

UTS JARINGAN SERVER



Dosen Pengajar:

Asep Kurniawan

IDENTITAS MAHASISWA

Ibrahim Alvaro

TMJ 4 B

2207421035

PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN

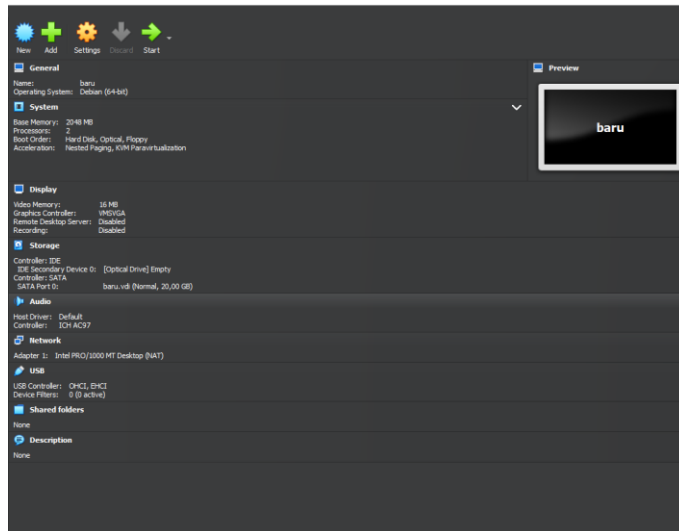
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

POLITEKNIK NEGERI JAKARTA

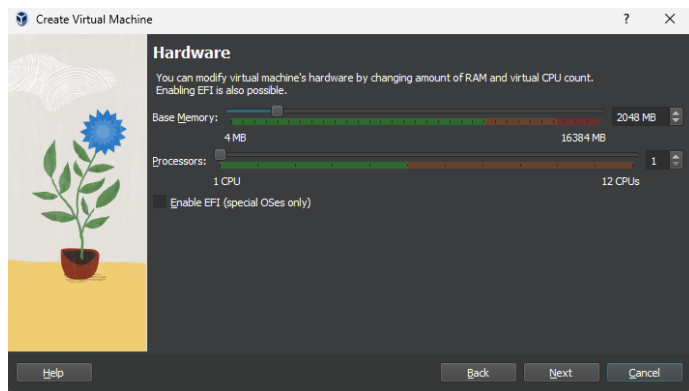
TAHUN AJARAN 2023/2024

MEMBUAT VIRTUAL MACHINE

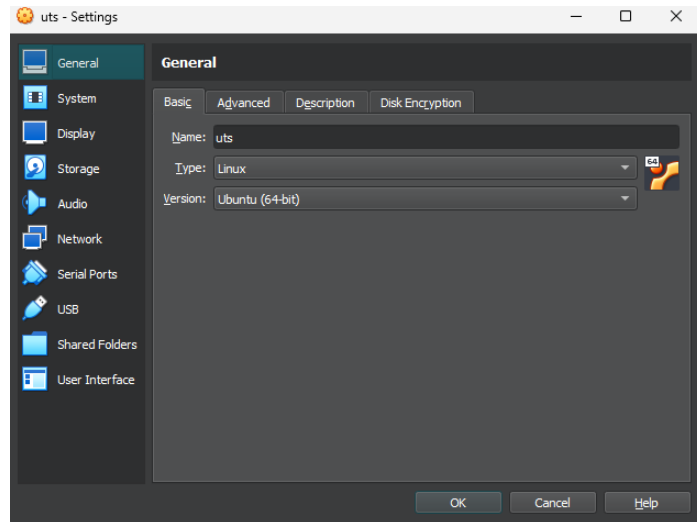
Untuk memulai membuat mesin virtual untuk Ubuntu Server kita bisa dengan mulai dengan cara klik **New**.



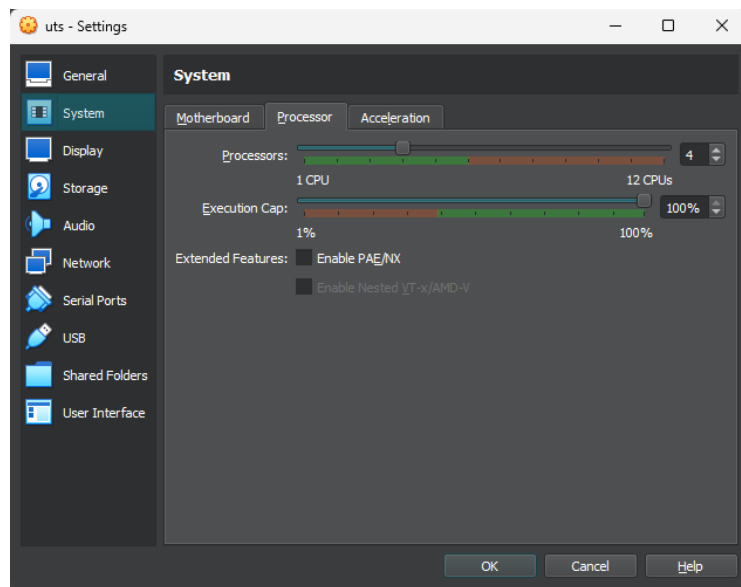
Selanjutnya kita buat nama untuk mesin virtual yang akan kita buat, untuk nama ini bebas bisa anda sesuaikan dengan nama yang kamu inginkan. Pada option **Type** kita pilih **Linux** dan pada **Version** kita pilih **Ubuntu (64-bit)**.



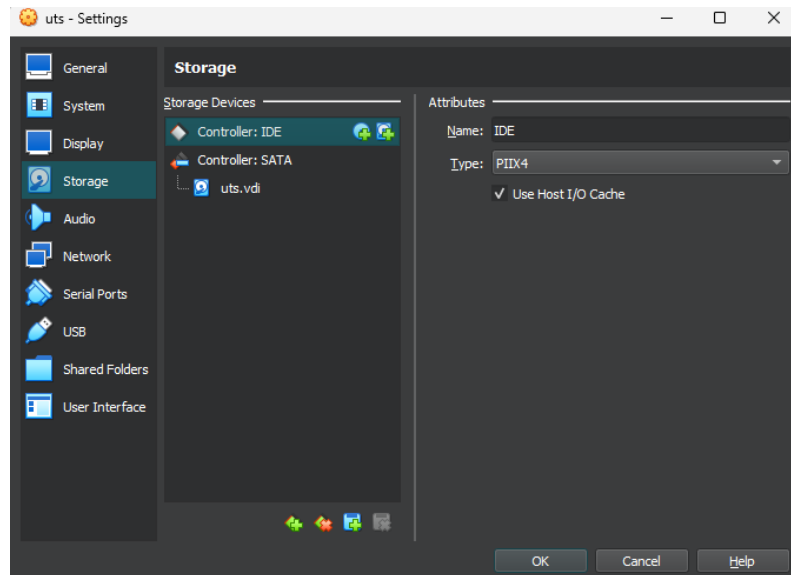
Setelah kita tekan **next** maka kita sudah selesai melakukan pembuatan virtual mesin selanjutnya klik **settings** untuk melakukan konfigurasi lainnya.



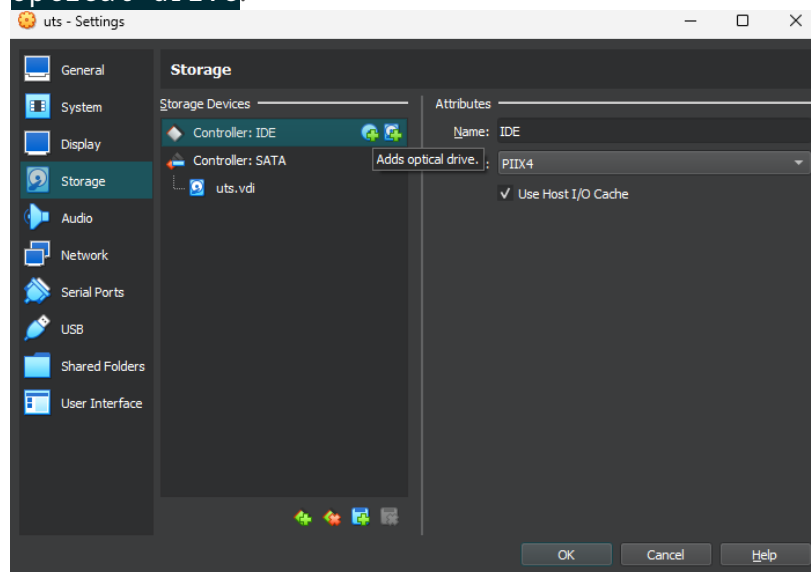
Pada tab **system** kita tentukan core prosesor yang akan kita gunakan dalam virtual mesin ini. Secara default mesin virtual yang kita buat hanya menggunakan **1** core saja namun jika prosesor host kalian memiliki banyak core saya lebih menganjurkan untuk menambah ini. Langkah ini sebenarnya hanya optional tapi agar didapatkan peforma terbaik pada virtual mesin kalian untuk core prosesor ini bisa kalian tambah agar nantinya mesin virtual yang dibuat bisa memiliki peforma yang optimal , maka pada tutorial ini saya menambah jumlah core yang semula hanya **1** saja menjadi **4**.



