255.255.255.0	10.50.0.1	eth0
service	10.50.0.250	255.255.255.0	10.50.0.1	eth0
Mikrotik (Router)	10.50.0.1	255.255.255.0	-	eth3

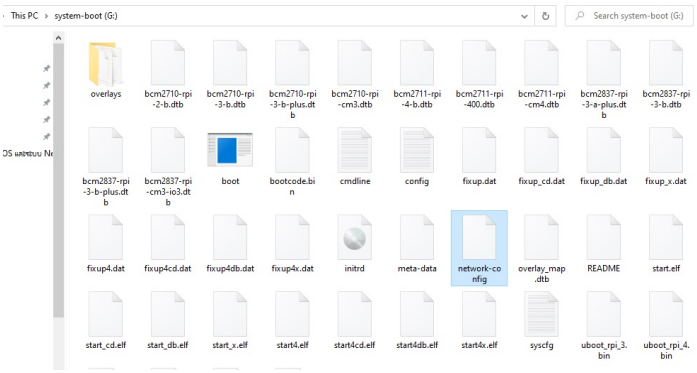
สำหรับการกำหนดที่อยู่เครือข่าย หรือ IP Address ให้กับบอร์ด Raspberry Pi นั้นสามารถทำได้อยู่ 3 วิธี คือ

1. กำหนด IP Address ลงไปในไฟล์ network-config โดยไฟล์นี้จะอยู่ใน partition system-boot ใน MicroSD Card ที่เรา burn เรียบร้อยแล้ว
2. ให้ Mikrotik หรือ Router แจก DHCP (Dynamic Host Configuration Protocol) แล้ว Mark Static ใน IP > DHCP Server > Leases (วิธีนี้สำหรับผู้ที่ใช้งาน Mikrotik หรือมีพื้นฐานมาบ้าง หรือ กรณีที่มี Router ที่แจก DHCP ให้อยู่แล้ว)
3. เชื่อมต่อจอภาพ เมาส์ คีย์บอร์ด กับบอร์ด Raspberry Pi จากนั้นจึงกำหนด IP Address โดยใช้ netplan ตามปกติ

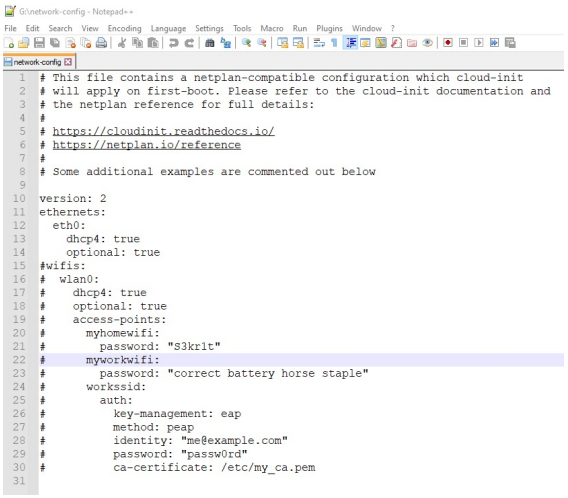
ขั้นตอนที่ 3.1 : กำหนด IP Address ลงไปในไฟล์ network-config

1.เริ่มแรกก็ให้ท่านเสียบ MicroSD Card ที่ทำการ burn เรียบร้อยแล้ว ของบอร์ด Raspberry Pi ที่ต้องการ เข้ากับ PC หรือ Notebook
2.จากนั้นให้เข้าไปที่ partition ที่ชื่อว่า system-boot





3. ขั้นตอนต่อไปให้ท่านทำการหาไฟล์ที่ชื่อว่า **network-config** แล้วให้ใช้โปรแกรมพวก Text editor เช่น notepad , notepad++ ในการเปิดไฟล์ จะได้หน้าตาแบบนี้



4. ทำการปิดการรับ DHCP และกำหนด Static IP Address / Subnet / Gateway และ DNS Server ลงไปตามตัวอย่างด้านล่างนี้ครับ

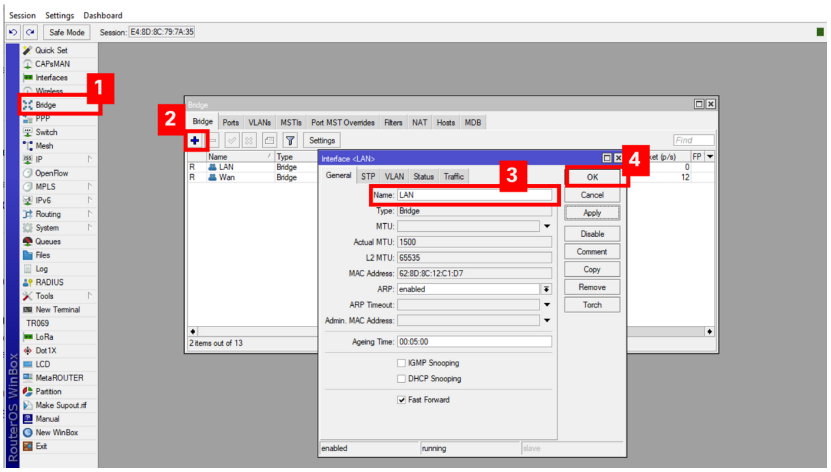
```
version: 2
  eth0:
    addresses:
      - 10.50.0.10/24
    gateway4: 10.50.0.1
    nameservers:
      addresses: [8.8.8.8, 8.8.4.4]
    optional: true
```

```
version: 2
  eth0:
    addresses:
      - 10.50.0.10/24
    gateway4: 10.50.0.1
    nameservers:
      addresses: [8.8.8.8, 8.8.4.4]
    optional: true
```

