Tài liệu hướng dẫn làm các bài thực hành khóa học LPIC-1

I. Thực hành cấu hình một Default Boot Target

- **Giới thiệu bài thực hành**

  Để tham gia thi LPIC-1 exam, chúng ta cần hiểu về làm thế nào để thay đổi defaulttarget cho một hệ điều hành Linux Systemd. Bài thực hành này giúp bạn thực hành để có thể hiểu cách cấu hình sử dụng defaut targer nào cho Linux server của bạn

- **Các bước của bài thực hành:**

1. Kiểm tra Default Target hiện tại của HĐH sử dụng lệnh:

```
systemctl get-default
```

2. Thay đổi Default Target hiện tại của HĐH, sang giao diện dòng lệnh:

```
sudo systemctl set-default multi-user.target
```

3. Kiểm tra lại Default Target hiện tại của HĐH:

```
systemctl get-default
```

4. Thay đổi lại Default Target hiện tại của HĐH, sang giao diện đồ họa:

```
sudo systemctl set-default graphical.target
```

II. Thực hành Cài đặt(install) và quản lý (manage) Packages trong hệ điều hành Debian/Ubuntu

Cài đặt và gỡ bỏ các gói Packages trong Linux distribution, là một kỹ năng quan trọng, đối với một system admin. Trong bài thực hành này, bạn sẽ làm việc với trình quản lý packages và các tiện ích cài đặt apt và dpkg để quản lý các gói trên các bản Ubuntu/Debian Linux.

Các bước của bài thực hành:

1. Cài đặt Apache web server package:

- Cập nhật hệ điều hành: sudo apt update

- Cài đặt các packages: sudo apt install apache2 wget

2. Xác nhận Apache web server đang chạy trong server :

- Kiểm tra Apache web server đang chạy trong HĐH :

curl http://localhost

- Nếu cách kiểm tra trên hoạt động, chúng ta sử dụng lệnh wget để xem đầu ra của yêu cầu http. Chúng ta sử dụng lệnh wget để tải về một file từ Apache vào trong thư mục home, đặt tên file là:

local _index.response:

wget http://localhost > local _index.response

ls -la

mv index.html local_index.response

III. Thực hành sử dụng lệnh sed để sửa đổi một file văn bản

- Giới thiệu:

Một ai đó đã nhầm lẫn và viết từ cows thay vì từ Ants trong file văn bản fable.txt . Chúng ta phải thay thế tất cả các trường hợp của từ " cows " bằng từ " Ants ", dù từ cows có là các chữ cái in hoa hay không.

- Xem nội dung file.

  cat fable.txt
- Sửa chữa lỗi này, chúng ta sẽ chạy một lệnh sed.

  Tùy chọn -i có nghĩa là “Thực hiện việc sửa file," tức là không tạo ra một file khác. Chữ cái I gần cuối có nghĩa là "không phân biệt chữ hoa chữ thường" và có nghĩa là cho dù cows có chứa các chữ cái in hoa hay không, hãy thay đổi nó thành Ants. tùy chọn g có nghĩa là thực hiện việc này toàn cầu, trên toàn bộ file. Dưới đây là lệnh hoàn chỉnh:

  sed -i 's/cows/Ants/Ig' fable.txt

  Sau đó chúng ta chạy lại lệnh cat, chúng ta sẽ thấy rằng tất cả các từ cows đã biến mất.

IV. Tạo cấu trúc thư mục trong Linux

Một kỹ sư quản trị hệ thống Linux(system administrator) cần phải biết cách tạo files và thư mục trong Linux. Bài thực hành này sẽ giúp bạn tạo một cấu trúc thư mục mới và thêm các files mới vào cấu trúc này.

1. Tạo các thư mục cha

Chúng ta có thể tạo cấu trúc thư mục sử dụng ba lệnh sau:

```
[test_user@host]$ mkdir -p Projects/ancient
[test_user@host]$ mkdir Projects/classical
[test_user@host]$ mkdir Projects/medieval
```

- Có thể tạo thư mục sử dụng phương pháp Bash Expansion

Chúng ta có thể tạo các thư mục trên bằng một cách khác, sử dụng Bash Expansion. Việc này sẽ giảm thiểu việc phải dùng nhiều lệnh mkdir. **Lệnh này tạo cùng một lúc ba thư mục:**

```
[test_user@host]$ mkdir -p Projects/{ancient,classical,medieval}
```

2. Tạo các thư mục con

Bây giờ, hãy tạo các thư mục con tiếp theo. Một lần nữa, có hai cách để làm việc này.

Bằng phương pháp thủ công

```
[test_user@host]$ mkdir Projects/ancient/egyptian
[test_user@host]$ mkdir Projects/ancient/nubian
[test_user@host]$ mkdir Projects/classical/greek
[test_user@host]$ mkdir Projects/medieval/britain
[test_user@host]$ mkdir Projects/medieval/japan
```

Với Bash Expansion

```
[test_user@host]$ mkdir Projects/ancient/{egyptian,nubian}
[test_user@host]$ mkdir Projects/classical/greek
[test_user@host]$ mkdir Projects/medieval/{britain,japan}
```

Tạo một số files trống

Chúng ta phải tạo một vài files văn bản trống để các nhân viên phát triển phần mềm sử dụng, vậy chúng ta sẽ sử dụng lệnh touch để tạo files văn bản:

```
[test_user@host]$ touch Projects/ancient/nubian/further_research.txt
[test_user@host]$ touch Projects/classical/greek/further_research.txt
```

3. Đổi tên một thư mục con

Chúng ta nghĩ rằng đã hoàn thành công việc, nhưng đã nhận lại phản hồi từ một số nhân viên của nhóm phát triển là họ cần đổi tên thư mục classical thành greco-roman. Chúng ta có thể làm việc này dễ dàng với lệnh mv:

```
[test_user@host]$ mv Projects/classical Projects/greco-roman
```

Và bây giờ chúng ta bàn giao lại cho phòng phát triển phần mềm.

V. Thực hành làm việc với các files nén

1. Tạo một file có kích thước 10MB, ta có thể chạy lệnh sau:

```
dd if=/dev/zero of=myfile.txt bs=1M count=10
```

Trong lệnh trên, if=/dev/zero sẽ đọc dữ liệu từ thiết bị /dev/zero (nơi mà mọi byte đều là 0). of=myfile.txt sẽ ghi dữ liệu đọc được vào file myfile.txt. bs=1M

sẽ thiết lập kích thước block là 1MB và count=10, sẽ chỉ định số lượng block được tạo ra, do đó tạo ra một file có kích thước 10MB.