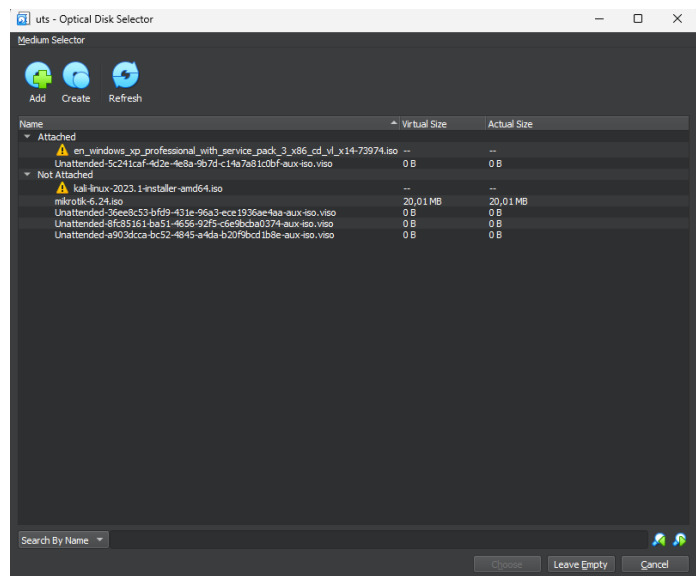
Selanjutnya pada tab **storage** kita remove saja **controller IDE** karane kita akan gunakan **Controller SATA** saja.



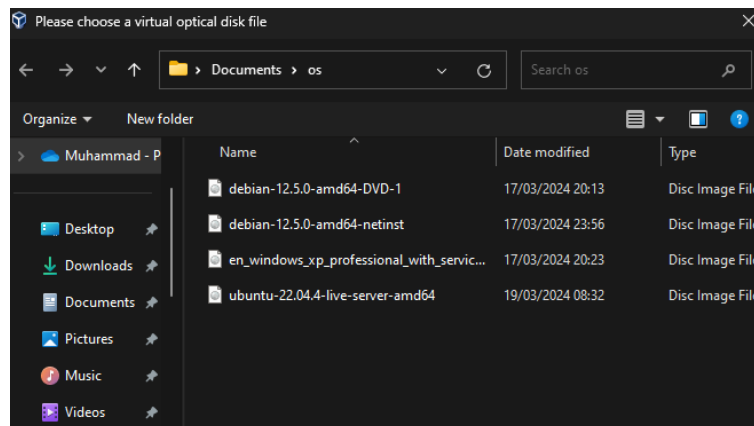
Kemudian untuk menambahkan Installer ubuntu yang sudah didownload caranya adalah dengan klik **Adds optical drive.**



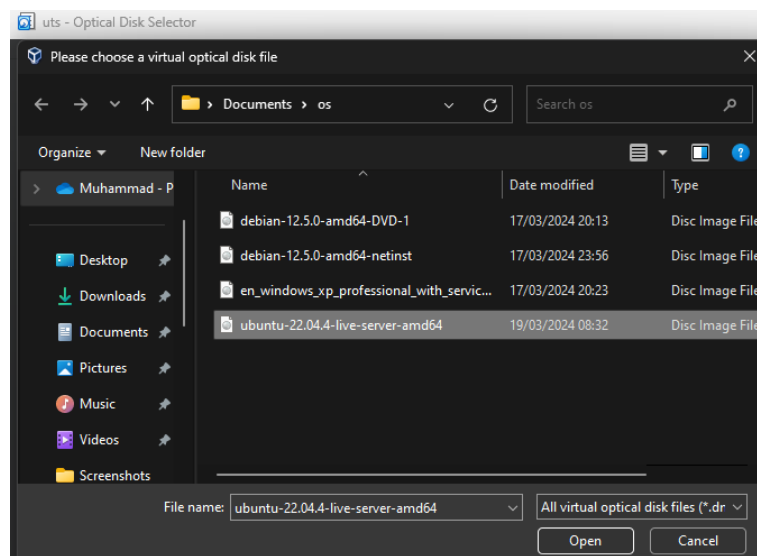
Klik **Add**.



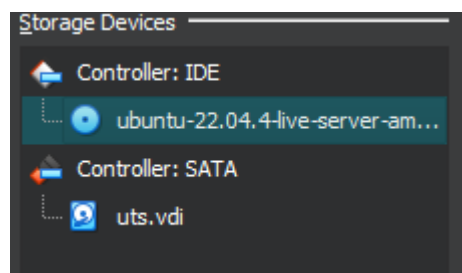
Kemudian cari file **.iso** installer Ubuntu 20.04 LTS jika sudah klik **Open**.



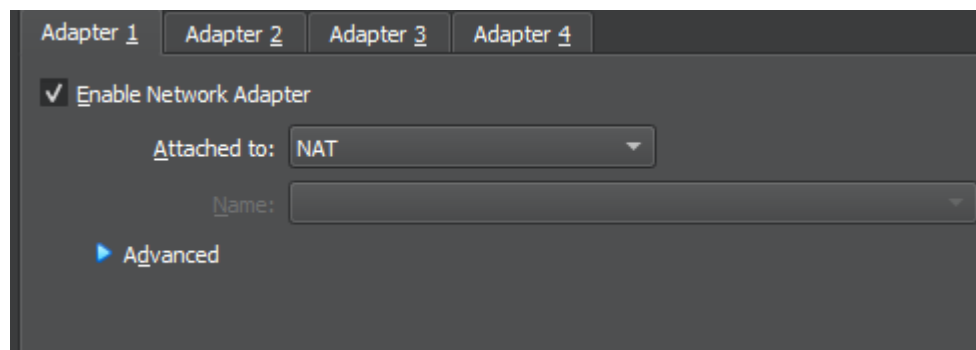
Klik **Choose** pada installer yang sudah kita tambahkan.



Jika sudah maka nantinya optical drive installer telah berhasil ditambahkan.

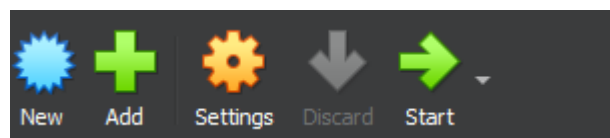


Selanjutnya pada tab **Network** kita disable untuk networking nya karena secara default pada setiap mesin virtual yang kita buat sudah otomatis terhubung dengan internet melalui jaringan **NAT** virtualbox. Mengapa kita matikan internetnya ? Karena agar instalasi Ubuntu OS nantinya cepat dan tidak melakukan proses update pada saat instalasi yang mengakibatkan proses instalasi menjadi lama. Saya cenderung mematikan koneksi internet pada virtual mesin saya ketika melakukan instalasi dan saya lebih senang ketika melakukan update dan upgrade ketika instalasi sudah selesai dan mengganti dari mode **NAT** ke **Bridge** yang nantinya agar host dan mesin virtual kita bisa berada dalam satu network yang sama. Untuk langkah ini saya berikan opsi opsional jika kalian tidak mengikutinya tidak apa-apa atau bisa juga dengan cara memutus koneksi internet langsung dari host anda sehingga mesin virtual tidak bisa terkoneksi dengan internet. Jika sudah tidak ada konfigurasi yang ingin anda edit atau tambahkan bisa kalian tekan **OK**.