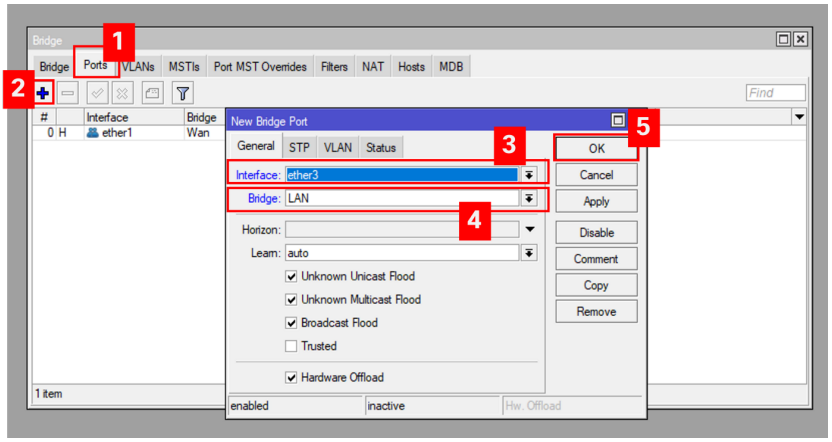
5. ทำการ Save การแก้ไขไฟล์ **network-config** จากนั้นทำขั้นตอนที่ 1 – 4 อีกครั้งแบบนี้กับ MicroSD Card ที่จะนำไปใส่ใน Raspberry Pi Cluster ของเราทุกตัว เพียงแค่นี้ Raspberry Pi ก็จะมีการถูกกำหนด IP Address เรียบร้อย

<u>ขั้นตอนที่ 3.2 : กำหนดให้ Mikrotik หรือ Router แจก DHCP ให้กับ Raspberry Pi ใน Cluster</u>

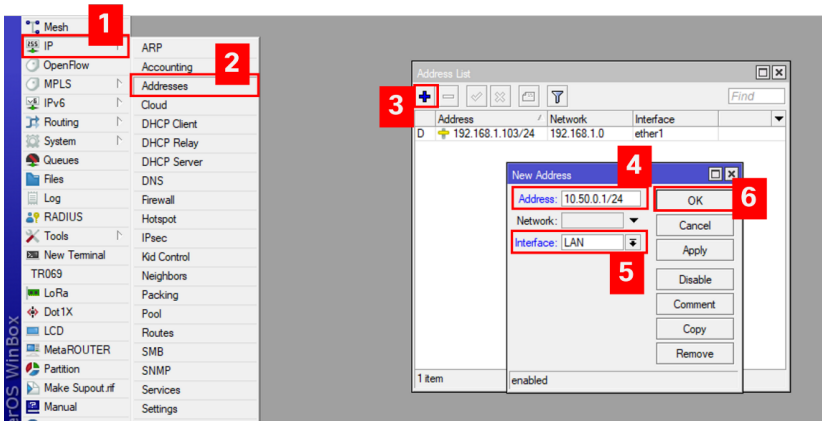
1.เริ่มต้นจากการเข้าโปรแกรม **Winbox** แล้ว Login เข้าไปในตัว Mikrotik ของเราครับ



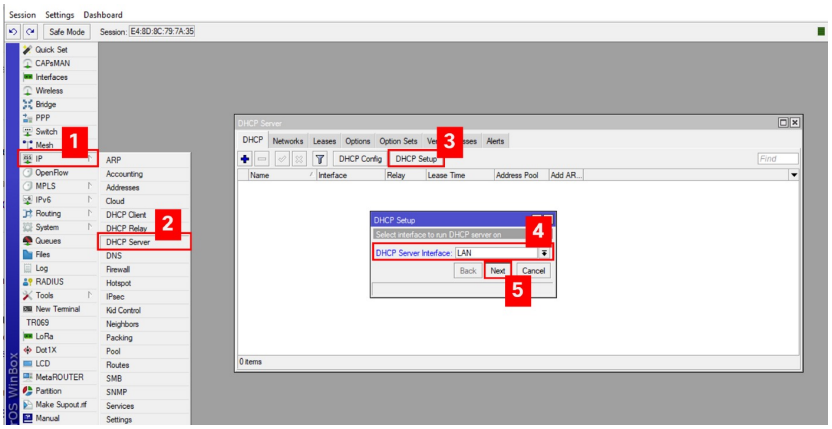
2. จากนั้นให้ท่านสร้าง Bridge ของ Lan โดยให้ไปที่เมนู Bridge(1) > ในแท็บ Bridge ในคลิก + (2) > จากนั้นในช่อง Name (3)ในใส่เป็นชื่อที่ต้องการ ผมใส่เป็น LAN > สุดท้ายกด OK (4)



3. จากนั้นเราจะเพิ่ม Interface ลงไปใน Bridge โดยเริ่มจากไปที่แท็บ Ports(1) > แล้วกดที่เครื่องหมาย + (2) > เสร็จแล้วให้เลือก Interface (3) ที่เราต้องการจะเพิ่มลงใน Bridge (ถ้าตาม Diagram ของผมคือ ether3) > ส่วน Bridge (4) ก็ทำการเลือก Bridge ที่เราต้องการ หรือที่เราสร้างในขั้นตอนที่ 2. ของผมชื่อว่า LAN > สุดท้ายก็กด OK (5)

