

Ubuntu

Linux tutorials

- [Thay đổi kernel mặc định trên Ubuntu](#)
- [Scripts cài đặt docker và docker-compose trên Ubuntu và Debian](#)
- [Extend LVM partition và disk trên Ubuntu](#)
- [Backup Daily Weekly Monthly với tar rsync và cron](#)
- [Quản lý Linux Server qua giao diện web với Cockpit](#)
- [Cài đặt iptables trên Ubuntu](#)
- [Sử dụng Docker Compose trên Ubuntu 20.04LTS](#)
- [Tạo một SSH key cơ bản](#)

Thay đổi kernel mặc định trên Ubuntu

Kernel hoạt động như một cầu nối (middleware) giữa phần cứng và phần mềm chạy trên hệ thống.

Kernel giao tiếp với phần cứng thông qua các trình điều khiển (drivers) được tích hợp sẵn bên trong nó hoặc được cài đặt bổ sung dưới dạng một module.

Việc cập nhật kernel – điều này giúp tận dụng tối đa thế mạnh tài nguyên phần cứng của chúng.

Ngoài ra, cập nhật kernel còn giúp khai thác tối đa những tính năng mới của nó và vá các lỗ hổng bảo mật đã được phát hiện trong phiên bản tiền nhiệm.

Trong nội dung bài viết này, mình sẽ chia sẻ cách thực hiện cập nhật kernel trên Ubuntu (hoặc Debian)

Kiểm tra phiên bản kernel hiện tại

Để kiểm tra phiên bản Linux kernel đang được cài đặt trên hệ thống, sử dụng lệnh sau:

```
uname -sr
```

All text found or type unknown

Nâng cấp kernel trên Ubuntu

```
sudo apt update && sudo apt upgrade  
sudo apt install wget
```

Tải Bash script để nâng cấp kernel

Bạn có thể tải xuống trực tiếp gói Kernel Deb từ kho lưu trữ Linux của Ubuntu. Tuy nhiên, để dễ dàng cài đặt, cập nhật và gỡ cài đặt Kernels trên Ubuntu, chúng ta có thể sử dụng scripts có sẵn tại [GitHub](#). Tải về server Ubuntu của bạn.

```
wget https://raw.githubusercontent.com/pimlie/ubuntu-mainline-kernel.sh/master/ubuntu-mainline-kernel.sh
chmod +x ubuntu-mainline-kernel.sh
sudo mv ubuntu-mainline-kernel.sh /usr/local/bin/
```

Kiểm tra Kernel versions

Chạy lệnh sau:

```
ubuntu-mainline-kernel.sh -c
```

Alt text found or type unknown

Liệt kê tất cả Kernel versions

```
ubuntu-mainline-kernel.sh -r
```

Alt text found or type unknown

Cài một Kernel versions

```
sudo ubuntu-mainline-kernel.sh -i <version-number>
```

Sau khi hoàn thành kiểm tra lại cài đặt

```
sudo ubuntu-mainline-kernel.sh -l
```

Reboot lại hệ thống

```
sudo reboot
```

Kiểm tra lại

```
uname -r
```

Thay đổi hoặc set Default Kernel Version

```
sudo nano /etc/default/grub
```

Thêm

```
GRUB_SAVEDEFAULT=true  
GRUB_DEFAULT=saved
```

Alt text found or type unknown

```
sudo update-grub  
sudo reboot
```

Khi khởi động lại ấn **ESC** chọn **Advanced** Alt text found or type unknown

Xoá hoặc gỡ Kernel

Chạy lệnh sau chọn Kernel xoá

```
sudo ubuntu-mainline-kernel.sh -u
```

Mời cà phê

Nếu cảm thấy blog mang đến những thông tin hữu ích cho công việc, cuộc sống, đam mê của bạn, đừng ngại ủng hộ một ly cà phê để mình có thêm động lực chia sẻ thêm nhiều kinh nghiệm, kiến thức nhé.

Bạn có thể ủng hộ mình qua:

Ví MoMo



[Ví MoMo](#)

Paypal

[Hỗ trợ qua Paypal](#)

Cám ơn mọi người!

Scripts cài đặt docker và docker-compose trên Ubuntu và Debian

Chuẩn bị

- Ubuntu / Debian
- Docker
- Docker Compose

Cài đặt

SSH vào server và tạo file sau:

```
nano docker-install-ubuntu.sh
```

```
#!/bin/bash
sudo apt update

echo "Install needed"
sudo apt-get install ca-certificates curl gnupg lsb-release -y

echo "Add Docker's official GPG key"
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update -y
sudo apt-get install docker-ce docker-ce-cli containerd.io -y
sudo usermod -aG docker $(whoami)
newgrp docker

sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
docker-compose --version
```

Lưu lại và thoát.

```
chmod 755 docker-install-ubuntu.sh
sh docker-install-ubuntu.sh
```

Mời cà phê

Nếu cảm thấy blog mang đến những thông tin hữu ích cho công việc, cuộc sống, đam mê của bạn, đừng ngại ủng hộ một ly cà phê để mình có thêm động lực chia sẻ thêm nhiều kinh nghiệm, kiến thức nhé.

Bạn có thể ủng hộ mình qua:

Ví MoMo



[Ví MoMo](#)

Paypal

[Hỗ trợ qua Paypal](#)

Cám ơn mọi người!