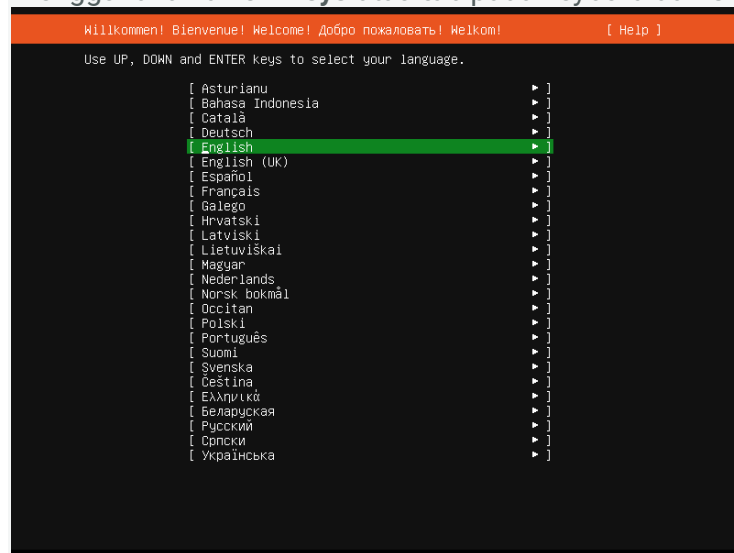


INSTALASI UBUNTU SERVER

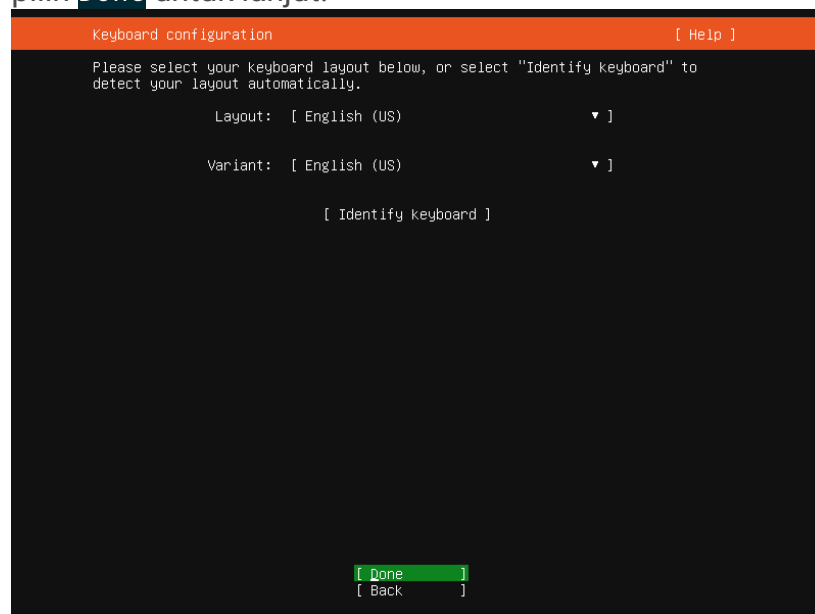
Setelah mesin virtual sudah selesai dibuat selanjutnya kita mulai menyalakan mesin virtual dengan klik **Start**.



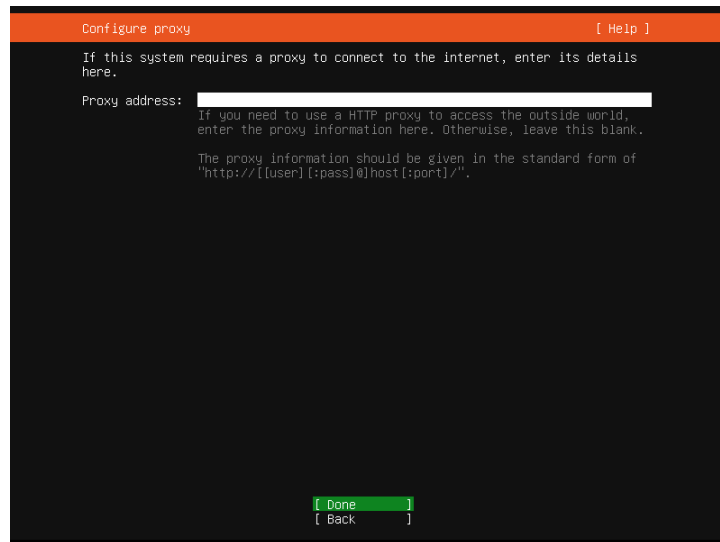
Pilih bahasa instalasi yang kalian inginkan. Disini saya memilih untuk memilih bahasa **English** saja. Karena secara default Installer Ubuntu Server ini menggunakan mode CLI maka untuk navigasinya kalian bisa menggunakan **arrow keys** atau **tab** pada keyboard dan **enter** untuk memilih/konfirmasi.



Selanjutnya kita diminta untuk melakukan konfigurasi keyboard layout. Biarkan default saja dan pilih **Done** untuk lanjut.



Untuk **Proxy Address** kita kosongi saja.



Configure proxy [Help]

If this system requires a proxy to connect to the internet, enter its details here.

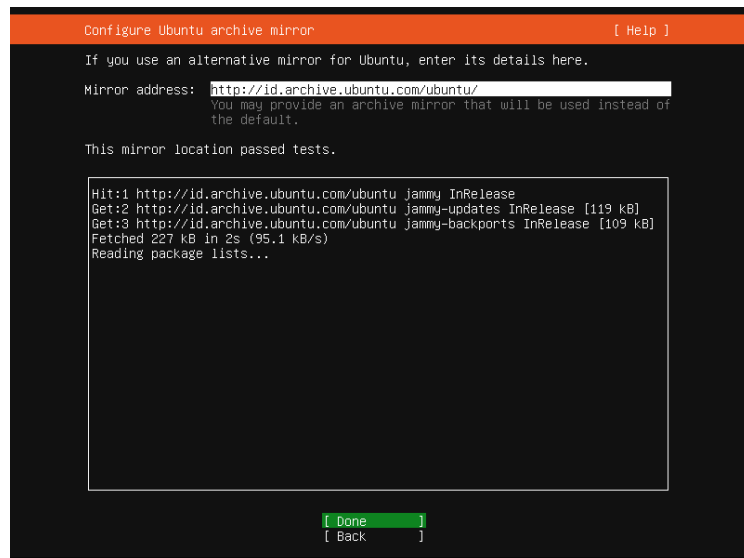
Proxy address:

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]/".

[Done]
[Back]

Untuk **Mirror Address** kita biarkan default saja.



Configure Ubuntu archive mirror [Help]

If you use an alternative mirror for Ubuntu, enter its details here.

Mirror address:

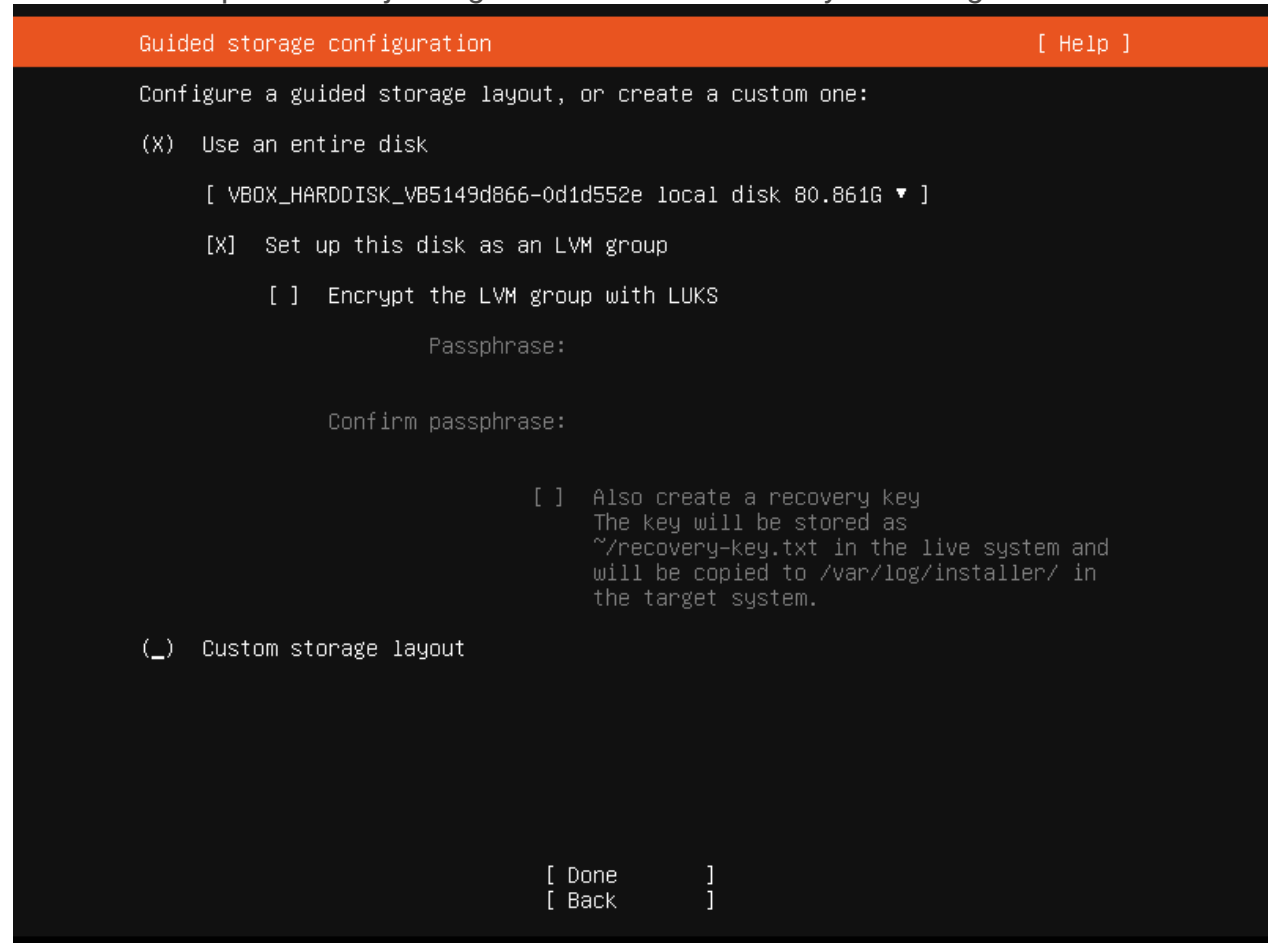
You may provide an archive mirror that will be used instead of the default.