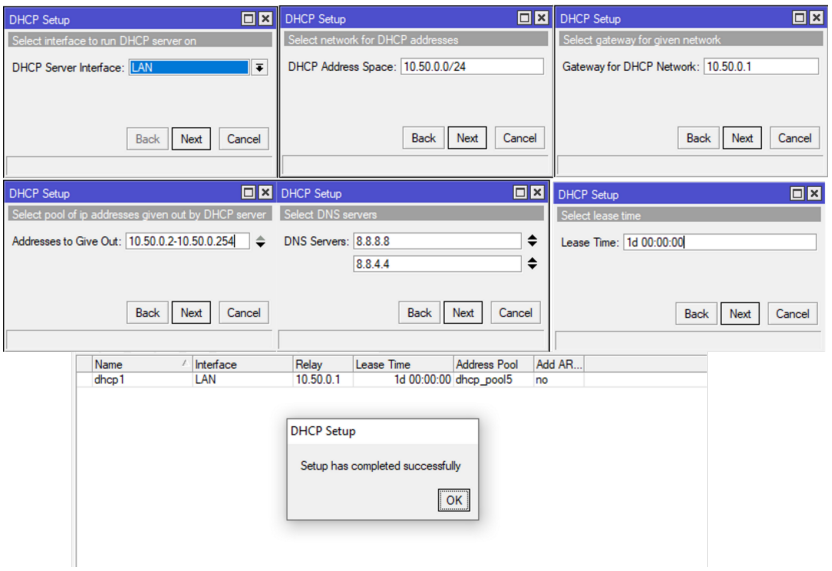


4. ต่อไปเราจะไปกำหนด Subnet และ IP Address ให้กับ Bridge ของเรากันครับ โดยไปที่เมนู IP (1) > Addresses (2) > คลิกที่เครื่องหมาย **+** (3) > จากนั้นในช่อง Address : (4) ให้เราทำการกำหนดเป็น IP Address ของ Bridge นี้ผมกำหนดเป็น 10.50.0.1/24 > จากนั้นเลือก Interface (5) ที่ต้องการกำหนด IP Address ให้ ของคือ LAN > จากนั้นกด OK (6)

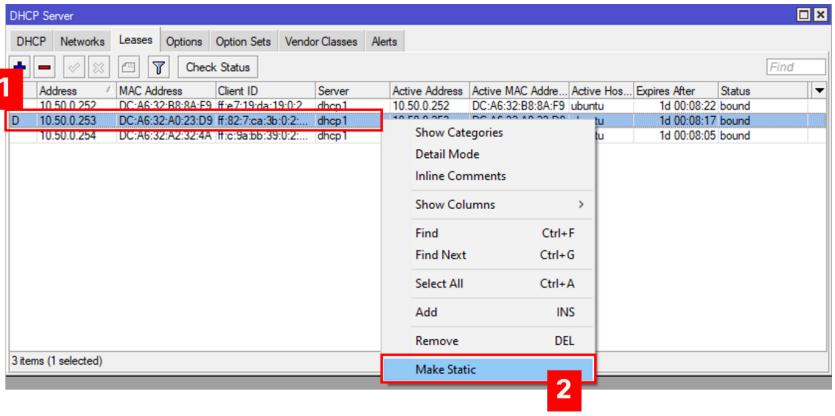


5. ต่อไปเราจะเริ่มสร้างตัว DHCP Server กันครับ โดยเริ่มจากไปที่เมนู IP (1) > DHCP Server (2) > จากนั้นให้กด **DHCP Setup** (3) จากนั้นก็กำหนดค่าต่าง ๆ ดังนี้

- **DHCP Server Interface** : ให้ท่านเลือก interface ที่ต้องการจะทำการแจก IP ในที่นี้จะแจกเข้าไปใน Bridge ที่ชื่อ LAN ที่มี interface ether3 ของเราอยู่ใน Bridge จากนั้นกด Next (5)
- **DHCP Address Space** : ให้ท่านกำหนด Subnet ลงไปของผมจะเป็น 10.50.0.0/24
- **Gateway for DHCP Network :** กำหนด IP Address ที่ต้องการให้เป็น Gateway ของ DHCP นี้ โดยผมจะเลือกใช้เป็น 10.50.0.1 (IP เบอร์แรกของ Subnet หรือ IP Address ของ Bridge ที่เรากำหนดไว้ในขั้นตอนที่ 4 นั่นเอง)
- **Addresses to Give Out :** กำหนดช่วง IP Address ที่ต้องการจะให้ DHCP แจกให้
- **DNS Server :** กำหนด DNS Server ที่ต้องการแจก ผมใช้ Public DNS ของ Google คือ 8.8.8.8 กับ 8.8.4.4
- **Lease Time :** กำหนดเวลาหมดอายุของ IP Address ผมกำหนดเป็น 1d 00:00:00 หรือ 1 วัน



6. จากนั้นกด Next ครั้งสุดท้ายจะปรากฏข้อความว่า “ Setup has completed successfully” เป็นอันเสร็จสิ้นการสร้าง DHCP Server บน Mikrotik



7. วิธีตรวจสอบว่า DHCP Server ของเราทำงานและแจก IP Address ให้กับบอร์ด Raspberry Pi ของเราไหม สามารถตรวจสอบได้จากแท็บ Leases ก็จะปรากฏให้เราเห็นว่าแจก IP Address ใดไปบ้าง และเราก็ยังสามารถกำหนดให้เป็นการแจก IP เดิมให้กับอุปกรณ์เสมือนเป็น Static ได้โดยการเลือก IP Address ที่ต้องการ (1) > คลิกขวาแล้วเลือก Make Static (2)

ขั้นตอนที่ 4 : ใช้งาน Raspberry Pi ผ่าน SSH

1. ในการเริ่มต้นใช้งานบอร์ด Raspberry Pi เราจะทำการใช้งานในรูปแบบของ Command Line Interface (CLI) ผ่านทาง Secure Shell (SSH)
2. ให้ท่านเรียกใช้งาน **PowerShell (Windows)** , Terminal (macOS, Linux) หรือ โปรแกรม Putty , MobaXTerm หรืออื่น ๆ ที่สามารถใช้งาน SSH ได้
3. ให้เรียกใช้งานส่วนของ SSH โดยพิมพ์ ssh username@destination_ip โดย username เริ่มต้นของ Ubuntu server 20.04 คือ **ubuntu** จากนั้นตามด้วย IP Address ของบอร์ด Raspberry Pi ที่ต้องการใช้งาน จะได้เป็น **ssh ubuntu@10.50.0.xx** จากนั้นกด **Enter**

```
PS C:\Users\RedzeNAS> ssh ubuntu@10.50.0.20
```