Extend LVM partition và disk trên Ubuntu

Bài toán

Bạn đang dùng server **Ubuntu 20.04 LTS** và Disk đang dùng **LVM Disk**. Không may **DATA** trên phân vùng này phình ra. Bạn cần mở rộng thêm. Mình ví dụ:

- Disk ban đầu cài 400GB.
- Mở rộng thêm mới 100GB

Bạn kiểm tra dung lượng Disk như hình bên dưới:

```
df -h
```

Alt text found or type unknown

Mở rộng 100GB bằng cách thêm vật lí trên server. Sau khi mở rộng bạn kiểm tra disk

```
sudo fdisk -l
```

Alt text found or type unknown

Lúc này phân vùng **/dev/sda** đã tăng thêm 100GB (total 500GB). Nhưng trên phân vùng **/dev/sda** vẫn chưa mapping dung lượng mới. Bạn có thể kiểm tra

```
sudo lsblk
```

Alt text found or type unknown

Cấu hình để phân vùng nhận mapping dung lượng mới

```
sudo growpart /dev/sda 3
```

All text found or type unknown

```
sudo lsblk
```

All text found or type unknown

```
sudo pvresize /dev/sda3
```

All text found or type unknown

```
sudo lvextend -l +100%FREE /dev/mapper/ubuntu--vg-ubuntu--lv
```

All text found or type unknown

```
sudo resize2fs /dev/mapper/ubuntu--vg-ubuntu--lv
```

All text found or type unknown

Tham khảo

[Ubuntu – Extend LVM partition and disk](#)

Mời cà phê

Nếu cảm thấy blog mang đến những thông tin hữu ích cho công việc, cuộc sống, đam mê của bạn, đừng ngại ủng hộ một ly cà phê để mình có thêm động lực chia sẻ thêm nhiều kinh nghiệm, kiến thức nhé.

Bạn có thể ủng hộ mình qua:

Ví MoMo



[Ví MoMo](#)

Paypal

[Hỗ trợ qua Paypal](#)

Cảm ơn mọi người!

Backup Daily Weekly Monthly với tar rsync và cron

Bài viết này mình chia sẻ mọi người cách tạo các bản sao lưu hàng ngày, hàng tuần và hàng tháng trên **Ubuntu/Debian** một cách tự động. Ở đây mình sử dụng các lệnh tar, find, rsync và cron để tự động hóa lịch sao lưu.

Daily Backup

Lệnh TAR sẽ tạo một kho lưu trữ nén của thư mục `/var/www/html` trong `/home/kien/backup/daily/data`.
Lệnh find sẽ xóa tất cả các bản sao lưu hàng ngày cũ hơn 7 ngày.

```
tar -zcf /home/kien/backup/daily/backup-$(date +%Y%m%d).tar.gz -C /var/www/ html  
find /home/tony/backup/daily/* -mtime +7 -delete
```

Weekly Backup

Lệnh TAR sẽ tạo một kho lưu trữ nén của thư mục `/var/www/html` trong `/home/kien/backup/weekly`. Lệnh `find` sẽ xóa tất cả các bản sao lưu hàng tuần cũ hơn 31 ngày hoặc 1 tháng.

```
tar -zcf /home/kien/backup/weekly/backup-$(date +%Y%m%d).tar.gz -C /var/www/ html  
find /home/kien/backup/weekly/* -mtime +31 -delete
```

Monthly Backup

Lệnh TAR sẽ tạo một kho lưu trữ nén của thư mục `/var/www/html` trong `/home/kien/backup/monthly`.
Lệnh find sẽ xóa tất cả các bản sao lưu hàng tháng cũ hơn 365 ngày hoặc 1 năm.

```
tar -zcf /home/kien/backup/monthly/backup-$(date +%Y%m%d).tar.gz -C /var/www/ html  
find /home/kien/backup/monthly/* -mtime +365 -delete
```

Automate with Cron

Bạn có thể tạo một cronjob để sao lưu hàng ngày, hàng tuần và hàng tháng. Mở lệnh `crontab -e` và thêm như sau:

```
15 0 * * * sh /home/tony/backup-daily.sh
30 0 * * 1 sh /home/tony/backup-weekly.sh
45 0 1 * * sh /home/tony/backup-monthly.sh
```

Thực hiện tập lệnh sao lưu hàng ngày hàng ngày vào lúc **12:15 AM**

Thực hiện tập lệnh sao lưu hàng tuần vào mỗi thứ Hai lúc **12:30** sáng Thực hiện tập lệnh sao lưu hàng tháng

Backup Externally

Tiếp theo, bạn backup sao lưu vào nơi lưu trữ server khác tránh trường hợp bị server có sự cố

```
rsync -a --delete /home/kien/backup/ root@161.35.143.122:/path/to/remote/backups/
```

Tạo một cron backup auto

```
0 2 * * * rsync -a --delete /home/kien/backup/ root@161.35.143.122:/path/to/remote/backups/
```

Mời cà phê

Nếu cảm thấy blog mang đến những thông tin hữu ích cho công việc, cuộc sống, đam mê của bạn, đừng ngại ủng hộ một ly cà phê để mình có thêm động lực chia sẻ thêm nhiều kinh nghiệm, kiến thức nhé.

Bạn có thể ủng hộ mình qua:

Ví MoMo



[Ví MoMo](#)

Paypal

[Hỗ trợ qua Paypal](#)

Cảm ơn mọi người!

Quản lý Linux Server qua giao diện web với Cockpit

Khi quản lý các Server Linux mình thích dòng lệnh hơn nhưng có một số tình huống mà bạn cần quản lý Server Linux của mình bằng **GUI**.

Ví dụ: Bạn muốn xem nhanh các thông kê Server sử dụng như **CPU, RAM, HDD...** đôi khi nó cũng nhanh hơn và dễ dàng hơn nếu bạn không quen các lệnh. Trong hướng dẫn này, mình chia sẻ các bạn một GUI để quản lý Server Linux của mình có tên là **Cockpit**.