2. Lấy Kích Thước File Gốc

- Đầu tiên, hãy xem kích thước ban đầu của File myfile.txt và ghi chú lại:

[test_user@host]$ ls -lh myfile.txt

3. Tạo file Nén ZIP

Gzip

- Đầu tiên, hãy thử nén với Gzip. Lệnh sau sẽ nén junk.txt bằng gzip:

[test_user@host]$ gzip myfile.txt

- Bây giờ, chạy ls để xem kích thước của file:

[test_user@host]$ ls -lh

- Lưu ý rằng lệnh gzip đã thay thế file gốc bằng một phiên bản nén của nó. Các lệnh nén khác mà chúng ta sử dụng cũng sẽ làm như vậy.
- Chú ý đến kích thước nhỏ hơn của file và sau đó giải nén nó để lấy lại phiên bản gốc:

[test_user@host]$ gunzip myfile.txt.gz

Bzip

- Bây giờ chúng ta sẽ thực hiện các bước tương tự, nhưng sử dụng phương pháp nén bzip2:

[test_user@host]$ bzip2 myfile.txt

- Lưu ý phương pháp nén này sẽ mất một chút thời gian hơn so với phương pháp trước. Hãy kiểm tra kích thước file kết quả để xem nó so với việc sử dụng gzip:

[test_user@host]$ ls -lh myfile.txt.bz2

- Nó nên nhỏ hơn myfile.txt.gz.

- Một lần nữa, giải nén file để lấy lại phiên bản gốc:

[test_user@host]$ bunzip2 myfile.txt.bz2

XZ

- Bây giờ chúng ta sẽ thử nghiệm một phương pháp nén mới hơn, XZ. Nó hoạt động với cú pháp tương tự như các phương pháp khác:

[test_user@host]$ xz myfile.txt

- Lưu ý rằng quá trình nén này cũng sẽ mất một chút thời gian. Sau khi lệnh hoàn thành, xem kích thước file của bạn:

[test_user@host]$ ls -lh

- File kết quả có kích thước gần như bằng với file trước đó. Bây giờ, giống như chúng ta đã làm với các file khác, hãy giải nén file:

[test_user@host]$ unxz myfile.txt.xz

- Giải nén file myfile.txt.xz bằng lệnh unxz.

- Tạo File tar

- Tiếp theo, chúng ta sẽ tập trung vào làm việc với file tar. Đầu tiên, chúng ta sẽ sử dụng Gzip để tạo một tarball:

[test_user@host]$ tar -cvzf gztar.tar.gz myfile.txt

- Sau đó, chúng ta sẽ tạo một file tar bằng cách sử dụng bzip2:

[test_user@host]$ tar -cvjf bztar.tar.bz2 myfile.txt

- Cuối cùng, chúng ta sẽ sử dụng XZ để tạo một file tar:

[test_user@host]$ tar -cvJf xztar.tar.xz myfile.txt

- Chạy lệnh ls để so sánh kích thước file:

[test_user@host]$ ls –lh

- Lưu ý rằng tạo file tar không thay thế file junk.txt gốc. Chú ý cách các file xz và bzip2 gần như bằng nhau về kích thước.

4. Thực hành đọc các file văn bản nén

- Nếu chúng ta muốn đọc nội dung của các file đã nén mà không cần giải nén chúng, thì có một cách để làm việc đó! Vậy hãy làm việc đó ngay bây giờ.
- Đầu tiên, chúng ta sẽ sao chép file /etc/passwd vào thư mục home của bạn:

[test_user@host]$ cp /etc/passwd /home/cloud_user/

Gzip

- Chúng ta có thể làm điều tương tự cho một file tar, nén file này bằng Gzip:

[test_user@host]$ tar -cvzf passwd.tar.gz passwd

- Và chúng ta có thể sử dụng lệnh zcat để đọc file này:

[test_user@host]$ zcat passwd.tar.gz

Bzip2

- Bây giờ chúng ta hãy nén file này, sử dụng bzip2, vào một tarball:

[test_user@host]$ tar -cvjf passwd.tar.bz2 passwd

- Chúng ta có thể sử dụng lệnh bzcat để đọc file nén:

[test_user@host]$ bzcat passwd.tar.bz2

XZ

- Cuối cùng, hãy tạo một file tar xz:

[test_user@host]$ tar -cvJf passwd.tar.xz passwd

- Và chúng ta có thể sử dụng lệnh xzcat để đọc nội dung của file:

[test_user@host]$ xzcat passwd.tar.xz

5. Kết luận

Bạn đã hoàn thành lab này, xin chúc mừng!

VI. Thực hành làm việc với Regular Expressions

1. Tìm các dịch vụ HTTP

- Chúng ta muốn đọc tất cả các dòng trong file /etc/services bắt đầu bằng http (nhưng không bao gồm các dòng bắt đầu bằng httpx) và gửi nội dung đến ~/http-services.txt.

- **Sử dụng lệnh sau để làm việc này:**

grep ^http[^x] /etc/services > ~/http-services.txt

Để kiểm tra xem chúng ta đã có những gì trong file mới, chạy lệnh:

cat ~/http-services.txt

2. Tìm các dịch vụ LDAP

- Việc này hơi khó hơn. Chúng ta muốn tìm tất cả các dòng trong /etc/services bắt đầu bằng chuỗi ký tự ldap.
- Ký tự thứ năm có thể là bất kỳ ký tự chữ hoặc số nào, nhưng ký tự thứ sáu không có ký tự là a. Chúng ta sẽ đưa đầu ra vào ~/lpic1-ldap.txt.

grep ^ldap.[^a] /etc/services > ~/lpic1-ldap.txt

- **Để kiểm tra xem chúng ta đã có những gì trong file mới, chạy lệnh:**

cat ~/lpic1-ldap.txt

3. **Tinh chỉnh kết quả HTTP**

- Chúng ta muốn đọc file ~/http-services.txt mà chúng ta đã tạo trước đó và chỉ xem các dòng không kết thúc bằng từ "service".
- **Lệnh grep sau sẽ giúp chúng ta làm việc này:**

grep -v service$ ~/http-services.txt > ~/http-updated.txt

- **Để kiểm tra xem chúng ta đã có những gì trong file mới, chạy lệnh:**

cat ~/http-updated.txt

4. **Kết luận**

Chúng ta đã hoàn thành và giờ đây chúng ta có một kiến thức cơ bản về việc sử dụng các biểu thức chính quy đơn giản để tìm thông tin cần thiết từ các file văn bản.
Chúc mừng bạn!