4. โดยในการเชื่อมต่อบอร์ด Raspberry Pi ผ่าน SSH ในครั้งแรกเครื่องของ Host จะมีการเก็บ fingerprint ให้ทำการพิมพ์ **yes** และกด **Enter**

```
PS C:\Users\RedzeNAS> ssh ubuntu@10.50.0.20
The authenticity of host '10.50.0.20 (10.50.0.20)' can't be established.
ECDSA key fingerprint is SHA256:WSG3ISd9Oi4PLYDOHLeEFP9w/N3LPMr5uURgftmaC5s.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

5. จะมีการถามหารหัสผ่านให้ท่านพิมพ์ **ubuntu** โดยในขณะพิมพ์จะไม่ปรากฏให้เห็น เมื่อพิมพ์เสร็จให้กด Enter **(Default : ubuntu / ubuntu)**

```
PS C:\Users\RedzeNAS> ssh ubuntu@10.50.0.20
The authenticity of host '10.50.0.20 (10.50.0.20)' can't be established.
ECDSA key fingerprint is SHA256:WSG3ISd9Oi4PLYDOHLeEFP9w/N3LPMr5uURgftmaC5s.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.50.0.20' (ECDSA) to the list of known hosts.
ubuntu@10.50.0.20's password:
```

6. ท่านจะสามารถเข้ามาที่บอร์ด Raspberry Pi ได้และตัว OS จะให้ท่านทำการเปลี่ยนรหัสผ่านทันทีโดยในช่อง Current password: ให้ท่านพิมพ์ **ubuntu** แล้วกด **enter**

```
WARNING: Your password has expired.
You must change your password now and login again!
Changing password for ubuntu.
Current password:
```

7. จะปรากฏช่อง New password: ให้ท่านตั้งรหัสผ่านใหม่ จากนั้นกด Enter จะปรากฏให้กรอกรหัสผ่านอีกครั้งเพื่อยืนยันการเปลี่ยนแปลง

```
WARNING: Your password has expired.
You must change your password now and login again!
Changing password for ubuntu.
Current password:
New password:
Retype new password:
```

8. เมื่อรหัสผ่านเปลี่ยนแปลงเรียบร้อยแล้วจะมีข้อความ **passwd: password updated successfully** ปรากฏเป็นอันเรียบร้อย
9. จากนั้น SSH ที่เชื่อมต่ออยู่จะหลุดไป ให้ท่านเชื่อมต่อใหม่โดยใช้ username คือ ubuntu และ password คือ รหัสที่ท่านได้ตั้งค่าไป

```
ubuntu@node01: ~
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Jan  5 15:15:29 UTC 2022

  System load:  0.02               Temperature:           67.2 C
  Usage of /:   10.0% of 29.05GB   Processes:             141
  Memory usage: 4%                 Users logged in:       1
  Swap usage:   0%                 IPv4 address for eth0: 10.50.0.10

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

35 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


*** System restart required ***
Last login: Wed Jan  5 14:54:24 2022 from 10.50.0.249
ubuntu@node01:~$
```

10. เมื่อเชื่อมต่อ SSH กลับมาแล้วให้ท่านเปลี่ยนชื่อ hostname ของบอร์ด Raspberry Pi ให้เป็นตามตาราง ในขั้นตอนที่ 3 โดยใช้คำสั่งดังต่อไปนี้

```
# sudo hostnamectl set-hostname node1
# sudo reboot
```

เปลี่ยนจาก [node1] เป็น [node2] , [node3] , [node4] , [service] ตามจำนวนเครื่องใน cluster ทั้งหมด

ขั้นตอนที่ 5 : ติดตั้งและตั้งค่าการใช้งาน NFS (Share Storage)

ขั้นตอนที่ 5.1 : เตรียม Storage Drive และ mount directory

1. เชื่อมต่อ SSD Storage หรือ USB Storage ที่ช่อง USB ของบอร์ด Raspberry Pi ที่จะใช้เป็น NFS Server ในที่นี้คือเครื่อง [service]
2. ทำการ SSH เข้าไปที่บอร์ด Raspberry Pi โดย **ssh ubuntu@10.50.0.250**

```
ubuntu@service:~$ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
loop0         7:0    0    49M  1 loop /snap/core18/2252
loop1         7:1    0  57.5M  1 loop /snap/core20/1274
loop2         7:2    0  57.5M  1 loop /snap/core20/1332
loop3         7:3    0    49M  1 loop /snap/core18/2289
loop4         7:4    0  60.7M  1 loop /snap/lxd/21843
loop5         7:5    0    62M  1 loop /snap/lxd/21032
loop6         7:6    0  37.5M  1 loop /snap/snapd/14296
loop7         7:7    0  37.6M  1 loop /snap/snapd/14553
sda           8:0    0 119.2G  0 disk
└sda1         8:1    0 119.2G  0 part
mmcblk0     179:0    0  29.8G  0 disk
├mmcblk0p1 179:1    0   256M  0 part /boot/firmware
└mmcblk0p2 179:2    0  29.6G  0 part /
```

3. ตรวจสอบว่า SSD Storage หรือ USB Storage ที่เชื่อมต่อเข้าบนบอร์ด Raspberry Pi มีชื่อว่าอะไร เพื่อเราจะได้เรียกใช้งานต่อไป โดยใช้คำสั่งสำหรับการตรวจสอบดังนี้

```
# lsblk
```

ในกรณีนี้ SSD Storage หรือ USB Storage ถูก mount อยู่ที่ /dev/sda1