<https://www.youtube.com/embed/XKYbEXSrajw>

Làm thế nào để cài một web GUI trên Linux server

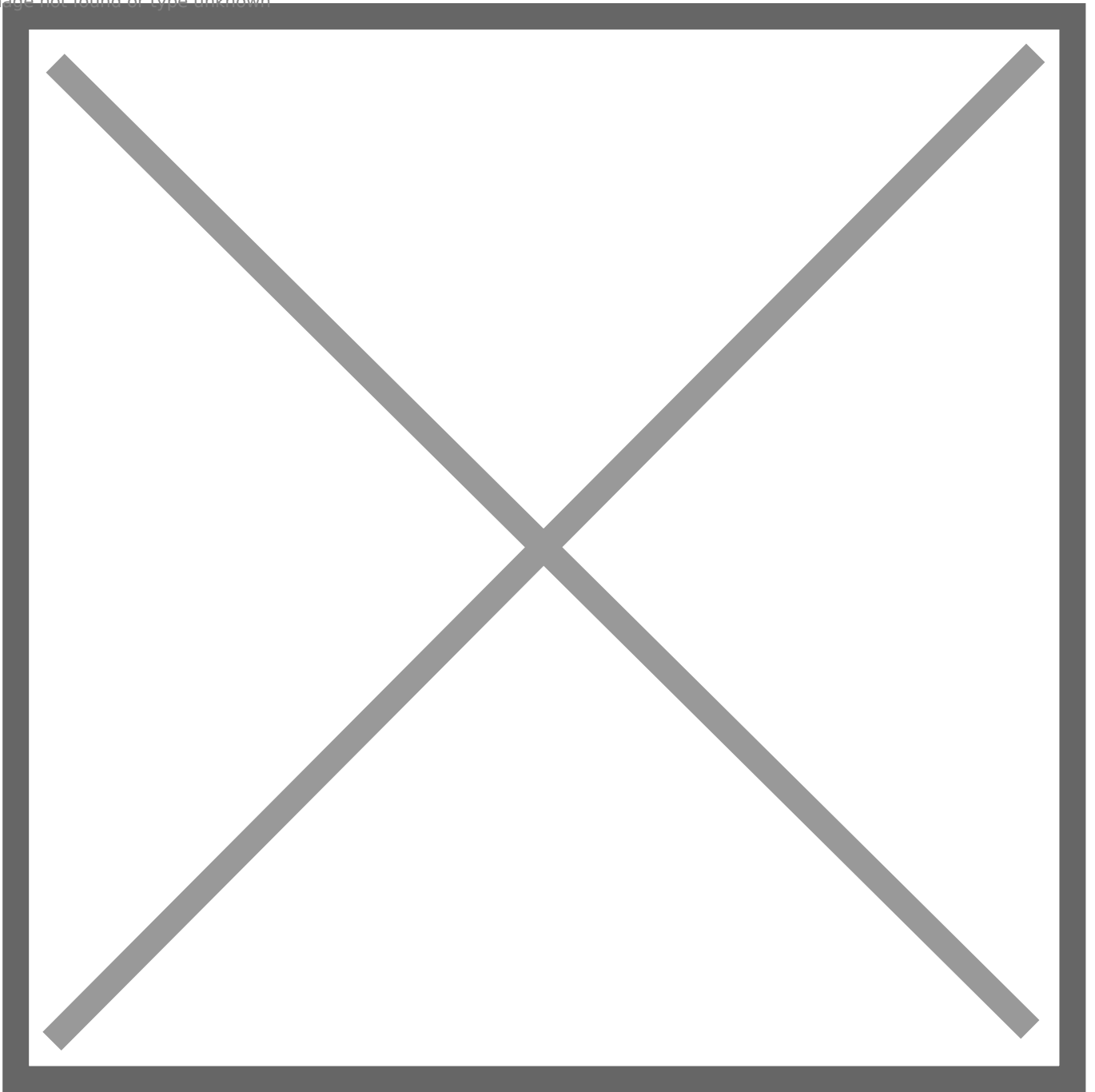
Khi tìm kiếm một GUI web tốt để quản lý Server Linux của mình, thường bạn sẽ tìm thấy các công cụ như **Webmin** hoặc **Cpanel**. Gần đây mình vừa tìm thấy một dự án mã nguồn mở có tên **Cockpit**. Ban đầu **Cockpit** được phát triển cho **RHEL** và **CentOS**, nhưng nó cũng chạy trên một loạt các bản phân phối **Linux** phổ biến như **Ubuntu, Debian**. **Cockpit** trông nhẹ, trực quan và dễ sử dụng.