This mirror location passed tests.

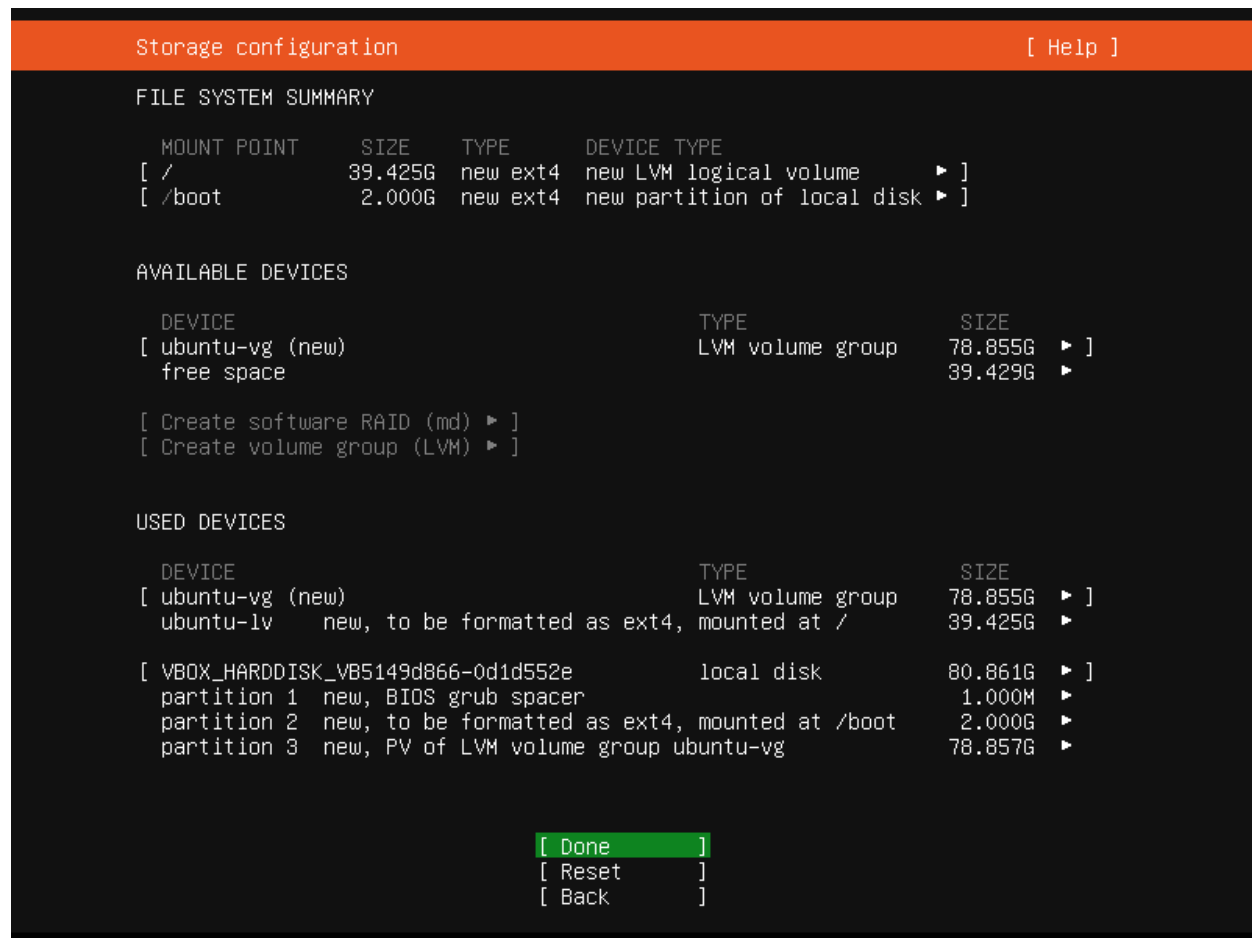
```
Hit:1 http://id.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://id.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://id.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Fetched 227 kB in 2s (95.1 kB/s)
Reading package lists...
```

[Done]
[Back]

Untuk opsi storage ini saya menggunakan default dan langsung memilih **Done** untuk lanjut. Kalian juga bisa memilih opsi manual jika ingin melakukan custom layout storage.



Selanjutnya akan muncul keterangan bahwa dari storage default yang kita pilih tadi installer akan membagi storage kita menjadi 3 partisi yaitu **bios_grub, boot, dan / (root)**. Jika dirasa sudah mengerti maka pilih **Done**.



Maka selanjutnya kita diminta untuk membuat akun yang nantinya kita gunakan untuk masuk kedalam sistem operasi ubuntu server yang telah kita install. Ini tidak harus sama, sesuaikan dengan keinginan kalian.

Profile setup

[Help]

Enter the username and password you will use to log in to the system. You can configure SSH access on a later screen but a password is still needed for sudo.

Your name:

Your servers name:
The name it uses when it talks to other computers.

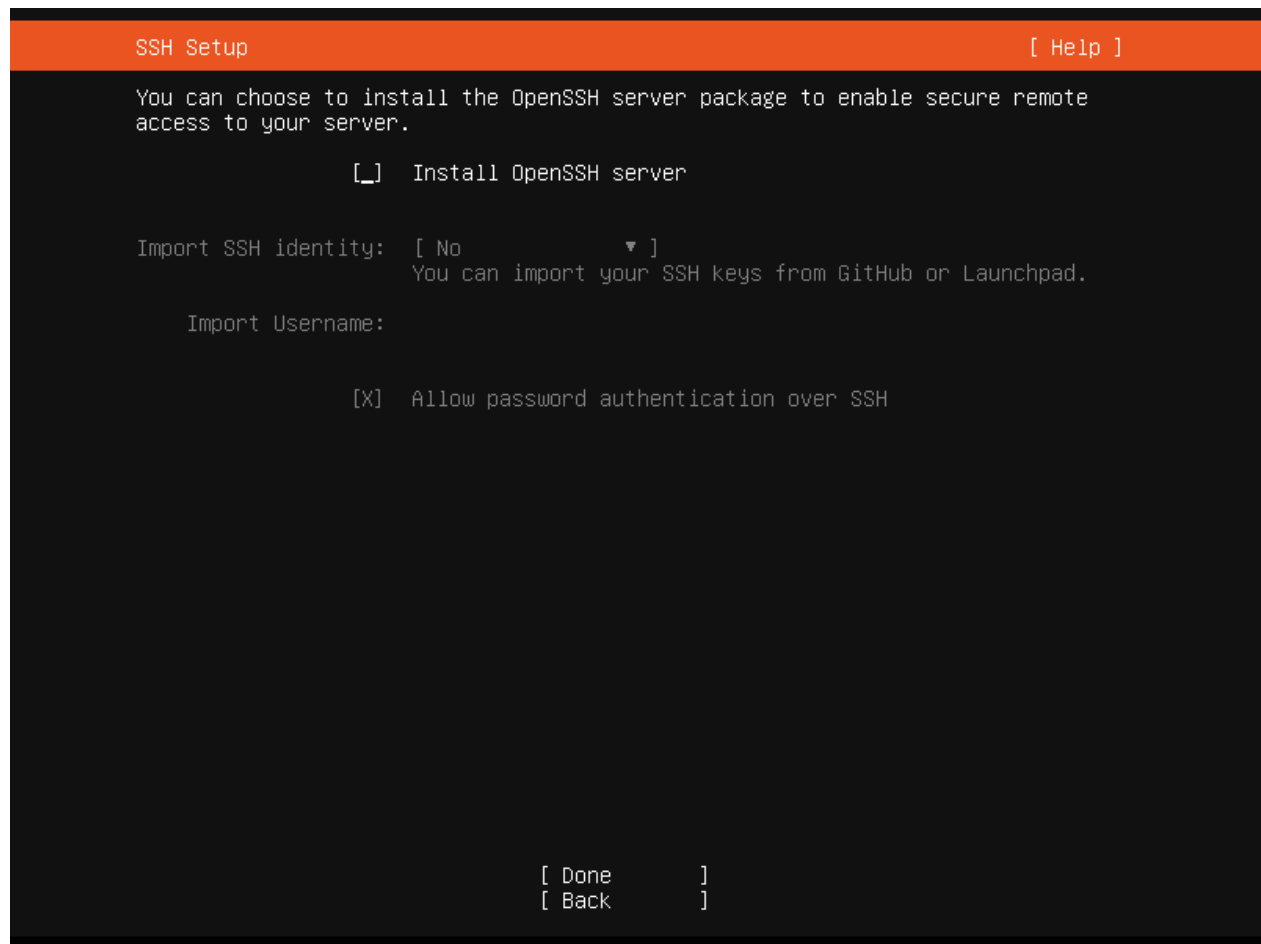
Pick a username:

Choose a password:

Confirm your password:

[Done]

Selanjutnya akan ada tawaran untuk langsung melakukan instalasi OpenSSH Server yang bisa kita gunakan untuk remote server nantinya. Tapi pada step ini optional dan saya memilih untuk tidak menginstallnya.



Tunggu proses instalasi ubuntu server hingga selesai. Jika sudah maka nantinya kita diminta untuk melakukan **reboot** untuk kemudian booting masuk ke sistem operasi yang telah selesai kita install.

Install complete!

[Help]

```
    acquiring and extracting image from cp:///tmp/tmp0zdg3yml/mount ▲
configuring keyboard
  curtin command in-target
executing curtin install curthooks step
  curtin command install
    configuring installed system
      running 'curtin curthooks'
        curtin command curthooks
          configuring apt configuring apt
          installing missing packages
          Installing packages on target system: ['grub-pc']
          configuring iscsi service
          configuring raid (mdadm) service
          installing kernel
          setting up swap
          apply networking config
          writing etc/fstab
          configuring multipath
          updating packages on target system
          configuring pollinate user-agent on target
          updating initramfs configuration
          configuring target system bootloader
          installing grub to target devices
final system configuration
  calculating extra packages to install
  configuring cloud-init
  downloading and installing security updates
  curtin command in-target /
```

[View full log]
[Cancel update and reboot]

Tunggu proses booting selesai hingga muncul prompt login masuk sistem operasi. Jika sudah ada prompt untuk masuk ke sistem operasi masukan username dan password yang telah kita buat saat instalasi tadi. Sampai disini kita telah selesai melakukan instalasi sistem operasi ubuntu server. Jangan lupa untuk menghubungkan kembali koneksi internet pada virtual mesin untuk melakukan update dan upgrade sistem operasi ubuntu server agar mendapatkan pembaharuan sistem terkini.

```
Login incorrect
zhrfn login: zhrfn
Password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-106-generic x86_64)

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

System information as of Mon May 13 06:19:56 PM UTC 2024

System load:  0.86376953125      Processes:            138
Usage of /:   17.9% of 38.51GB   Users logged in:     0
Memory usage: 5%                IPv4 address for enp0s3: 10.0.2.15
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

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

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

zhrfn@zhrfn:~$
```

IFUPDOWN

Untuk melakukan konfigurasi network menggunakan ifupdown terlebih dahulu kita harus melakukan instalasi paket ifupdown terlebih dahulu. Jangan lupa sebelumnya lakukan update dan upgrade sistem terlebih dahulu `sudo apt update && apt upgrade`.

```

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  ppp rdhssd
The following NEW packages will be installed:
  ifupdown
0 upgraded, 1 newly installed, 0 to remove and 21 not upgraded.
Need to get 65.1 kB of archives.
After this operation, 248 kB of additional disk space will be used.
Get:1 http://id.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 ifupdown amd64 0.8.36+nmu1ubuntu3.1 [65.1 kB]
Fetched 65.1 kB in 2s (35.0 kB/s)
Selecting previously unselected package ifupdown.
(Reading database ... 74545 files and directories currently installed.)
Preparing to unpack .../ifupdown_0.8.36+nmu1ubuntu3.1_amd64.deb ...
Unpacking ifupdown (0.8.36+nmu1ubuntu3.1) ...
Setting up ifupdown (0.8.36+nmu1ubuntu3.1) ...
Creating /etc/network/interfaces.
Created symlink /etc/systemd/system/multi-user.target.wants/networking.service → /lib/systemd/system/networking.service.
Created symlink /etc/systemd/system/network-online.target.wants/networking.service → /lib/systemd/system/networking.service.
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
zhrfn@zhrfn:~$ _

```

KONFIGURASI IP STATIC

Kita lihat terlebih dahulu nama interfaces yang akan kita konfigurasi, sebagai contoh disini nama network interfaces yang akan kita konfigurasi adalah `enp0s3`.

Untuk melakukan konfigurasi ip address secara static menggunakan ifupdown kita cukup melakukan edit pada file `/etc/network/interfaces`. Disini saya melakukan edit file menggunakan text editor `nano`.


```
GNU nano 6.2 /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source /etc/network/interfaces.d/*
```

```
zhrfn@zhrfn:~$ sudo nano /etc/network/interfaces
```

lalu ketik config dibawah ini.

```
GNU nano 6.2 /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source /etc/network/interfaces.d/*

# the loopback network interface
auto lo
iface lo inet loopback

auto enp0s3
iface enp0s3 inet static
    address 10.1.1.9/24
    gateway 10.1.1.1
    dns-nameservers 10.1.1.1
```

[Read 14 lines]

Help	Write Out	Where Is	Cut	Execute	Location	Undo
Exit	Read File	Replace	Paste	Justify	Go To Line	Redo

Kemudian jangan lupa save dan exit. Lalu kita restart service network.

```
$ sudo /etc/init.d/networking restart
```

Pastikan ketika restart service tidak ada response error karena jika ada mungkin ada kesalahan penulisan config. Setelah service networking berhasil di restart maka kita dapat melihat IP Address berhasil kita pasang, samapi disini kita sudah bisa melakukan konfigurasi IP Address static menggunakan ifupdown.

```
zhrfn@zhrfn:~$ sudo /etc/init.d/networking restart
Restarting networking (via systemctl): networking.service.
zhrfn@zhrfn:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:20:22:c5 brd ff:ff:ff:ff:ff:ff
    inet 10.1.1.9/24 brd 10.1.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet 192.168.1.8/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 86374sec preferred_lft 86374sec
    inet6 2001:448a:2020:7580:a00:27ff:fe20:22c5/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 187323sec preferred_lft 100923sec
    inet6 fe80::a00:27ff:fe20:22c5/64 scope link
        valid_lft forever preferred_lft forever
zhrfn@zhrfn:~$ _
```

Konfigurasi IP Dynamic

Kemudian untuk konfigurasi IP Address secara Dynamic atau memperoleh IP address dari DHCP Server menggunakan DHCP Client. Kita cukup mengubah konfigurasi pada file `/etc/network/interfaces`. Kita cukup ubah seperti berikut. Setelah itu exit dan save lalu jangan lupa restart servicenya kembali.

```
GNU nano 6.2 /etc/network/interfaces
```

```
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
```

```
source /etc/network/interfaces.d/*
```

```
# the loopback network interface
```

```
auto lo
```

```
iface lo inet loopback
```

```
auto enp0s3
```

```
iface enp0s3 inet dhcp
```

```
root@zhrfn:~# nano /etc/network/interfaces_
```

```
auto enp0s3
iface enp0s3 inet dhcp
```

```
root@zhrfn:~# sudo etc/init.d/networking restart
sudo: etc/init.d/networking: command not found
root@zhrfn:~# etc/init.d/networking restart
-bash: etc/init.d/networking: No such file or directory
root@zhrfn:~# sudo etc/init.d/networking restart
sudo: etc/init.d/networking: command not found
root@zhrfn:~# sudo /etc/init.d/networking restart
Restarting networking (via systemctl): networking.service
.
root@zhrfn:~# _
```

Konfigurasi Multi Interfaces

Untuk melakukan konfigurasi pada multi interfaces. Caranya sama seperti tadi hanya saja kita perlu lihat nama network interfacenya terlebih dahulu. Sebagai contoh disini kita akan melakukan konfigurasi pada kedua network interfaces tersebut dengan **enp0s3** secara dynamic dan **enp0s8** secara static.