```
ubuntu@service:~$
ubuntu@service:~$ sudo mkfs.ext4 /dev/sda1
mke2fs 1.45.5 (07-Jan-2020)
/dev/sda1 contains a ext4 file system
        last mounted on /clusterfs-ssd on Mon Jan 10 04:56:43 2022
Proceed anyway? (y,N) N
Creating filesystem with 7562256 4k blocks and 1892352 inodes
Filesystem UUID: 4a60d54c-6865-4357-b408-8aa26a434a84
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks):
done
Writing superblocks and filesystem accounting information:
done
```

4. ทำการ Format SSD Storage หรือ USB Storage ดังกล่าวให้อยู่ในระบบไฟล์รูปแบบ EXT4 เพื่อที่ตัว OS และ NFS เรียกใช้งานพื้นที่ดังกล่าวได้ โดยใช้คำสั่ง

```
# sudo mkfs.ext4 /dev/sda1
```

```
ubuntu@service:~$ mkdir /clusterfs-ssd
ubuntu@service:~$ sudo chown nobody.nogroup -R /clusterfs-ssd
ubuntu@service:~$ sudo chmod 777 -R /clusterfs-ssd
```

```
ubuntu@service:~$ cd /
ubuntu@service:/$ ls
bin   clusterfs-ssd  etc   lib         media  opt   root  sbin  srv  tmp  var
boot  dev            home  lost+found  mnt    proc  run   snap  sys  usr
ubuntu@service:/$
```

5. ทำการสร้าง mount directory โดย mount directory ของทุก node ใน cluster ควรจะมีชื่อที่เหมือนกัน โดยในที่นี้จะใช้ชื่อว่า /clusterfs-ssd โดยใช้คำสั่งดังต่อไปนี้ในการสร้าง

```
# sudo mkdir /clusterfs-ssd
# sudo chown nobody.nogroup -R /clusterfs-ssd
# sudo chmod 777 -R /clusterfs-ssd
```

6. ตั้งค่าให้ mount ตัว Storage อัตโนมัติโดยเริ่มจากหา UUID ของ Storage ที่ต้องการจะ mount โดยใช้คำสั่งตรวจสอบดังนี้

```
# blkid
```

```
ubuntu@service:/$ blkid
/dev/mmcblk0p1: LABEL_FATBOOT="system-boot" LABEL="system-boot" UUID="5496-E6C8" TYPE="vfat" PARTUUID="f66f0719-01"
/dev/mmcblk0p2: LABEL="writable" UUID="675ba907-3741-428c-afa4-c00f1b649e3c" TYPE="ext4" PARTUUID="f66f0719-02"
/dev/loop0: TYPE="squashfs"
/dev/loop1: TYPE="squashfs"
/dev/loop2: TYPE="squashfs"
/dev/loop3: TYPE="squashfs"
/dev/loop4: TYPE="squashfs"
/dev/loop5: TYPE="squashfs"
/dev/loop6: TYPE="squashfs"
/dev/loop7: TYPE="squashfs"
/dev/sda1: UUID="e30bb0e4-6ede-4a21-9146-8cfa07195afc" TYPE="ext4" PARTUUID="a0741b2a-aa54-4c06-95ad-a8ea514cb960"
ubuntu@service:/$
```

ทำการบันทึก UUID ของ /dev/sda1 ในที่นี้คือ e30bb0e4-6ede-4a21-9146-8cfa07195afc

7. แก้ไข fstab สำหรับการ mount storage ตอน boot บอร์ด Raspberry Pi โดยใช้คำสั่ง

```
# sudo nano /etc/fstab
```

```
ubuntu@service:/$
ubuntu@service:/$ sudo nano /etc/fstab
```

8. แก้ไข fstab โดยเพิ่มบรรทัดที่มีคำสั่งดังต่อไปนี้ลงไปในไฟล์
โดย UUID จะเป็น UUID ของ Storage ที่เราบันทึกเอาไว้ และ ชื่อ mount directory ที่เราได้ทำการสร้างเอาไว้ก่อนหน้านี้

```
UUID=e30bb0e4-6ede-4a21-9146-8cfa07195afc /clusterfs-ssd ext4 defaults 0 2
```

```
  GNU nano 4.8                          /etc/fstab
LABEL=writable  /         ext4   defaults        0 1
LABEL=system-boot       /boot/firmware  vfat    defaults        0       1

UUID=e30bb0e4-6ede-4a21-9146-8cfa07195afc /clusterfs-ssd ext4 defaults 0 2

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line
```

9. ทำการ mount ตัว Storage กับ directory โดยใช้คำสั่ง

```
# sudo mount -a
```

10. ทำการแก้ไข permission ในการเข้าถึง directory ที่ทำการ mount แล้วโดยใช้คำสั่ง

```
ubuntu@service:/$ sudo chown nobody.nogroup -R /clusterfs-ssd
ubuntu@service:/$ sudo chmod 766 -R /clusterfs-ssd
ubuntu@service:/$
```

```
# sudo chown nobody.nogroup -R /clusterfs-ssd
# sudo chmod 766 -R /clusterfs-ssd
```

ขั้นตอนที่ 5.2 : ติดตั้ง NFS Server และ export share directory

1. ติดตั้ง NSF Server โดยใช้คำสั่ง

```
# sudo apt install nfs-kernel-server -y
```

2. แก้ไขไฟล์ /etc/exports เพื่อกำหนด directory ที่เราต้องการที่จะแชร์และสิทธิในการเข้าถึง โดยใช้คำสั่ง

```
# sudo nano /etc/exports
```

ต่อไปจะเป็นการเพิ่ม directory ที่ต้องการจะแชร์ กับ hostname หรือ subnet ที่สามารถเข้าถึง directory นี้ได้โดยเพิ่มบรรทัดด้านล่างนี้

```
/clusterfs-ssd 10.50.0.0/24(rw,sync,no_root_squash,no_subtree_check)
```

```
  GNU nano 4.8                          /etc/exports
# /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)
#


/clusterfs-ssd 10.50.0.0/24(rw,sync,no_root_squash,no_subtree_check)