Cockpit có các tùy chỉnh nâng cao và mở rộng, do có các plugin tùy chỉnh. Trước khi cài đặt phần mềm này trên Server Linux của mình, bạn nên kiểm tra tài liệu cài đặt nếu bản phân phối Linux của bạn được hỗ trợ. **Lưu ý**, phần mềm này có thể hơi khác trên một số bản phân phối vì chúng không phải lúc nào cũng có các tính năng giống nhau. Trong ví dụ này, mình cài đặt điều này trên Server **Ubuntu 20.04 LTS**.

```
sudo apt install cockpit
```

Sau khi cài đặt xong vào trình duyệt gõ IP:9090 để login với user đã tạo trên Server

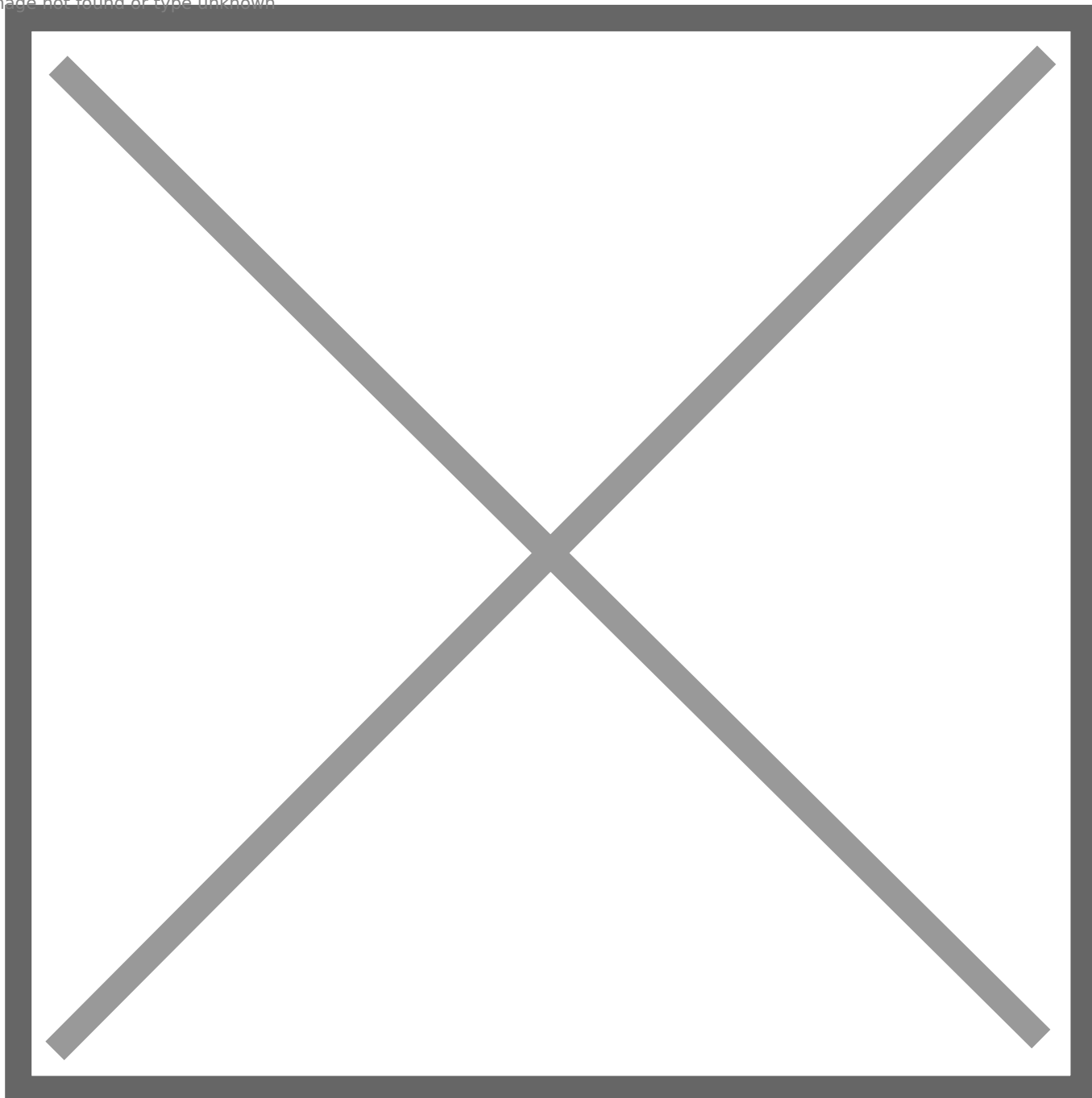
Image not found or type unknown



Các tính năng cơ bản cockpit

Đây là giao diện **Cockpit** khi bạn mở nó trên Ubuntu 20.04 LTS. Nó cung cấp các tính năng chính để quản lý toàn bộ máy chủ Linux của bạn.

Image not found or type unknown



Với các tính năng cơ bản, bạn khắc phục sự cố bằng cách xem nhật ký hoặc start, stop dịch vụ. Bạn cũng có thể định cấu hình lưu trữ HDD, mạng và quản lý tài khoản người dùng Linux. Ngoài ra, các tác vụ khác có thể được thực hiện dễ dàng, như thực hiện cập nhật gói cài đặt hoặc mở một thiết bị đầu cuối dựa trên web khá trực quan. Các bạn tự vô vọc để xem nhé