VII. Thực hành sử dụng Vim để tạo và sửa đổi file

1. **Tạo một file mới:**

- **Di chuyển đến thư mục home**
- Mở file notes.txt với lệnh "vim notes.txt"

- Nhập vào văn bản "Beginning of Notes File" bằng cách chuyển sang chế độ chèn (insert mode) bằng phím "i", sau đó nhập văn bản vào file.
- Để lưu và thoát khỏi Vim, nhấn Esc để thoát khỏi chế độ chèn, gõ ":wq!" và nhấn Enter.

2. **Ghi dữ liệu vào file notes.txt:**

- Sử dụng lệnh "cat /etc/redhat-release >> notes.txt" để ghi nội dung của tệp /etc/redhat-release vào cuối file notes.txt mà không ghi đè vào nội dung của file.

- **Sửa file notes.txt:**

  Mở lại file notes.txt với lệnh "vim notes.txt".
  Sử dụng các phím mũi tên để di chuyển đến dấu ngoặc đầu tiên trước từ "CentOS ".
  Xóa văn bản từ vị trí con trỏ đến cuối dòng bằng phím "SHIFT D" hoặc "d$".
  Tạo hai dòng trống ở cuối file bằng cách nhấn phím "o" hai lần.
  Lưu và thoát khỏi Vim bằng cách gõ ":wq!".

3. **Ghi thêm dữ liệu vào file notes.txt và sửa nội dung:**

- Sử dụng lệnh "free -m >> notes.txt" để ghi đầu ra của lệnh free -m vào cuối file notes.txt.
- Mở lại file notes.txt với lệnh "vim notes.txt".
  Di chuyển đến dòng "Swap" bằng các phím mũi tên.
  Xóa dòng này bằng cách gõ "dd".
  Tạo một dòng trống dưới con trỏ bằng phím "o".

  Nhập "This is a practice system." và tạo một dòng trống bằng phím "Enter".
- Lưu và thoát khỏi Vim bằng cách gõ ":wq!".

4. **Hoàn thành file ghi chú notes.txt:**

- Ghi thêm đầu ra của lệnh "dbus-uuidgen --get" vào cuối file notes.txt bằng lệnh "dbus-uuidgen --get >> notes.txt".
- Mở lại file notes.txt với lệnh "vim notes.txt".
  Đi đến cuối file bằng phím "G".
  Nhập "Dbus ID = " vào đầu dòng mới, để bắt đầu một dòng mới và để khoảng trắng giữa dấu bằng và đầu ra của lệnh "dbus-uuidgen --get".
- Lưu và thoát khỏi Vim bằng cách gõ ":wq!".

VIII. Thực hành cấu hình gắn thêm một ổ cứng mới vào hệ điều hành Linux

1. **Tạo một phân vùng mới Trước khi chúng ta gắn (mount) một phân vùng mới lên, chúng ta phải tạo phân vùng đó trước.**

Mở một cửa sổ terminal và đăng nhập vào Linux server của bạn

$ ssh user@x.x.x.x Nhập mật khẩu

- Tiếp theo, chúng ta chạy lệnh lsblk để xác nhận xem chúng ta có thiết bị /dev/nvme0n2 có sẵn hay không.
- Sau khi xác nhận, chúng ta sẽ tạo một phân vùng trên đĩa /dev/nvme0n2 bằng lệnh fdisk.
- **Lưu ý:** Chúng ta sẽ cần sử dụng sudo trước các lệnh này. Phân vùng chúng ta tạo sẽ chiếm toàn bộ ổ đĩa:

[user@host]$ lsblk

[user@host]$ sudo fdisk /dev/nvme0n2

Lưu ý: Bạn cần phải xem kỹ kết quả của lệnh lsblk để biết được tên ổ mà bạn mới gắn vào máy ảo Linux của bạn

- Sau khi chạy fdisk, chúng ta sẽ phải thực hiện một số công việc.
- Tại lệnh Command (m for help):, nhập n để tạo một phân vùng mới, sau đó nhấn Enter.
- Loại phân vùng của chúng ta sẽ là p, primary. Nhấn Enter cho các tùy chọn Partition number, First sector và Last sector.
- Việc này sẽ làm cho fdisk sẵn sàng để tạo phân vùng.
- Gõ p tại lệnh Command (m for help): Để in ra dạng của ổ sau khi chúng ta thực hiện các thay đổi.
- Nếu tất cả đã đúng, gõ w và nhấn Enter để ghi các thay đổi lên ổ đĩa.

2. **Tiếp theo tạo filesystem, chúng ta phải tạo một filesystem để chúng ta có thể đọc và ghi dữ liệu.**

- Chúng ta sẽ định dạng phân vùng thành filesystem XFS bằng lệnh mkfs.xfs. Sau khi hoàn tất việc này, chúng ta sẽ chạy blkid trên phân vùng vừa tạo để lấy UUID.

- **Chúng ta phải ghi lại thông tin UUID , vì chúng ta sẽ cần nó sau này:**

[user@host]$ sudo mkfs.xfs /dev/nvme0n2

[user@host]$ sudo blkid /dev/nvme0n2

3. **Gắn filesystem mới :**

- Chúng ta có thể gắn phân vùng này bằng cách thủ công sử dụng lệnh mount, nhưng nó sẽ không được gắn tự động; nó sẽ không được gắn sau khi khởi động lại chẳng hạn.

- Chúng ta sẽ chỉnh sửa file /etc/fstab và tạo một mục mới cho ổ mới ở cuối file.

  sudo vi /etc/fstab

- Khi bạn muốn them dòng văn bản: nhấn phím esc và sau đó gõ i để chuyển sang chế độ chèn và nhập bình thường.

- Khi bạn muốn lưu: nhấn phím esc và sau đó gõ :wq!

- Định dạng nên tuân theo định dạng như sau (hãy chắc chắn sử dụng UUID của ổ từ bước trước):

  UUID=YOURUUID /opt xfs defaults 0 0

- Chúng ta có thể lưu file (:wq!), Sau đó chạy:

  [user@host]$ sudo mount -a

- Lệnh trên sẽ gắn (mount)tất cả những gì được liệt kê trong fstab, bao gồm phân vùng mới của chúng ta.

- Và chạy lệnh df -h /opt sẽ hiển thị khoảng 10 GB dung lượng ổ mới cho thư mục /opt.

IX. Quản lý thuộc tính và quyền truy cập files

Mục tiêu:

Thiết lập lại quyền truy cập cho một thư mục với các quyền sau:

Tất cả mọi người đều có thể truy cập vào thư mục, tất cả mọi người có thể đọc các file trong thư mục. Không ai có thể thực thi các file trong thư mục, áp dụng tất cả các quyền trên cho tất cả các thư mục con theo cách đệ quy.