^X ExitHelp    ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line
```

3. จากนั้นทำการ Update ค่าของ NFS Server โดยพิมพ์คำสั่งดังต่อไปนี้

```
# sudo exportfs -a
```

ขั้นตอนที่ 5.3 : ติดตั้ง NFS Client และ mount share directory

1. ให้ติดตั้ง NSF Client ให้กับเครื่องที่เหลือ [node1], [node2] , [node3] , [node4] โดยใช้คำสั่ง

```
# sudo apt install nfs-common -y
```

2. ทำการสร้าง mount directory โดย mount directory ของทุก node ใน cluster ควรจะมีชื่อที่เหมือนกัน โดยในที่นี้จะใช้ชื่อว่า /clusterfs-ssd โดยใช้คำสั่งดังต่อไปนี้ในการสร้าง

```
# sudo mkdir /clusterfs-ssd
# sudo chown nobody.nogroup -R /clusterfs-ssd
# sudo chmod 777 -R /clusterfs-ssd
```

```
ubuntu@node01:~$ mkdir /clusterfs-ssd/
ubuntu@node01:~$ sudo chown nobody.nogroup -R /clusterfs-ssd
ubuntu@node01:~$ sudo chmod 777 -R /clusterfs-ssd
```

```
ubuntu@node01:~$ cd /
ubuntu@node01:/$ ls
bin   clusterfs-ssd  dev  home  lost+found  mnt  proc  run   snap  sys  usr
boot  clusterfs-usb  etc  lib   media       opt  root  sbin  srv   tmp  var
```

3. แก้ไข fstab สำหรับการ mount storage ตอน boot บอร์ด Raspberry Pi โดยใช้คำสั่ง

```
# sudo nano /etc/fstab
```

```
ubuntu@node01:/$
ubuntu@node01:/$ sudo nano /etc/fstab
```

4. แก้ไข fstab โดยเพิ่มบรรทัดที่มีคำสั่งดังต่อไปนี้ลงไปในไฟล์
โดย 10.50.0.250:/clusterfs-ssd จะเป็น IP Address หรือ Hostname ของ NFS Server ที่เราได้แชร์ Directory เอาไว้ ส่วน /clusterfs-ssd ก็เป็น directory ที่เราต้องการจะ mount ไว้

```
10.50.0.250:/clusterfs-ssd /clusterfs-ssd nfs  defaults  0  2
```

```
  GNU nano 4.8                          /etc/fstab
LABEL=writable  /        ext4   defaults        0 1
LABEL=system-boot       /boot/firmware  vfat    defaults        0       1

UUID=4a60d54c-6865-4357-b408-8aa26a434a84 /clusterfs-usb ext4 defaults 0 2

10.50.0.250:/clusterfs-ssd  /clusterfs-ssd   nfs   defaults   0 0


^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line
```

5. ทำการ mount ตัว Storage กับ directory โดยใช้คำสั่ง

```
# sudo mount -a
```

6. ตรวจสอบการทำงานของ NFS Share โดยลองทำการสร้างไฟล์ที่ [node1] และกลับไปดูไฟล์ที่ [service] โดยใช้คำสั่ง

```
[node1] # touch /clusterfs-ssd/testfs-2022.txt
[node1] # ls /clusterfs-ssd
```

```
ubuntu@node01:~$ touch /clusterfs-ssd/testfs-2022.txt
ubuntu@node01:~$ ls /clusterfs-ssd/
apps  lost+found  modulefiles  share  slurm  slurm.conf  test.txt  testfs-2022.txt
ubuntu@node01:~$
```

```
[service] # ls /clusterfs-ssd
```

```
ubuntu@service:~$ ls /clusterfs-ssd/
apps  lost+found  modulefiles  share  slurm  slurm.conf  test.txt  testfs-2022.txt
ubuntu@service:~$
```

ขั้นตอนที่ 6 : ติดตั้ง SLURM

ขั้นตอนที่ 6.1 : ตั้งค่า Control / Master Node

1. ในการติดตั้ง SLURM สำหรับ Control Node เราจะใช้เครื่อง [service] เริ่มจาก ทำการ SSH เข้าไปที่บอร์ด Raspberry Pi โดย

```
<PowerShell or Terminal> # ssh ubuntu@10.50.0.250
```

2. เพิ่มชื่อ hostname ของทุก node ใน cluster ลงไปในไฟล์ /etc/hosts โดยใช้คำสั่ง

```
# sudo nano /etc/hosts
```

แล้วทำการเพิ่มบรรทัดด้านล่างนี้ลงไปในไฟล์ โดยจะเป็น IP Address และ ชื่อของ hostname

```
  GNU nano 4.8                          /etc/hosts

# 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
ff02::3 ip6-allhosts

10.50.0.10 node01
10.50.0.20 node02
10.50.0.30 node03
10.50.0.40 node04

10.50.0.250 service

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line
```

```
10.50.0.10   node01
10.50.0.20   node02
10.50.0.30   node03
10.50.0.40   node04
10.50.0.250 service
```

3. ติดตั้ง SLURM โดยติดตั้งจาก package slurm-wlm โดยใช้คำสั่ง

```
# sudo apt install slurm-wlm -y
```

4. เมื่อติดตั้ง package slurm-wlm เรียบร้อยแล้ว ต่อไปจะเป็นการ config เบื้องต้น โดยใช้คำสั่งต่อไปนี้ เพื่อทำการดาวน์โลดไฟล์ตัวอย่าง

```
# cd /etc/slurm-llnl/
# sudo wget https://raw.githubusercontent.com/redzenova/Apollo-HPC/main/SLURM/slurm.conf
```