Image not found or type unknown

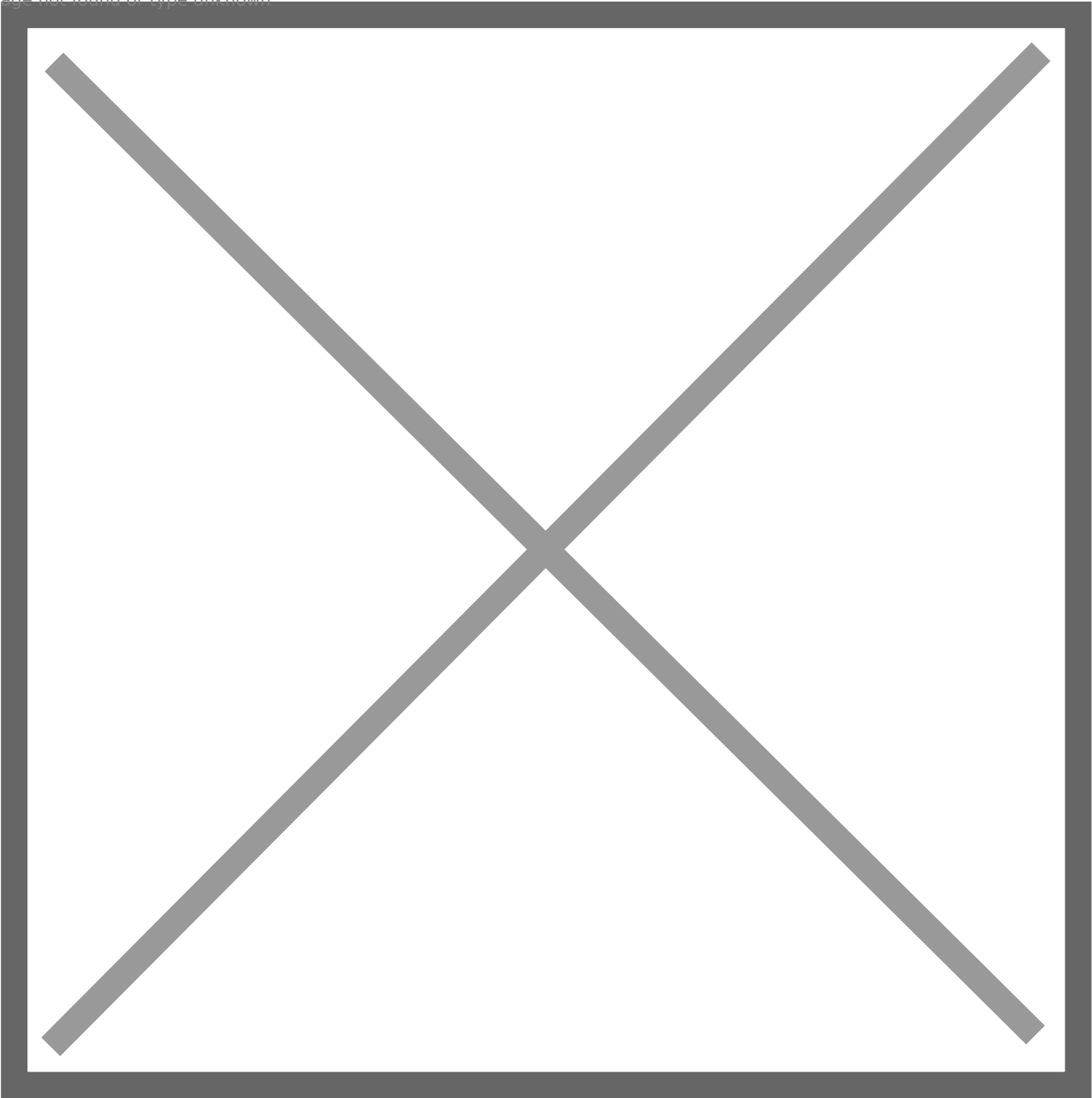
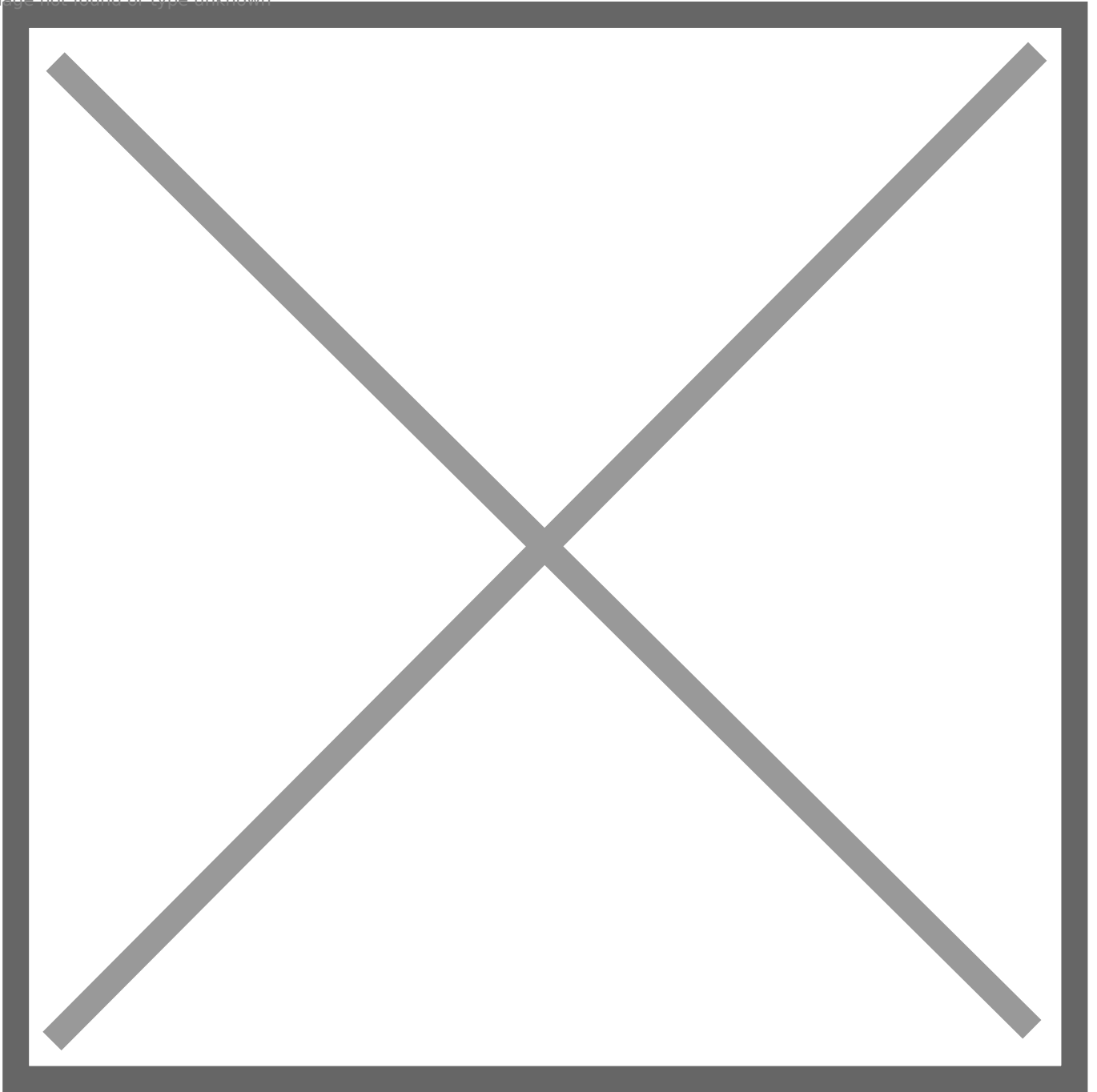