1. **Sử dụng 1 user có quyền sudo để Đăng nhập vào Linux server của các bạn:**

ssh user@ip_server

Sau đó sử dụng lệnh sau để vào quyền root:

sudo -i

2. **Tạo thư mục và files (môi trường) cho bài thực hành như sau:**

mkdir /opt/myapp

mkdir /opt/myapp/testapi

touch /opt/myapp/testapifile.cc

touch /opt/myapp/testapi/api.sh

echo "ls -al ." > /opt/myapp/testapi/api.sh

- **Chạy lệnh:**

[root@localhost ~]# /opt/myapp/testapi/api.sh

- **Sẽ thấy báo không có quyền chạy file: "Permission denied"**

3. **Cấp quyền cho file chạy:**

chmod u+x /opt/myapp/testapi/api.sh

Chạy lại file chạy:

[root@localhost ~]# /opt/myapp/testapi/api.sh

Chúng ta có thể thấy kết quả trả về.

- **Cấp quyền cho thư mục myapp chỉ cho phép user root mới có thể truy cập được vào thư mục này:**

chmod 600 /opt/myapp

- **Sử dụng lệnh exit để thoát ra khỏi quyền root:**

exit

4. **Giải pháp:**

Cấp quyền truy cập vào thư mục:

- Chuyển tới thư mục opt bằng lệnh: cd /opt
- Mở tất cả các files và quyền hạn của thư mục bằng lệnh: ls –la
- Thử mở thư mục myapp bằng lệnh cd myapp/
- Nhận thông báo "Permission denied" do hiện tại các quyền truy cập bị giới hạn cho người dùng user của bạn.
- Hãy thay đổi quyền bằng cách nhập lệnh: sudo chmod 777 myapp và nhập mật khẩu khi được yêu cầu.

- Mở lại file và quyền hạn của thư mục bằng lệnh ls -la.
- Sau đó, thử mở lại thư mục bằng lệnh cd myapp và chúng ta đã có thể mở được thư mục.
- Thay đổi quyền truy cập cho thư mục:
- Bước tiếp theo là cấp quyền đọc và ghi cho tất cả người dùng cho thư mục này. Tuy nhiên, chúng ta cũng cần đảm bảo không ai có thể thực thi file trong thư muc.
- Hãy bắt đầu bằng cách xóa quyền thực thi bằng lệnh: sudo chmod -f -x -R *.
- Sau đó, cấp quyền đọc và ghi cho tất cả mọi người bằng lệnh: sudo chmod 666 -f -R *.
- Lưu ý là việc này cũng sẽ xóa quyền thực thi mà chúng ta đã xóa ở bước trước,
- nhưng chúng ta muốn hiển thị cách làm đó rõ ràng.
- Liệt kê lại files và quyền hạn của thư mục bằng lệnh: ls -la. Chúng ta có thể thấy rằng tất cả mọi người đều có quyền đọc và ghi.
- **Lưu ý:** Để cho phép người dùng có thể truy cập được vào các thư mục, các thư mục phải được thiết lập phân quyền đúng.

X. Thực hành tạo Hard Links và Soft link trong Linux

1. Tạo một liên kết mềm (symbolic link)

- Tạo một liên kết mềm (symbolic link) từ tập tin /etc/redhat-release đến tập tin liên kết mới có tên
- **Là release trong thư mục home của người dùng phuongluuho:**

  ln -s /etc/redhat-release release

- Xác minh rằng liên kết là hợp lệ:
  ls –l
- **Thử xem bạn có thể đọc nội dung của file này không:**

  cat release

- **Xem bạn có thể đọc nội dung của liên kết không:**

  cat /etc/redhat-release

- Cả 2 files này là giống nhau.
- Kiểm tra số inode cho liên kết
- Xem số inode của /home/name_user/release:

  ls -i release

- **Kiểm tra số inode cho /etc/redhat-release:**

  ls -i /etc/redhat-release

- Inodes là khác nhau, vì liên kết mềm chỉ là một điểm vào file mới trỏ đến file gốc.

2. Tạo một liên kết cứng

- **Tạo một thư mục gọi là docs :**

  mkdir docs

- **Sao chép /etc/services vào thư mục docs :**

  cp /etc/services docs/

- **Tạo một liên kết cứng từ tập tin /home/name_user/docs/services đến vị trí liên kết mới có tên /home/name_user/services:**

  ln docs/services services

- **Xác minh số inode cho liên kết cũng như số inode cho file /etc/services gốc:**

  ls -l

- Việc này sẽ thấy đây là một liên kết cứng, không phải là liên kết mềm. Vì nó sẽ không có mũi tên trỏ đến file thực sự mà nó được liên kết đến, như một liên kết mềm. để xác nhận, hãy kiểm tra hai file này bằng lệnh cat và đảm bảo chúng giống nhau:

- **Xem nội dung của các số inode:**

  ls -i services

  ls -i docs/services

- Bạn sẽ thấy 2 files có cùng số inode, có nghĩa là chúng là cùng một files.

3. Tạo một liên kết cứng giữa các phân vùng filesystems

- **Xem các ổ và phân vùng trong server:**

  Lsblk

- Như các bạn đã biết thì. Trong bài học về Mount và Unmount Filesystems, chúng ta đã thực hiện mount phân vùng sdb1 vào thư mục /opt

  xvda 202:0 0 10G 0 disk

  ├─xvda1 202:1 0 1M 0 part

  └─xvda2 202:2 0 10G 0 part /

  xvdb 202:16 0 2G 0 disk

  └─xvdb1 202:17 0 2G 0 part /opt

- Tạo một Hard Link qua các File System khác nhau
- Xem các thiết bị khối riêng lẻ:

  lsblk -f

- **Bạn sẽ thấy một cái gì đó tương tự như sau:**

  NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

  xvda 202:0 0 10G 0 disk

```
├─xvda1 202:1 0 1M 0 part
└─xvda2 202:2 0 10G 0 part /
xvdb 202:16 0 2G 0 disk
└─xvdb1 202:17 0 2G 0 part /opt
```

- Bạn có thể thấy ở đây / và /opt nằm trên hai phân vùng riêng biệt. Vì mỗi phân vùng có bộ inode riêng của nó, nên hard link qua các phân vùng không hoạt động được. Soft link thì có thể hoạt động được.
- **Thử tạo một hard link từ /home/cloud_user/docs/services đến /opt/services:**

```
ln /home/name_user/docs/services /opt/services
```

- Bạn sẽ nhận được thông báo lỗi "failed to create hard link".
- Cố gắng tạo một Soft Link qua các File System khác nhau
- Thử tạo cùng một loại liên kết qua các phân vùng khác nhau, sử dụng tùy chọn -s để tạo một soft link:

```
sudo ln -s /etc/redhat-release /opt/release
```

- Nếu không có bất kỳ báo lỗi nào, điều này có nghĩa là tạo soft link đã thành công.
- **Xem lại nội dung của inode:**
- **Xem số inode của /etc/redhat-release và /opt/release:**

```
ls -i /etc/redhat-release
ls -i /opt/release
```

- Bạn sẽ thấy là 2 files này có các inode khác nhau, nhưng liên kết vẫn hoạt động.