Kita lakukan konfigurasi seperti berikut. Dapat kita ketahui bahwa pada konfigurasi ini pada network interfaces **enp0s8** sengaja tidak memiliki gateway dan DNS Server dikarenakan ini adalah konfigurasi ip address untuk network local nantinya dan untuk wan kita gunakan network interfaces **enp0s3**.

```
# Include files from /etc/network/interfaces.d:
source /etc/network/interfaces.d/*

# the loopback network interface
auto lo
iface lo inet loopback

auto enp0s3
iface enp0s3 inet static
    address 192.168.10.1/24

root@zhrfn:~# sudo /etc/init.d/networking restart
Restarting networking (via systemctl): networking.service.
root@zhrfn:~# _
```

Pastikan kemabli tidak ada salah penulisan konfigurasi, kemudian restart service networkingnya. Jika berhasil maka IP Address telah berhasil dipasang.

```

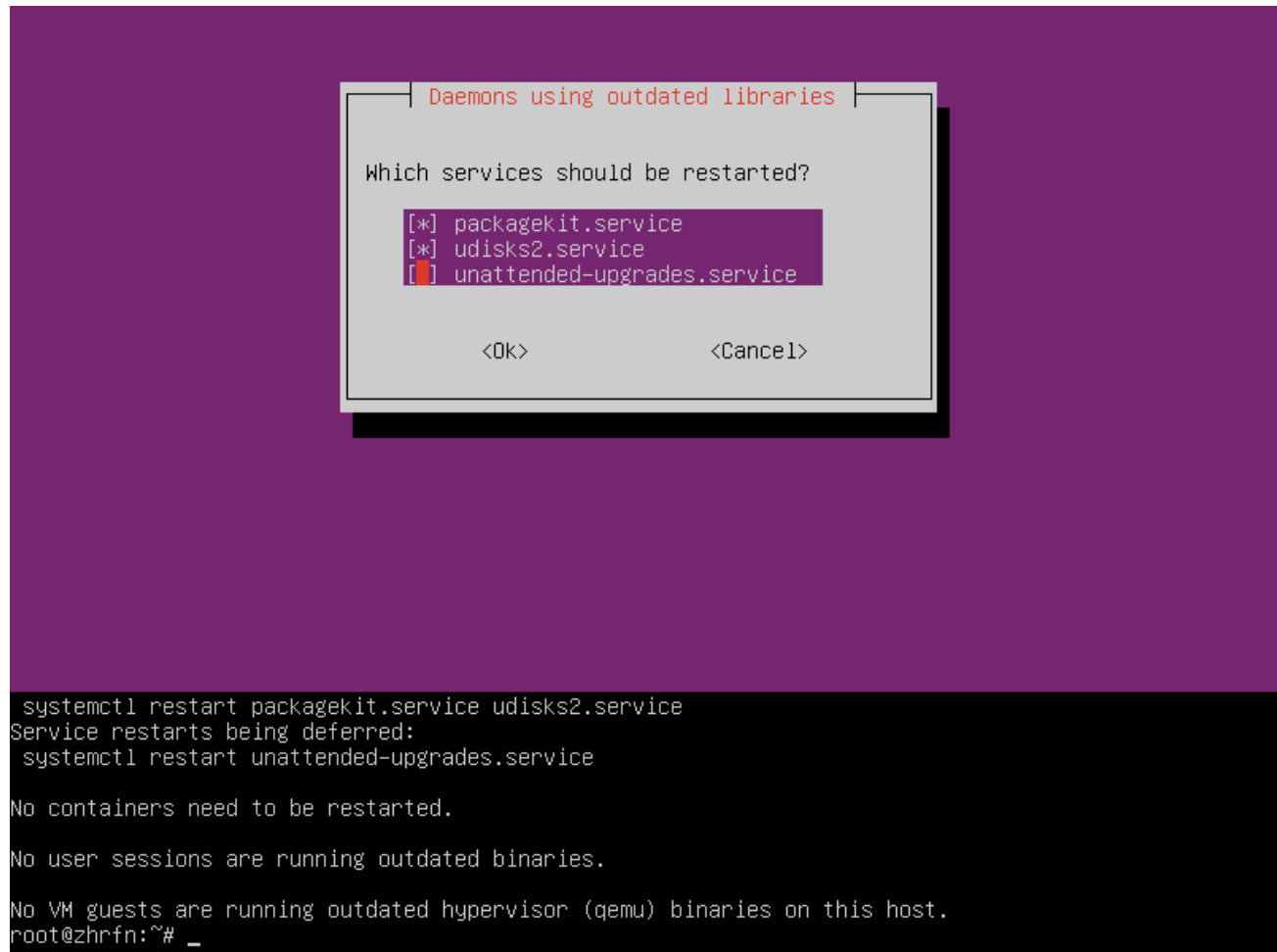
root@zhrfn:~# sudo /etc/init.d/networking restart
Restarting networking (via systemctl): networking.service.
root@zhrfn:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:20:22:c5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.9/24 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 85378sec preferred_lft 85378sec
    inet 192.168.1.8/24 metric 100 brd 192.168.1.255 scope global secondary dynamic enp0s3
        valid_lft 85381sec preferred_lft 85381sec
    inet6 2001:448a:2020:7580:a00:27ff:fe20:22c5/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 185558sec preferred_lft 99158sec
    inet6 fe80::a00:27ff:fe20:22c5/64 scope link
        valid_lft forever preferred_lft forever
root@zhrfn:~# _

```

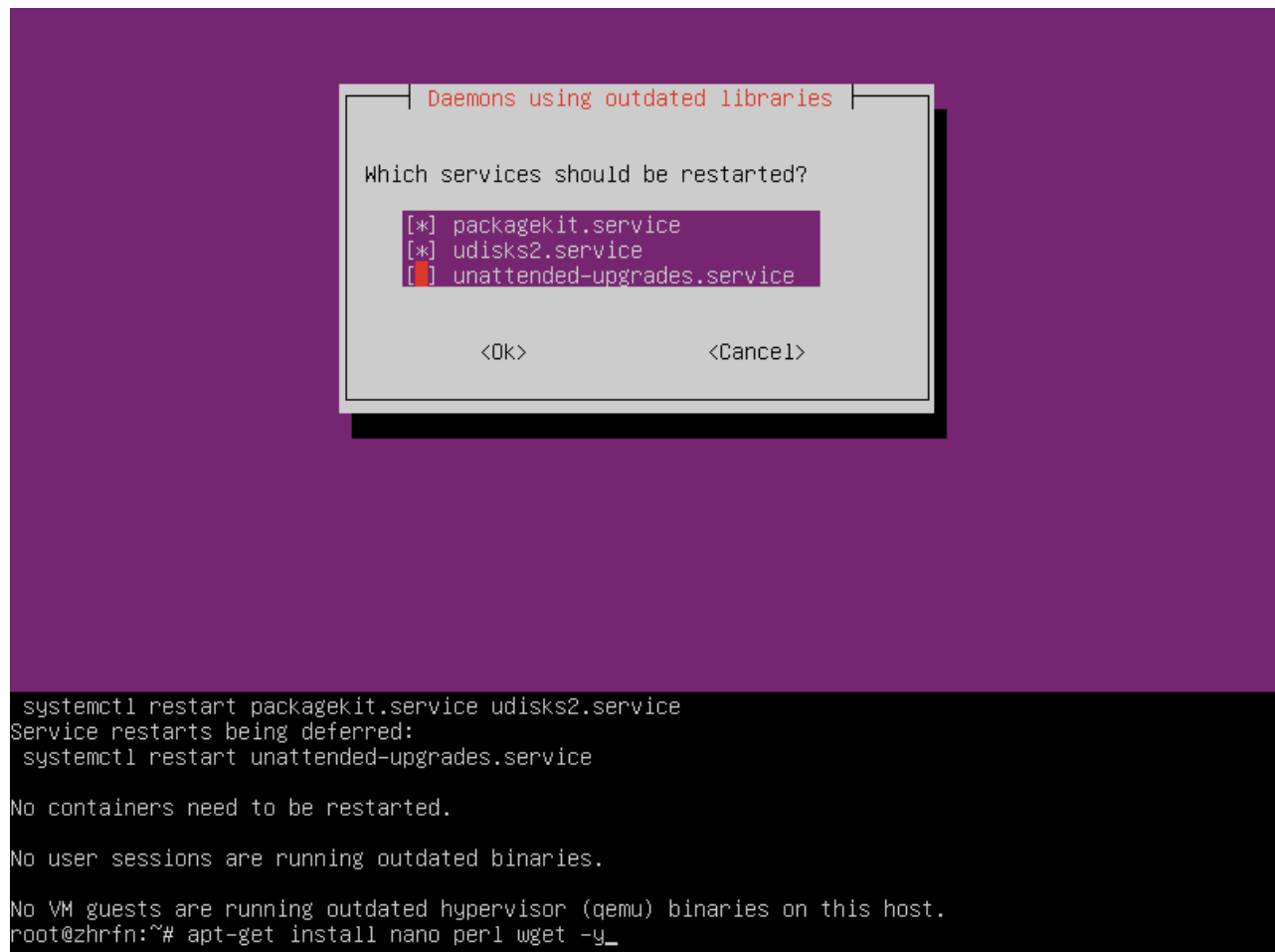
Sampai disini kita telah berhasil melakukan konfigurasi IP Address baik secara static maupun dynamic menggunakan ifupdown.