Image not found or type unknown



Cài thêm Cockpit plugin

Bạn cũng có thể cài plugin tùy chỉnh từ cộng đồng. Ở đây mình cài plugin để quản lý các nhóm lưu trữ tên ZFS. Cài xong reload lại web GUI.

```
git clone https://github.com/optimans/cockpit-zfs-manager.git  
sudo cp -r cockpit-zfs-manager/zfs /usr/share/cockpit
```

Image not found or type unknown



Quản lý máy ảo thông qua GUI web Cockpit

Một tính năng khác là quản lý các máy ảo của bạn đang chạy trên KVM. Để kích hoạt tính năng này, bạn cần cài đặt thêm gói. Các bạn thử tính năng này nhé

```
sudo apt install cockpit-machines</pre>
```

Mời cà phê

Nếu cảm thấy blog mang đến những thông tin hữu ích cho công việc, cuộc sống, đam mê của bạn, đừng ngại ủng hộ một ly cà phê để mình có thêm động lực chia sẻ thêm nhiều kinh nghiệm, kiến thức nhé.

Bạn có thể ủng hộ mình qua:

Ví MoMo



[Ví MoMo](#)

Paypal

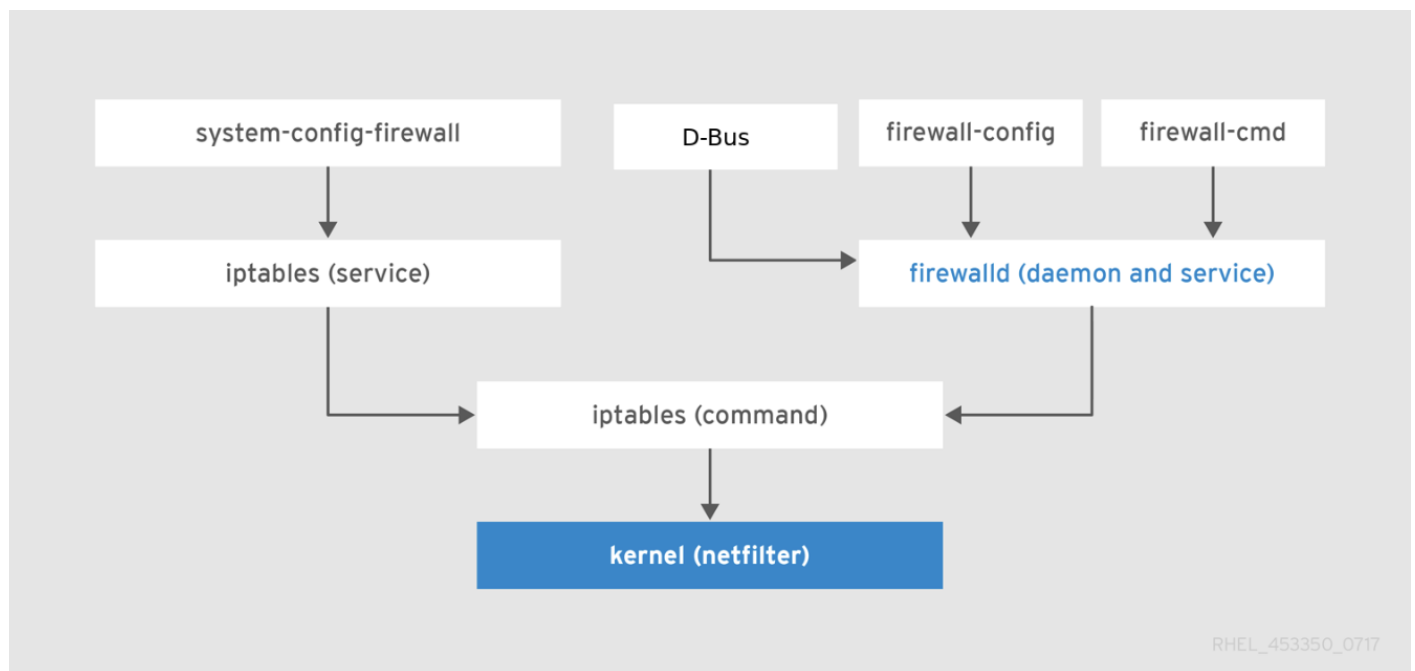
[Hỗ trợ qua Paypal](#)