XI. Thực hành tạo hàm(Function) và Alias

1. **Tạo môi trường cho bài thực hành:**
   - **Cài đặt apache web server:**

     ```
     sudo yum install httpd
     ```

   - **Tạo thư mục main trong thư mục html:**

     ```
     sudo mkdir /var/www/html/main
     ```

   - **Download file wordpress vào thư mục/var/www/html/main:**

     ```
     wget http://wordpress.org/latest.tar.gz
     ```

2. **Tạo Alias**

- Bước đầu tiên là tạo một alias cho shell Bash để cho phép xem trạng thái dịch vụ của webserver. Bạn sẽ đặt tên cho alias này là "webstat".
- Khi bạn chạy lệnh "webstat" , bạn sẽ thấy kết quả của lệnh "systemctl status httpd.service".
- Các alias và hàm được tạo bởi người dùng sẽ được lưu trong file "/.bashrc". thêm alias sau vào file "/.bashrc" của bạn:

```
echo 'alias webstat="systemctl status httpd.service"' >> /home/your_user/.bashrc
```

3. Tải và kiểm tra alias

- Bây giờ chúng ta đã tạo một alias để hiển thị trạng thái của webserver, chúng ta cần cho Bash biết là chúng ta muốn sử dụng nó trong phiên làm việc hiện tại. Đầu tiên, chúng ta cần làm mới lại file "~/.bashrc" của bạn bằng lệnh "dot" (.):

```
. ~/.bashrc
```

- Sau khi môi trường Bash được làm mới với alias mới từ file "~/.bashrc", chúng ta có thể sử dụng alias mới của chúng ta:

```
webstat
```

- Chúng ta có thể thấy kết quả của lệnh, hiển thị trạng thái dịch vụ webserver đang chạy.

4. Tạo hàm

- Bước tiếp theo là tạo một hàm để hàm này sẽ lấy tên của một thư mục làm tham số và in ra tổng dung lượng mà thư mục đó đang sử dụng.
- Sử dụng trình soạn thảo văn bản vi, mở file "~/.bashrc" và thêm hàm sau vào cuối, phía dưới alias bạn đã tạo trước đó:

```
vim ~/.bashrc

function webspace()

{

du -h /var/www/html/$1;

}
```

- Lưu và đóng file của bạn. Sau đó, làm mới file ".bashrc" một lần nữa:

```
. .bashrc
```

5. Sử dụng hàm webspace

- Vì thư mục "/var/www/html" là nơi gốc cho tất cả các vị trí trang web cá nhân cho máy chủ web này, bạn chỉ cần cung cấp tên thư mục chứa một phần cụ thể của trang web cho hàm webspace. Để xem kích thước và nội dung của trang web, chạy lệnh này:

```
webspace main
```

- Lệnh này sẽ in ra nội dung của thư mục "/var/www/html/main" và tổng dung lượng của thư mục

XII. Thực hành tạo Users, Groups và quản lý User Accounts

1. **Cài đặt trình soạn thảo văn bản Nano:**

```
sudo yum install -y nano
```

- Nhập mật khẩu của bạn tại nhắc lệnh.
- Tạo một thư mục mới cho việc chia sẻ tài nguyên files & thư mục có tên là /test_scripts.

```
sudo mkdir /test_scripts
```

- Chuyển đến thư mục gốc và liệt kê nội dung chi tiết.

```
cd /
ls -la
```

- Thiết lập Quyền cho user chủ sở hữu và Nhóm group
- Cấp đầy đủ quyền cho chủ sở hữu và nhóm và thu hồi quyền cua các users khác.

```
sudo chmod 770 test_scripts/
```

- Liệt kê lại nội dung của thư mục.

```
ls -la
```

2. **Tạo 1 nhóm có tên là testers**:

```
groupadd testers
```

- Thay đổi quyền sở hữu nhóm của thư mục test_scripts thành nhóm testers.

```
sudo chgrp testers test_scripts/
```

- Thêm người dùng mới vào hệ ĐH
- Chuyển đến thư mục home.

```
cd /home
```

- Thêm người dùng binhtd.

```
sudo adduser -m binhtd
```

- Xác nhận là thư mục binhtd đã được tạo.

```
ls -la
```

- Kiểm tra nội dung của thư mục binhtd.

```
sudo ls -la binhtd/
```

- Thêm người dùng giangnh, duccm và tungnm.

  sudo adduser -m giangnh

  sudo adduser -m duccm

  sudo adduser -m tungnm

- Tạo mật khẩu cho user duccm:
  passwd duccm
- Xác minh rằng việc này đã thành công.

  ls -la

3. Thêm các người dùng mới vào một nhóm

- Thêm người dùng vào file /etc/group.

  sudo nano /etc/group

- Trong trình soạn thảo văn bản Nano, tìm dòng bắt đầu bằng testers:x:.
- Thêm binhtd, giangnh, duccm, tungnm vào cuối dòng:

  testers:x:1002:binhtd, giangnh, duccm, tungnm

- **Lưu ý** là GID của testers có thể không phải là 1002
- Nhấn Ctrl + X để thoát khỏi trình soạn thảo Nano.
- Nhấn Y sau đó Enter để lưu các thay đổi.
- Chuyển sang người dùng duccm

  sudo su duccm

- Chuyển đến thư mục /test_scripts/.

  cd /test_scripts

- Liệt kê nội dung của thư mục.

  ls -la

- Tạo một file mới để kiểm tra.

  touch test.txt

- Xác nhận là File mới đã được tạo.

  ls -la

- Chuyển sang một người dùng có trong server của bạn ví dụ user: phuonglh.

  exit

  sudo su phuonglh

- Chuyển đến thư mục gốc.

  cd /

- Liệt kê nội dung của thư mục.

  ls -la

- Chuyển đến thư mục test_scripts.

  cd test_scripts/

- Bạn sẽ nhận được thông báo "Permission denied".