```
ubuntu@service:~$ cd /etc/slurm-llnl/
ubuntu@service:/etc/slurm-llnl$ sudo wget https://raw.githubusercontent.com/redzenova/Apollo-HPC/main/SLURM/slurm.conf
```

5. เมื่อขั้นตอนที่ 4 เสร็จเรียบร้อยเราจะได้ไฟล์ที่ชื่อว่า slurm.conf ให้ทำการแก้ไขเนื้อหาภายในโดยใช้ nano จากคำสั่ง

```
# sudo nano /etc/slurm-llnl/slurm.conf
```

```
  GNU nano 4.8                          /etc/slurm-llnl/slurm.conf
#ctldHost=localhost
#ctldAddr=localhost
#ClusterName=cluster
#SlurmctldPort=6817
#SlurmdPort=6818
#SlurmdSpoolDir=/var/lib/slurm-llnl/slurmd
#SlurmUser=root
#SlurmdUser=root
#StateSaveLocation=/var/lib/slurm-llnl/slurmctld
#SwitchType=switch/none
#TaskPlugin=task/affinity
#
# TIMERS
#KillWait=30
#MinJobAge=300
#SlurmctldTimeout=120
#SlurmdTimeout=300
#
# SCHEDULING
#SchedulerType=sched/backfill
#SelectType=select/linear
#
# LOGGING AND ACCOUNTING
#AccountingStorageType=accounting_storage/none
#ClusterLogFile=/var/log/slurm-llnl/slurmctlds.log
#JobCompHost=localhost
#JobCompLoc=/var/log/slurm-llnl/slurm_jobcomp.log
#JobCompPasswd=
#JobCompPort=
#JobCompType=jobcomp/none
#JobCompUser=root
#JobAcctGatherFrequency=30
#JobAcctGatherType=jobacct_gather/none
#SlurmctldDebug=3
#SlurmctldLogFile=/var/log/slurm-llnl/slurmctlds.log
#SlurmdDebug=3
#SlurmdLogFile=/var/log/slurm-llnl/slurmd.%h.log
#
# POWER SAVE SUPPORT FOR IDLE NODES (optional)
#SuspendProgram=
#ResumeProgram=
#SuspendTimeout=
#ResumeTimeout=
#ResumeRate=
#SuspendExcNodes=
#SuspendExcPartNodes=
#PrologSlurmctld=
#PrologSlurmctld=
#Prolog=
#Prolog=
#PrologSlurmctld=
#PrologSlurmctld=
#Epilog=
#Epilog=
#EpilogSlurmctld=
#EpilogSlurmctld=
```

slurm.conf

CONFIG	Parameter	Info
ClusterName	apollo	ชื่อของ cluster
SlurmctldHost	service	Hostname หรือ IP Address ของเครื่องที่จะใช้เป็น Control หรือ Master Node
# COMPUTE NODES	NodeName=**node01** CPUs=4 State=UNKNOWN NodeName=**node02** CPUs=4 State=UNKNOWN NodeName=**node03** CPUs=4 State=UNKNOWN NodeName=**node04** CPUs=4 State=UNKNOWN	ระบุ Hostname หรือ IP Address ของเครื่องจะใช้เป็น Compute Node
# PARTITION	PartitionName=**general** Nodes=node[01-04] Default=**YES** MaxTime=168:00:00 DefaultTime=01:00:00 State=**UP**	สร้าง Partition ที่ไว้สำหรับใช้งาน

6. กำหนดการเข้าถึงทรัพยากรของ Slurm ผ่าน cgroup โดยเพิ่มไฟล์ cgroup.conf และ cgroup_allowed_devices_file.conf ลงไปใน directory ที่ /etc/slurm-llnl โดยสามารถดาวน์โหลดไฟล์ได้โดยใช้คำสั่งดังต่อไปนี้

```
# cd /etc/slurm-llnl/
# sudo wget https://raw.githubusercontent.com/redzenova/Apollo-HPC/main/SLURM/cgroup.conf
# sudo wget https://raw.githubusercontent.com/redzenova/Apollo-HPC/main/SLURM/cgroup_allowed_devices_file.conf
```

7. ทำการ copy ไฟล์ config ต่าง ๆ ไปยัง NFS Share Storage เพื่อใช้สำหรับกำหนดค่าให้ node อื่น ๆ โดยใช้คำสั่งดังนี้

```
# sudo cp slurm.conf cgroup.conf cgroup_allowed_devices_file.conf /clusterfs-ssd
# sudo cp /etc/munge/munge.key /clusterfs-ssd
```

```
ubuntu@service:~$ cd /etc/slurm-llnl/
ubuntu@service:/etc/slurm-llnl$ sudo cp slurm.conf cgroup.conf cgroup_allowed_devices_file.conf /clusterfs-ssd
ubuntu@service:/etc/slurm-llnl$ sudo cp /etc/munge/munge.key /clusterfs-ssd
ubuntu@service:/etc/slurm-llnl$
```

8. เปิดและสั่งการทำงานของ Slurm Control

Munge :

```
# sudo systemctl enable munge
# sudo systemctl start munge
```

Slurm Daemon :

```
# sudo systemctl enable slurmd
# sudo systemctl start slurmd
```

Slurm Control Daemon :

```
# sudo systemctl enable slurmctld
# sudo systemctl start slurmctld
```

9. ทำการ reboot โดยใช้คำสั่ง

```
# sudo reboot
```

ขั้นตอนที่ 6.2 : ตั้งค่า Compute Node

1. ในการติดตั้ง SLURM สำหรับ Compute Node เราจะใช้เครื่อง [node1], [node2] , [node3] และ [node4] เริ่มจาก ทำการ SSH เข้าไปที่บอร์ด Raspberry Pi โดยเริ่มที่ [node1]

```
<PowerShell or Terminal> # ssh ubuntu@10.50.0.10
```

2. เพิ่มชื่อ hostname ของทุก node ใน cluster ลงไปในไฟล์ /etc/hosts โดยใช้คำสั่ง