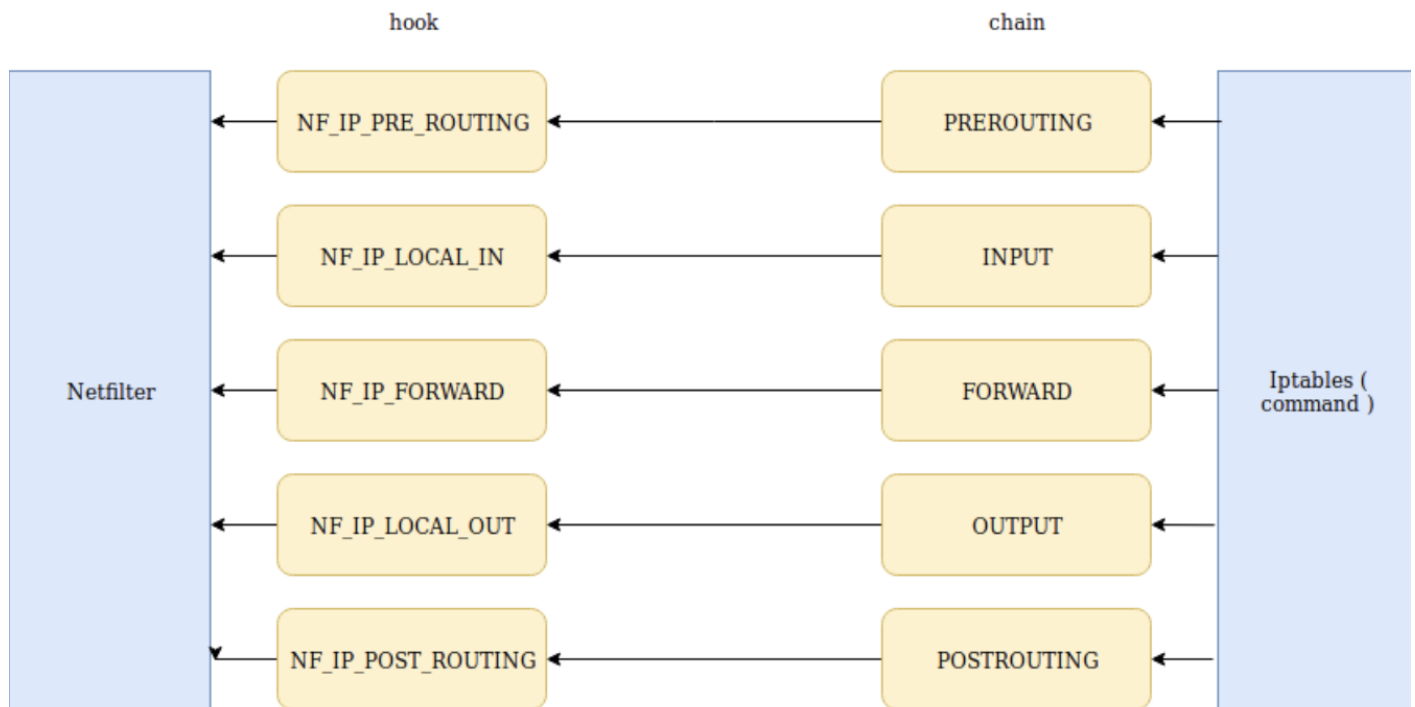
Cám ơn mọi người!

Cài đặt iptables trên Ubuntu

Tổng quan và cài đặt

Iptables chỉ là giao diện dòng lệnh để tương tác với tính năng packet filtering của `netfilter` framework.





Trên **Ubuntu** sử dụng ufw làm firewall nên để cài iptables ta cần disable ufw trước

```
ufw disable
```

Cài đặt iptables

```
apt-get install iptables iptables-persistent
```

Cho phép giữ các phiên đang kết nối để việc thực hiện không bị gián đoạn (trong trường hợp bạn đang SSH vào server)

```
iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

Trong đó:

- **-m conntrack**: Áp dụng cho các kết nối thuộc module "*Connection Tracking*" với 4 kiểu kết nối là **NEW**, **ESTABLISHED**, **RELATED** và **INVALID**. Cụ thể ở rule này chúng ta sẽ sử dụng kiểu **RELATED** và **ESTABLISHED** để lọc các kết nối đang truy cập.
- **-ctstate RELATED,ESTABLISHED**: Khai báo loại kết nối được áp dụng của module "*Connection Tracking*".

Đổi policy mặc định của iptables thành DROP

```
iptables --policy INPUT DROP
```

Lưu lại cấu hình

```
netfilter-persistent save
netfilter-persistent reload
```

Thêm một rule

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -j DROP
```

Trong đó:

- **-p tcp**: Giao thức được áp dụng.
- **-dport 22** và **-dport 80**: Cổng cho phép áp dụng.
- **-j DROP** : Nhưng sẽ khóa toàn bộ các kết nối còn lại.

Kiểm tra rule

```
iptables -nvL
```

Mời cà phê

Nếu cảm thấy blog mang đến những thông tin hữu ích cho công việc, cuộc sống, đam mê của bạn, đừng ngại ủng hộ một ly cà phê để mình có thêm động lực chia sẻ thêm nhiều kinh nghiệm, kiến thức nhé.

Bạn có thể ủng hộ mình qua:

Ví MoMo



[Ví MoMo](#)

Paypal

[Hỗ trợ qua Paypal](#)

Cảm ơn mọi người!