XIII. Thực hành lập lịch cho các tác vụ systemd với Timer Units.

1. **Chúng ta cần phải đăng nhập với quyền root để hoàn thành bài thực hành. Đăng nhập vào root bằng lệnh sudo:**

   [user@$host ~]$ sudo su –

- **Trước hết,Tạo thư mục để lưu file backup:**
  mkdir /var/backups
- Tiếp theo, chúng ta tạo 1 file kịch bản script cho thực hiện việc backup thư mục
- Để tạo nội dung file web-backup.sh, bạn có thể làm như sau:
  - Mở trình soạn thảo văn bản, ví dụ như nano hoặc vim.
  - **Tạo một tập lệnh để sao lưu thư mục Documents trong thư mục /root :**
    sudo nano /usr/local/sbin/web-backup.sh
  - **Tạo, và Lưu file script này với tên là web-backup.sh trong thư mục /usr/local/sbin/. Với nội dung như sau:**

    #!/bin/bash

    # This script backs up the webapp directory to /var/backups/webapp

    tar -czvf /var/backups/etc_$(date +%Y-%m-%d_%H-%M-%S).tar.gz /etc/

- Phân quyền cho file này có thể chạy thực thi, bằng cách sử dụng lệnh:
  chmod +x /usr/local/sbin/web-backup.sh

2. **Bước 2 là chúng ta tạo một file dịch vụ để chạy script backup trên, với đuôi .service trong thư mục /etc/systemd/system/, như sau:**

   sudo vi /etc/systemd/system/web-backup.service

- **file service backup có nội dung như sau:**

  [Unit]

  Description=Backup the web site, hàng ngày, nếu không làm sếp mắng.

[Service]

Type=simple

ExecStart=/usr/local/sbin/web-backup.sh

[Install]

WantedBy=multi-user.target

- **Lưu và đóng file. Sau đó cấp quyền thực thi cho file backup script:**

  sudo chmod +x /usr/local/sbin/web-backup.sh

- Khởi động lại systemd để cập nhật các thay đổi

  sudo systemctl daemon-reload

- Kích hoạt service backup để nó chạy mỗi khi hệ thống khởi động.

  sudo systemctl enable web-backup.service

3. Bước cuối cùng: Tạo một file đơn vị Timer

- Sau khi đã có tài nguyên, chúng ta sẵn sàng tạo tệp đơn vị Timer. Để làm điều này, sử dụng lệnh vi cùng với web-backup.timer:

[root@$host ~]# vi web-backup.timer

Điền thông tin như sau:

[Unit]

Description=Fire off the backup

[Timer]

OnCalendar=*-*-* 23:00:00

Persistent=true

Unit=web-backup.service

[Install]

WantedBy=multi-user.target

Lưu ý: 23:00:00 có thể được thay đổi thành giờ khác. Ở đây, Tôi chỉ lấy 11 giờ tối làm ví dụ.

Khi file đúng, hãy ghi và thoát đúng cách ra khỏi vi bằng phím Esc, : (hai chấm), sau đó w, sau đó q.

- Đặt file vào đúng vị trí. Sau đó, sử dụng lại lệnh cp, sao chép cả file timer vào /etc/systemd/:

  [root@$host ~]# cp web-backup.timer /etc/systemd/system/

- Để systemd chạy các file. Sau khi các files của chúng ta được đặt đúng vị trí, chúng ta cần tải lại systemd daemon để nó có thể tính toán các thành phần phụ thuộc của dịch vụ:

  [root@$host ~]# systemctl daemon-reload

- **Bây giờ cho phép dịch vụ chạy khi khởi động:**

  [root@$host ~]# systemctl enable web-backup.timer

- Sau khi tạo các files và đặt vào đúng vị trí, chúng ta cần cho systemd biết để chạy chúng. Để làm việc này, chúng ta cần tải lại hệ thống dịch vụ systemd để tính toán các phụ thuộc của dịch vụ:

  [root@$host ~]# systemctl daemon-reload

- **Sau đó, chúng ta cần kích hoạt(bật) các dịch vụ để chúng được chạy khi khởi động:**

  [root@$host ~]# systemctl enable web-backup.service

  [root@$host ~]# systemctl enable web-backup.timer

- Sau khi các liên kết tượng trưng được tạo ra, hãy khởi động các dịch vụ theo cách thủ công:

  [root@$host ~]# systemctl start web-backup.timer web-backup.service

- Kiểm tra trạng thái của cả timer và dịch vụ:

  [root@$host ~]# systemctl status web-backup.timer

  [root@$host ~]# systemctl status web-backup.service

- Cả hai đều hiển thị đang chạy, như vậy chúng ta có thể backup dữ liệu cho nhóm phát triển phần mềm.

XIV. Thực hành điều tra System Service log files để xử lý lỗi sử dụng Journal Control

1. **Login vào máy Linux của bạn sử dụng lệnh:**

   ssh root@ip-server

- Cài đặt apche web server:
  yum install httpd
- Hãy khởi động dịch vụ web.

  systemctl start httpd.service

2. **Backup file cấu hình của apache webserver :**

   cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.bak

- **Xóa file cấu hình của apache web server để giả lập là dịch vụ này bị lỗi:**

  rm -rf /etc/httpd/conf/httpd.conf

- Khởi động lại dịch vụ:

  systemctl restart httpd.service

- Kiểm tra trạng thái của dịch vụ web.

  systemctl status httpd.service

- Cố gắng khởi động dịch vụ web.

  systemctl start httpd.service

- Sau khi dịch vụ không khởi động được, kiểm tra journal.

  journalctl -u httpd.service

- Kiểm tra thư mục chứa file cấu hình httpd .

  ls /etc/httpd/conf

- Khôi phục lại file cấu hình httpd gốc.

  mv /etc/httpd/conf/ httpd.conf.bak /etc/httpd/conf/httpd.conf

- Khởi động lại dịch vụ.

  systemctl restart httpd.service

3. **Xác nhận là dịch vụ máy chủ web đang chạy.**

- Cài đặt trình duyệt web :

  sudo yum install --enablerepo=powertools elinks -y

- Kiểm tra trạng thái của dịch vụ.

systemctl status httpd.service

- Thử truy cập vào trang web apache local.