Cara Install dan Konfigurasi Ansible di Ubuntu 20.04

Login sebagai *root* dan *update package* yang ada pada server agar *up-to-date*.



Install beberapa *packages* yang dibutuhkan.



The image shows a terminal window with a purple background. A dialog box titled "Daemons using outdated libraries" is displayed in the center. The dialog asks "Which services should be restarted?" and lists three services: `packagekit.service`, `udisks2.service`, and `unattended-upgrades.service`. The first two are marked with a red asterisk, and the third is marked with a red square. Below the list are buttons for "<Ok>" and "<Cancel>".

```
systemctl restart packagekit.service udisks2.service
Service restarts being deferred:
systemctl restart unattended-upgrades.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@zhrfn:~# apt-get install nano perl wget -y_
```

Sebelum memulai instalasi dan konfigurasi Ansible di Ubuntu 20.04, kamu perlu perlu meng-*install* Firewall agar lebih aman. Di tutorial kali ini kita akan menggunakan CSF (ConfigServer Security & Firewall). Instalasinya dilakukan dengan perintah di bawah ini:


```

'csget.pl' -> '/etc/cron.daily/csget'
mode of 'auto.generic.pl' changed from 0755 (rwxr-xr-x) to 0700 (rwx-----)
'/etc/csf/csf.conf' -> '/var/lib/csf/backup/1715628018_pre_v14_20_upgrade'

*** IPV6 Enabled

*** IPV6_SPI set to 1

TCP ports currently listening for incoming connections:

UDP ports currently listening for incoming connections:

Note: The port details above are for information only, csf hasn't been auto-configured.

Don't forget to:
1. Configure the following options in the csf configuration to suite your server: TCP_*, UDP_*
2. Restart csf and lfd
3. Set TESTING to 0 once you're happy with the firewall, lfd will not run until you do so
'lfd.service' -> '/usr/lib/systemd/system/lfd.service'
'csf.service' -> '/usr/lib/systemd/system/csf.service'
Created symlink /etc/systemd/system/multi-user.target.wants/csf.service -> /lib/systemd/system/csf.service.
Created symlink /etc/systemd/system/multi-user.target.wants/lfd.service -> /lib/systemd/system/lfd.service.
Failed to disable unit: Unit file firewalld.service does not exist.
Failed to stop firewalld.service: Unit firewalld.service not loaded.
Unit firewalld.service does not exist, proceeding anyway.
Created symlink /etc/systemd/system/firewalld.service -> /dev/null.
'/etc/csf/csfwebmin.tgz' -> '/usr/local/csf/csfwebmin.tgz'

Installation Completed

root@zhrfn:/usr/src/csf# _

```

Di bagian konfigurasi CSF, ubah “make TESTING='1' menjadi 0”.

```

GNU nano 6.2 /etc/csf/csf.conf *
#####
# SECTION:Initial Settings
#####
# Testing flag - enables a CRON job that clears iptables incase of
# configuration problems when you start csf. This should be enabled until you
# are sure that the firewall works - i.e. incase you get locked out of your
# server! Then do remember to set it to 0 and restart csf when you're sure
# everything is OK. Stopping csf will remove the line from /etc/crontab
#
# lfd will not start while this is enabled
TESTING = "0"

# The interval for the crontab in minutes. Since this uses the system clock the
# CRON job will run at the interval past the hour and not from when you issue
# the start command. Therefore an interval of 5 minutes means the firewall
# will be cleared in 0-5 minutes from the firewall start
TESTING_INTERVAL = "5"

# SECURITY WARNING
# =====
#
# Unfortunately, syslog and rsyslog allow end-users to log messages to some
# system logs via the same unix socket that other local services use. This
# means that any log line shown in these system logs that syslog or rsyslog
# maintain can be spoofed (they are exactly the same as real log lines).
#
# Since some of the features of lfd rely on such log lines, spoofed messages
# can cause false-positive matches which can lead to confusion at best, or
# blocking of any innocent IP address or making the server inaccessible at
# worst.
#
# Any option that relies on the log entries in the files listed in
# /etc/syslog.conf and /etc/rsyslog.conf should therefore be considered

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo

```

Muat ulang *rules* CSF-nya.

```

ACCEPT icmpv6 opt in !lo out * ::0 -> ::0
ACCEPT icmpv6 opt in * out !lo ::0 -> ::0
ACCEPT all opt -- in !lo out * 0.0.0.0/0 -> 0.0.0.0/0 ctstate RELATED,ESTABLISHED
ACCEPT all opt -- in * out !lo 0.0.0.0/0 -> 0.0.0.0/0 ctstate RELATED,ESTABLISHED
ACCEPT all opt in !lo out * ::0 -> ::0 ctstate RELATED,ESTABLISHED
ACCEPT all opt in * out !lo ::0 -> ::0 ctstate RELATED,ESTABLISHED
csf: FASTSTART loading TCP_IN (IPv4)
csf: FASTSTART loading TCP6_IN (IPv6)
csf: FASTSTART loading TCP_OUT (IPv4)
csf: FASTSTART loading TCP6_OUT (IPv6)
csf: FASTSTART loading UDP_IN (IPv4)
csf: FASTSTART loading UDP6_IN (IPv6)
csf: FASTSTART loading UDP_OUT (IPv4)
csf: FASTSTART loading UDP6_OUT (IPv6)
ACCEPT all opt -- in lo out * 0.0.0.0/0 -> 0.0.0.0/0
ACCEPT all opt -- in * out lo 0.0.0.0/0 -> 0.0.0.0/0
LOGDROPOUT all opt -- in * out !lo 0.0.0.0/0 -> 0.0.0.0/0
LOGDROPIN all opt -- in !lo out * 0.0.0.0/0 -> 0.0.0.0/0
ACCEPT all opt in lo out * ::0 -> ::0
ACCEPT all opt in * out lo ::0 -> ::0
LOGDROPOUT all opt in * out !lo ::0 -> ::0
LOGDROPIN all opt in !lo out * ::0 -> ::0
csf: FASTSTART loading DNS (IPv4)
csf: FASTSTART loading DNS (IPv6)
LOCALOUTPUT all opt -- in * out !lo 0.0.0.0/0 -> 0.0.0.0/0
LOCALINPUT all opt -- in !lo out * 0.0.0.0/0 -> 0.0.0.0/0
LOCALOUTPUT all opt in * out !lo ::0 -> ::0
LOCALINPUT all opt in !lo out * ::0 -> ::0
**WARNING* Binary location for [SENDMAIL] [/usr/sbin/sendmail] in /etc/csf/csf.conf is either incorrect, is not installed or is not executable
**WARNING* Binary location for [UNZIP] [/usr/bin/unzip] in /etc/csf/csf.conf is either incorrect, is not installed or is not executable
**WARNING* Missing or incorrect binary locations will break csf and lfd functionality
**WARNING* URLGET set to use LWP but perl module is not installed, fallback to using CURL/WGET

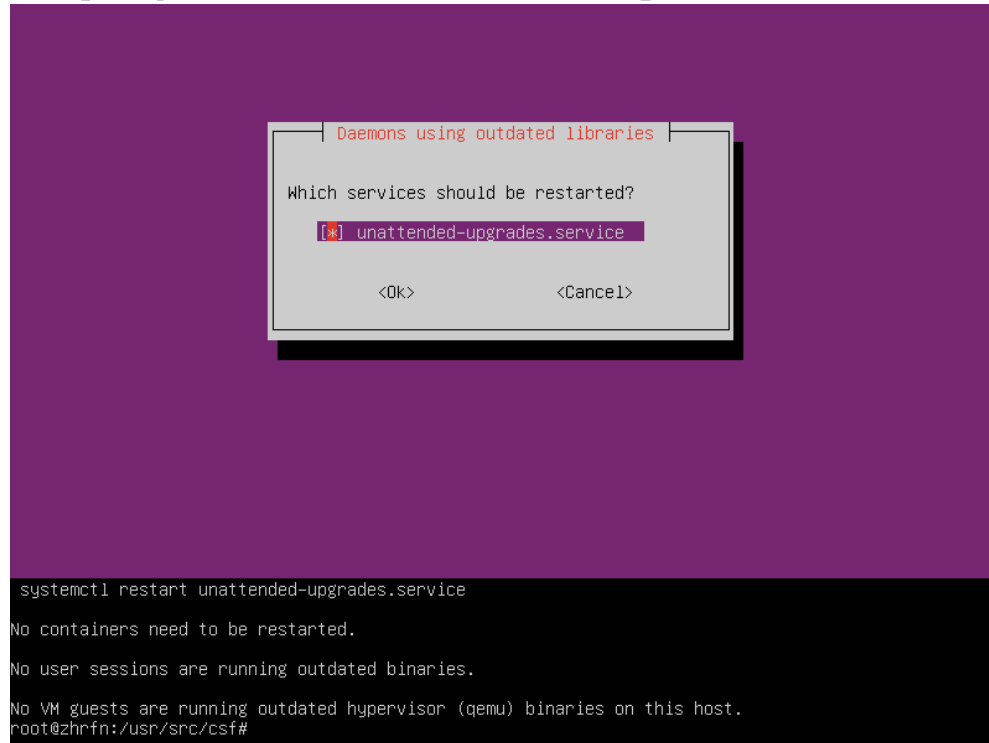
**WARNING* RESTRICT_SYSLOG is disabled. See SECURITY WARNING in /etc/csf/csf.conf.
root@zhfrn:/usr/src/csf# _

```

Tambahkan SSH Key di Control Node pada masing-masing Hosts.

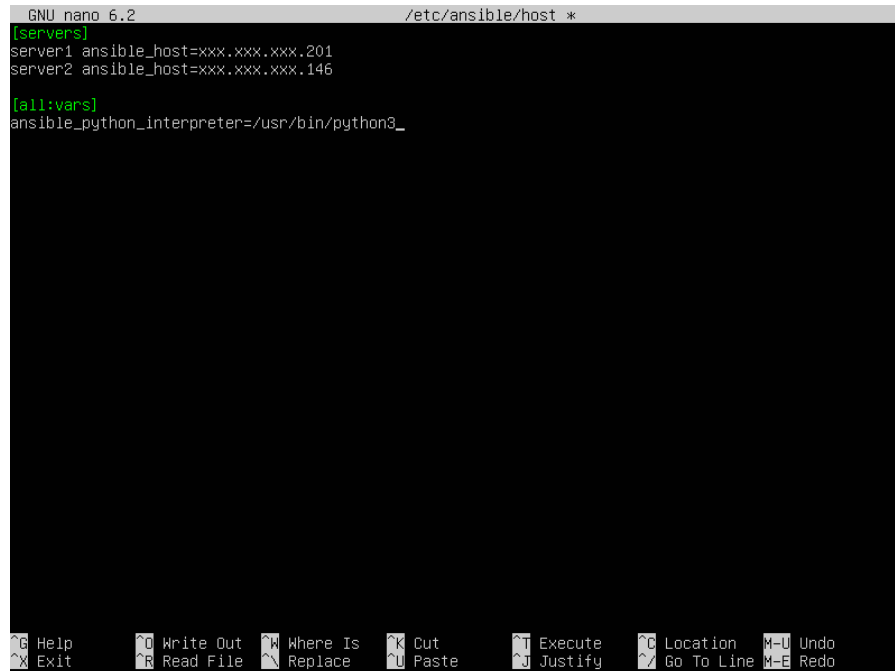
Install Ansible pada **Control Node**, di sini kami menggunakan *default repositories* dari Ubuntu 20.04.

```
$ apt-get install ansible -y
```



Inventory file ini akan berisi mengenai informasi Hosts yang akan di-*manage* oleh Ansible. Kamu bisa menambahkan banyak server di inventori ini dan bisa diatur berdasarkan *group* dan *subgroup*.

Tambahkan *script* di bawah ini.

A screenshot of a terminal window showing the nano text editor editing the file /etc/ansible/hosts. The editor has a dark background with green text. The content of the file is as follows:

```
GNU nano 6.2 /etc/ansible/hosts *
[servers]
server1 ansible_host=xxx.xxx.xxx.201
server2 ansible_host=xxx.xxx.xxx.146

[all:vars]
ansible_python_interpreter=/usr/bin/python3_
```

The bottom of the screen shows the nano editor's command palette with various shortcuts like ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^M-U Undo, ^X Exit, ^R Read File, ^N Replace, ^U Paste, ^J Justify, ^_ Go To Line, and ^M-E Redo.

Lalu simpan dan keluar. Setelah itu cek inventori menggunakan perintah di bawah ini.

```
ansible_python_interpreter=/usr/bin/python3
```

```
root@zhrfn:/usr/src/csf#  
root@zhrfn:/usr/src/csf# ansible-inventory --list -y
```

Maka *output*-nya akan seperti di bawah ini.

```
zhrfn@zhrfn:/etc/ansible$ ansible-inventory --list  
{  
  "_meta": {  
    "hostvars": {  
      "zhrfn1": {  
        "ansible_python_interpreter": "/usr/bin/python3"  
      }  
    }  
  },  
  "all": {  
    "children": [  
      "ungrouped",  
      "servers"  
    ]  
  },  
  "servers": {  
    "hosts": [  
      "zhrfn1"  
    ]  
  }  
}  
zhrfn@zhrfn:/etc/ansible$
```

Setelah melakukan *setting* pada Inventory File yang memasukan beberapa server sekaligus, sekarang waktunya untuk melakukan pengujian untuk *connect* ke Hosts tersebut melalui perintah SSH.

```
zhrfn@zhrfn:/etc/ansible$ ansible all -m ping
zhrfn1 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
zhrfn@zhrfn:/etc/ansible$
```

Maka *output*-nya akan seperti ini:

Setelah dikonfirmasi bahwa kamu dapat menggunakan dan menjalankan perintah Ansible, Control Node dapat berkomunikasi langsung kepada Hosts. Contoh yang akan kami berikan adalah pengecekan Disk Usage pada server Hosts

```
server1 | CHANGED | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
udev            3.9G   0  3.9G   0% /dev
tmpfs           798M 624K  798M   1% /run
/dev/vda1       155G  2.3G  153G   2% /
tmpfs           3.9G   0  3.9G   0% /dev/shm
tmpfs           5.0M   0  5.0M   0% /run/lock
tmpfs           3.9G   0  3.9G   0% /sys/fs/cgroup
/dev/vda15      105M  3.6M  101M   4% /boot/efi
tmpfs           798M   0  798M   0% /run/user/0

server2 | CHANGED | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
udev            2.0G   0  2.0G   0% /dev
tmpfs           395M 608K  394M   1% /run
/dev/vda1       78G   2.2G  76G   3% /
tmpfs           2.0G   0  2.0G   0% /dev/shm
tmpfs           5.0M   0  5.0M   0% /run/lock
tmpfs           2.0G   0  2.0G   0% /sys/fs/cgroup
/dev/vda15      105M  3.6M  101M   4% /boot/efi
tmpfs           395M   0  395M   0% /run/user/0
```