```
# sudo nano /etc/hosts
```

แล้วทำการเพิ่มบรรทัดด้านล่างนี้ลงไปในไฟล์ โดยจะเป็น IP Address และ ชื่อของ hostname

```
  GNU nano 4.8                          /etc/hosts

# 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
ff02::3 ip6-allhosts

10.50.0.10 node01
10.50.0.20 node02
10.50.0.30 node03
10.50.0.40 node04

10.50.0.250 service

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line
```

```
10.50.0.10   node01
10.50.0.20   node02
10.50.0.30   node03
10.50.0.40   node04
10.50.0.250 service
```

3. ติดตั้ง SLURM Client โดยติดตั้งจาก package slurm-client และ slurmd โดยใช้คำสั่ง

```
# sudo apt install slurmd slurm-client -y
```

4. เมื่อติดตั้ง package เรียบร้อยแล้ว ต่อไปจะเป็นการ config โดยใช้คำสั่งต่อไปนี้ เพื่อ copy ไฟล์ต่าง ๆ จาก NFS Share Storage

```
# sudo cp /clusterfs-ssd/munge.key /etc/munge/munge.key
# sudo cp /clusterfs-ssd/slurm.conf /etc/slurm-llnl/slurm.conf
# sudo cp /clusterfs-ssd/cgroup* /etc/slurm-llnl/
```

5. เมื่อขั้นตอนที่ 4 เสร็จเรียบร้อยแล้ว ต่อไปจะเป็นการเปิดและสั่งการทำงานของ Munge

```
# sudo systemctl enable munge
# sudo systemctl start munge
```

6. ทดสอบการทำงานของ Munge หากกำหนดค่าได้ถูกต้องสามารถทดสอบได้ด้วยคำสั่งดังต่อไปนี้ บนเครื่องที่เป็น Compute Node

```
# ssh service munge -n | unmunge
```

```
ubuntu@node01:~$ ssh service munge -n | unmunge
ubuntu@service's password:
STATUS:           Success (0)
ENCODE_HOST:      service (10.50.0.250)
ENCODE_TIME:      2022-02-13 16:11:08 +0000 (1644768668)
DECODE_TIME:      2022-02-13 16:11:08 +0000 (1644768668)
TTL:              300
CIPHER:           aes128 (4)
MAC:              sha256 (5)
ZIP:              none (0)
UID:              ubuntu (1000)
GID:              ubuntu (1000)
LENGTH:           0

ubuntu@node01:~$
```

- หากไม่ขึ้นดังภาพให้ตรวจสอบว่า munge.key ที่อยู่ใน /etc/munge/munge.key ของทุก Node ใน cluster เหมือนกัน หรือ อาจทำการ reboot เครื่อง

7. หากในขั้นตอนที่ 6 ตั้งค่าเรียบร้อยโดย Munge มี STATUS : SUCCESS(0) แล้วนั้น ให้ทำการเปิดและสั่งการทำงานของ Slurmd ดังนี้

```
# sudo systemctl enable slurmd
# sudo systemctl start slurmd
```

8. ทำการกำหนดค่าตามขั้นตอนที่ 6.2 ให้กับ Compute Node ทุกเครื่องใน Cluster

<u>ขั้นตอนที่ 7 : ทดสอบการทำงานของ SLURM</u>

1. หากดำเนินการติดตั้ง Slurm ทั้งตัว Control Node และ Compute Node เรียบร้อยแล้ว เราสามารถตรวจสอบได้โดยการ SSH เข้าไปที่บอร์ด Raspberry Pi ที่เป็น Control Node ในที่นี้คือ [service] โดยใช้คำสั่ง

```
<PowerShell or Terminal> # ssh ubuntu@10.50.0.250
```

2. ทดสอบสถานะของ Slurm และ Node ต่าง ๆ โดยใช้งานคำสั่ง sinfo ดังนี้

```
# sinfo
```

```
ubuntu@service:~$ sinfo
PARTITION   AVAIL  TIMELIMIT  NODES  STATE NODELIST
general*       up 7-00:00:00      3   idle node[02-04]
high-memory    up 7-00:00:00      1   idle node01
ubuntu@service:~$
```

จะมีรายละเอียดของ Partition และจำนวน Compute Node ที่อยู่ภายใต้ Partition แสดงขึ้นมาพร้อมสถานะเป็น up และ idle แสดงว่า Slurm ได้ถูกติดตั้งเรียบร้อยพร้อมใช้งานต่อไป แต่หากขึ้นสถานะ down หรือ State เป็นอย่างอื่นให้เราลองทำการตรวจสอบตามขั้นตอนการติดตั้งใหม่อีกครั้งหนึ่ง

3. ทดสอบการสร้าง Job ลงไปใน Slurm โดยใช้งานคำสั่ง srun ดังนี้

```
# srun --nodes=3 hostname
```

```
ubuntu@service:~$ srun --nodes=3 hostname
node04
node03
node02
ubuntu@service:~$
```

4. สามารถดูรายละเอียด config ของ Slurm โดยใช้งานคำสั่ง scontrol ดังนี้

```
# scontrol show config
```

```
ubuntu@service:~$ scontrol show config
Configuration data as of 2022-02-13T16:58:14
AccountingStorageBackupHost = (null)
AccountingStorageEnforce = none
AccountingStorageHost   = localhost
AccountingStorageLoc    = N/A
AccountingStoragePort   = 6819
AccountingStorageTRES   = cpu,mem,energy,node,billing,fs/disk,vmem,pages
AccountingStorageType   = accounting_storage/slurmdbd
AccountingStorageUser   = N/A
AccountingStoreJobComment = Yes
AcctGatherEnergyType    = acct_gather_energy/none
AcctGatherFilesystemType = acct_gather_filesystem/none
AcctGatherInterconnectType = acct_gather_interconnect/none
AcctGatherNodeFreq      = 0 sec
AcctGatherProfileType   = acct_gather_profile/none
```