lynx http://localhost

XV. Thực hành Cấu hình Thiết lập Chuyển tiếp Email Nội bộ

1. **Nếu trên Linux server của bạn chưa có dịch vụ MTA postfix, hãy setup sử dụng lệnh sau để cài đặt:**

sudo yum install postfix

- Sau đó khởi động dịch vụ:

sudo systemctl start postfix

- **Đầu tiên cần phải đăng nhập vào user root:**
- Tạo User system admin:

useradd -m admin

- Cấp quyền sudo user cho user này:

usermod -aG wheel admin

- Đặt password cho user:

passwd admin

2. **Cấu hình Alias**

[root@host]$

- Chạy lệnh sau để thêm một alias email của người dùng root, để tất cả các emails của user root sẽ được gửi đến tài khoản của user system admin thay vì user root:

[root@host]# echo "root: admin" >> /etc/aliases

- Tạo lại Aliases Database File
- Chạy lệnh sau để tạo lại một file /etc/aliases.db mới để quy tắc aliases email mới của bạn được ghi lại:

[root@host]# newaliases

3. **Gửi Email Thử nghiệm**

- Chúng ta cần tạo một email để kiểm tra quy tắc chuyển tiếp forwarding mới của bạn. Lệnh này sẽ gửi một bản sao của file /etc/services dưới dạng đính kèm email, cùng với dòng chủ đề "Root Testing", đến user root:

[root@host]# mail -s "Root Testing" -a "/etc/services" root@localhost < /dev/null

- Chúng ta đang sử dụng một email trống, vì chúng ta chỉ kiểm tra file đính kèm. Đọc Email, Sau đó Xóa nó
- Login vào tài khoản admin và chạy lệnh mail để xem email của admin user:

[root@host]# su –l admin

[admin@host]$ mail

- Thông báo được gửi từ người dùng root sẽ được hiển thị trong danh sách email. Nhấn số tương ứng với email (thường là 1) sau đó nhấn Enter. Sau khi chúng ta đã đọc xong email (nhấn space để cuộn đến file đính kèm), nhấn q để đóng email và quay lại giao diện chính của mail. Nhấn d tại dấu "&" để xóa email. Cuối cùng, nhấn q để thoát khỏi ứng dụng mail.

XVI. Thực hành CUPS Print Server

1. Cài đặt dịch vụ CUPS printer

sudo apt-get install cups printer-driver-cups-pdf

- Cài đặt Máy in PDF
- Mở ứng dụng terminal của bạn.
- Kiểm tra xem có bao nhiêu máy in đã được cài đặt.

lpstat -s

- Kiểm tra xem các loại kết nối (socket) máy in nào có sẵn.

sudo lpinfo -v

- Cài đặt máy in PDF để sử dụng với CUPS.

sudo lpadmin -p CUPS-PDF -v cups-pdf:/

- Xác định các file driver mà chúng ta có thể sử dụng với máy in của mình bằng cách truy vấn cơ sở dữ liệu CUPS để tìm các files chứa "PDF".

lpinfo --make-and-model "PDF" -m

- Sử dụng file driver CUPS-PDF_opt.ppd.

sudo lpadmin -p CUPS-PDF -v cups-pdf:/ -E -P /usr/share/ppd/cups-pdf/CUPS-PDF_opt.ppd

- Chạy lại lệnh lpstat.

lpstat -s

- Kiểm tra trạng thái của máy in vừa cài đặt.

lpc status

- Kích hoạt máy in để chấp nhận công việc, và đặt nó làm máy in mặc định.

```
sudo lpadmin -d CUPS-PDF -E
sudo cupsenable CUPS-PDF
sudo cupsaccept CUPS-PDF
```

- Chạy lại lệnh lpc status.

```
lpc status
```

2. Xác nhận Máy in trên đã sẵn sàng.

- In thử một Trang để Kiểm Tra máy in
- In một bản sao của file /etc/passwd đến file PDF trong thư mục home của user.

```
lpr /etc/passwd
```

- Xác nhận là đã có một bản sao của tập tin /etc/passwd trong thư mục home.

```
ls
```

- Chỉnh sửa Máy in và Làm việc với Hàng đợi In
- Cấu hình máy in sao cho không chấp nhận các công việc in mới.

```
sudo cupsreject CUPS-PDF
```

- Xác nhận trạng thái của máy in.

```
lpc status
```

- Thử Cố gắng in file /etc/group vào máy in.

```
lpr /etc/group
```

- Bạn sẽ nhận được thông báo cho biết máy in hiện không chấp nhận công việc in.

3. Cấu hình lại máy in để chấp nhận công việc in.

```
sudo cupsaccept CUPS-PDF
```

- Kiểm tra lại trạng thái của máy in.

```
lpc status
```

- Cấu hình máy in để chấp nhận công việc in vào hàng đợi nhưng không in văn bản.

```
sudo cupsdisable CUPS-PDF
```

- Kiểm tra lại trạng thái của máy in.

```
lpc status
```

- Cố gắng in file /etc/group lần nữa.

lpr /etc/group

- Liệt kê nội dung của thư mục /home.

ls

4. **Kiểm tra hàng đợi của máy in.**

Lpq

- Xóa công việc khỏi hàng đợi của máy in (nhớ thay thế JOB_ID bằng mã công việc từ đầu ra của lệnh của bạn).

lprm <JOB_ID>

- Xác nhận rằng công việc đã được xóa thành công khỏi hàng đợi của máy in.

lpq

- Bật lại khả năng in của máy in.

sudo cupsenable CUPS-PDF

- Kiểm tra lại trạng thái của máy in CUPS-PDF để đảm bảo là nó đã sẵn sàng nhận công việc in mới.

lpq

XVII. Thực hành giám sát mạng

Các bước trong bài thực hành như sau:

1) Cài đặt các công cụ clients trên 2 servers

2) Tạo file log chứa thông tin lưu lượng mạng

1. **Cài đặt các công cụ trên 2 máy Linux server làm máy chủ và máy client**

- Chúng ta phải cài đặt hai công cụ mà nhóm phát triển phần mềm sẽ sử dụng để tạo và theo dõi lưu lượng mạng.
- Sử dụng lệnh YUM để cài đặt trên máy chủ server1:

[root@server1]# yum install iptraf-ng nc

- Cài đặt 2 công cụ này trên máy chủ khác là server2:

[root@server2]# yum install iptraf-ng nc

- **Tạo file nhật ký log chứa thông tin lưu lượng mạng:**

- Trên máy chủ server 1, hãy chạy iptraf-ng và vào phần Cấu hình... Trong menu, đây là menu chúng ta điều khiển bằng bàn phím. Vào mục "IP traffic monitor".

- Trong menu tiếp theo, chọn eth0 Đặt đường dẫn file nhật ký log thành: /root/traffic_log.txt.
- Sau đó nhấn Enter vào màn hình lưu lượng IP. . và quá trình ghi nhật ký log file sẽ bắt đầu.