Sử dụng Docker Compose trên Ubuntu 20.04LTS

Cài đặt Docker Compose

```
apt -y install docker-compose
```

Thực hành về Docker Compose

Tạo một Web service container

vi Dockerfile

```
FROM ubuntu
MAINTAINER KienLeTV <admin@skienletv.com>

ENV DEBIAN_FRONTEND=noninteractive

RUN apt-get update
RUN apt-get -y install apache2

EXPOSE 80
CMD ["/usr/sbin/apachectl", "-D", "FOREGROUND"]

# define application configuration
root@kltv:~# vi docker-compose.yml
version: '3'
services:
  db:
    image: mariadb
```

```
volumes:
  - /var/lib/docker/disk01:/var/lib/mysql

environment:
  MYSQL_ROOT_PASSWORD: password
  MYSQL_USER: kienletv
  MYSQL_PASSWORD: password
  MYSQL_DATABASE: my_db

ports:
  - "3306:3306"

web:
  build: .
  ports:
    - "80:80"
  volumes:
    - /var/lib/docker/disk02:/var/www/html
```

Buid và run

```
docker-compose up -d
```

```
Building web
Step 1/7 : FROM ubuntu
---> 7e0aa2d69a15
Step 2/7 : MAINTAINER KienLeTV <admin@kienletv.com>
---> Using cache
---> a2fa80a6247c
Step 3/7 : ENV DEBIAN_FRONTEND=noninteractive
---> Running in cc43f41aa43d
.....
```

```
docker ps
docker-compose ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7b0535cfaf92	mariadb	"docker-entrypoint.s..."	About a minute ago	Up	About a minute	0.0.0.0:3306->3306/tcp root_db_1
73aa26c9ecd3	root_web	"/usr/sbin/apachectl..."	About a minute ago	Up	About a minute	0.0.0.0:80->80/tcp root_web_1

Một số lệnh khác

```
docker-compose logs
docker-compose exec db /bin/bash
docker-compose stop
docker-compose up -d web
docker-compose rm
```

Mời cà phê

Nếu cảm thấy blog mang đến những thông tin hữu ích cho công việc, cuộc sống, đam mê của bạn, đừng ngại ủng hộ một ly cà phê để mình có thêm động lực chia sẻ thêm nhiều kinh nghiệm, kiến thức nhé.

Bạn có thể ủng hộ mình qua:

Ví MoMo



[Ví MoMo](#)

Paypal

[Hỗ trợ qua Paypal](#)

Cám ơn mọi người!

Tạo một SSH key cơ bản

Tạo một SSH key

```
ssh-keygen -t ed25519 -C "Kiên Lê TV"
```

```
root@vps02-kienle:~/openvpn# ssh-keygen -t ed25519 -C "Kiên Lê TV"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:h2TC0ckGL6Uf3F95LIS+NPn8NyhweB+Ea0I7yDJtB9A Kiên Lê TV
The key's randomart image is:
+--[ED25519 256]--+
|      . .      . |
|    0 +  .. o   |
|   + % +...= o  |
|  . E B ==o +   |
|  . o S.+=.     |
|    . . +.oo.   |
|   o o . + ..o  |
|  o = +   . o...|
|   + . .   . .. |
+-----[SHA256]-----+
root@vps02-kienle:~/openvpn#
```

Kiểm tra file được tạo ra

```
ls -l /root/.ssh
```

```
root@vps02-kienle:~/.ssh# ls -l
total 8
-rw----- 1 root root 399 Apr 26 13:00 id_ed25519
-rw-r--r-- 1 root root 94 Apr 26 13:00 id_ed25519.pub
root@vps02-kienle:~/.ssh#
```

Hướng dẫn sử dụng SSH Key

Sau khi đã tạo xong SSH key ở các bước trên, để sử dụng bạn thực hiện tuần tự các bước sau.

Thêm Public key lên Server/VPS

Bạn hãy ssh vào **server/vps** với thông tin passwd root và thực hiện nhập tuần tự các lệnh sau.

```
mkdir ~/.ssh
chmod 700 ~/.ssh
touch ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
```

Khi tạo xong file `authorized_keys` bạn hãy mở file này lên với lệnh sau `vi ~/.ssh/authorized_keys` và copy toàn bộ chuỗi ký tự public key đã tạo vào nhé.

```
root@vps02-kienle:~/.ssh# cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIClA4AJxHe3/764RgXYirjHZWZii945u0QkFjuf0LzJm Kiên Lê TV
root@vps02-kienle:~/.ssh#
```

Sử dụng SSH Key với Linux/MacOS

Với **Linux/MacOS** để sử dụng SSH key bạn mở **Terminal** lên và sử dụng lệnh sau để SSH.

```
ssh -i duong_dan_file_private_key root@ip_may_chu -p nhap_vao_port_ssh
```

Ví dụ: `ssh -i .ssh/kienletv root@123.123.123.123 -p 2222`

Trong đó:

- `ssh -i`: cú pháp thực hiện
- `.ssh/kienletv`: đường dẫn đến file private key, và ở đây private key của mình tên là quan

- root: tên user
- @123.123.123.123: IP máy chủ
- -p 2222: option và Port ssh, ở đây mình đổi port ssh mặc định là 22 thành 2222

Mời cà phê

Nếu cảm thấy blog mang đến những thông tin hữu ích cho công việc, cuộc sống, đam mê của bạn, đừng ngại ủng hộ một ly cà phê để mình có thêm động lực chia sẻ thêm nhiều kinh nghiệm, kiến thức nhé.

Bạn có thể ủng hộ mình qua:

Ví MoMo



[Ví MoMo](#)

Paypal

[Hỗ trợ qua Paypal](#)

Cám ơn mọi người!