2. Lắng nghe thông tin lưu lượng mạng giữa 2 máy servers

- Hãy mở terminal thứ hai vào server1 để thực hiện việc bắt đầu để server 1 nghe netcat trên cổng 2525 với lệnh sau:

```
[root@server1]# nc -l 2525
```

- Gửi một số lưu lượng truy cập mạng từ server 2 vào server1
- Quay lại terminal của server2 mà chúng ta đã mở, gửi lưu lượng netcat đến server1 bằng lệnh nc (trong đó x.x.x.x là IP nội bộ của server1):

```
[root@server2]# nc x.x.x.x 2525
```

- Nó sẽ cho phép chúng ta nhập 1 số thông điệp, và chúng ta có thể nhập một số thông điệp tại dấu nhắc và nhấn Enter.
- Khi chúng ta thực hiện lệnh trên, nó sẽ hiển thị lại trong cửa sổ mà chúng ta đang nghe trên server1.

3. Một loạt các thông điệp được gửi từ server2 sẽ giống như ở dưới:

```
[root@server2]# nc x.x.x.x 2525
test
test
test
This is a test
```

- Trên server1, các thông điệp từ server2 gửi đến sẽ hiển thị lên Terminal như dưới:

```
[root@server1]# nc -l 2525
test
test
test
This is a test
```

- Như vậy là đủ thông tin lưu lượng truy cập được gửi từ server2 đến server1. Trên server2, nhấn Ctrl + C để tắt lệnh nc mà chúng ta đang chạy và quay lại terminal trên server 1 mà chúng ta đang chạy công cụ iptraf-ng. Nhấn x để dừng theo dõi và thoát

ra, sau đó chọn Exit Thoát khỏi menu chính.Kiểm tra File log chứa thông tin lưu lượng mạng

- Trên server1, nếu chúng ta chạy ls ls /var/log/iptraf-ng/, chúng ta sẽ thấy iptraffic.txt được liệt kê . Đọc nó để xem nó có nắm bắt được những gì chúng ta cần không:

  [root@server1]# less /var/log/iptraf-ng/iptraffic.txt

- Chúng ta sẽ thấy nội dung của file này hiển thị lưu lượng truy cập từ server2 đến server1 trên cổng 2525.

XVIII. Thực hành kiểm tra phân giải tên miền DNS

1. **Đánh giá cấu hình DNS hiện tại trong server**

- Kiểm tra xem hệ thống có thể phân giải tên máy chủ thành địa chỉ IP:

  host www.google.com

- **Lưu ý**: Lệnh này sẽ timeout. Hoặc sử dụng 1 dns nội bộ để phân giải tên miền
- Kiểm tra các IP của máy chủ DNS trong file /etc/resolv.conf:

  cat /etc/resolv.conf

- **Lưu ý:** Chúng ta có thể thấy ip của máy chu dns server, vd là: 192.168.x.x
- Chúng ta xóa toàn bộ địa chỉ dns server cũ trong file /etc/resolv.conf để thiết đặt dns server mới, sử dụng lệnh sau:

  sed -i '/nameserver/d' /etc/resolv.conf

- Kiểm tra lại xem hệ thống có thể phân giải tên máy chủ thành địa chỉ IP:

  host www.google.com

- **Lưu ý**: Lệnh này sẽ timeout. Do không có dns server để phân giải tên miền trong hệ điều hành
- Xem lại các kết nối mạng:

  nmcli con show

- Tên card mạng mặc định của chúng ta có thể là ens160. Xem lại các thiết lập địa chỉ IP DNS của chúng ta:

  nmcli -f ipv4.dns con show "ens160"

- Chúng ta thế card ens160 chưa được thiết đặt địa chỉ của 1 máy chủ dns .

2. **Cấu hình để hệ điều hành sử dụng dns của google:**

   sudo nmcli con mod "ens160" ipv4.dns "8.8.8.8"

- Sử dụng lệnh sau để xóa bỏ cài đặt DNS server nội bộ

  sudo nmcli con mod "ens160" ipv4.ignore-auto-dns yes

- Xác nhận các thiết lập bằng lệnh nmcli và sau đó kiểm tra file /etc/resolv.conf:

  nmcli -f ipv4.dns con show "ens160"

  cat /etc/resolv.conf

- Sử dụng lệnh sau để áp dụng các thay đổi đã cấu hình:

  sudo nmcli con up "ens160"

- Xác minh lại các thiết lập của chúng ta:

  cat /etc/resolv.conf

- Bây giờ, thử phân giải tên máy chủ thành địa chỉ IP:

  host www.google.com

- Hệ thống Linux của chúng ta nên có thể phân giải địa chỉ IP cho tên miền này.

XIX. Thực hành tạo sudo Users mới

1. Tạo hai người dùng mới

- Tạo một người dùng webuser trên hệ thống Linux:

  sudo useradd -m webuser

- Tạo một người dùng webadmin , và gán user này vào nhóm wheel:

  sudo useradd -G wheel -m webadmin

- Đặt mật khẩu cho cả hai tài khoản là Git12345!:

  sudo passwd webuser

  sudo passwd webadmin

- Xác nhận file /etc/sudoers Và Kiểm Tra quyền Truy Cập
- Xác nhận là file /etc/sudoers sẽ cho phép nhóm wheel có quyền chạy tất cả các lệnh với lệnh sudo:

  sudo visudo

- **Lưu ý**: không để một chú thích (#) comment trên dòng này của file:

  %wheel ALL=(ALL) ALL

- Chuyển sang tài khoản webadmin và sử dụng dấu gạch ngang (-) để sử dụng một login shell:

  sudo su - webadmin

- Thử test bằng cách Chạy lệnh đọc file /etc/shadow trong terminal:

  cat /etc/shadow

- Chạy lại lệnh với lệnh sudo:

  sudo cat /etc/shadow

- Sau khi xác minh là user webadmin có thể đọc được file /etc/shadow, đăng xuất ra khỏi tài khoản này:

  exit

2. Thiết lập user có quyền Quản trị dịch vụ Web

- Tạo một file sudoers mới trong thư mục /etc/sudoers.d :

  sudo visudo -f /etc/sudoers.d/web_admin

- Thêm 1 nội dung vào file, để cấp quyền cho người dùng quản trị dịch vụ web như sau:

Cmnd_Alias WEB = /bin/systemctl restart httpd.service, /bin/systemctl reload httpd.service

- Thêm một dòng khác vào file cho user webuser có thể sử dụng lệnh sudo kết hợp với bất kỳ lệnh nào được liệt kê trong nhóm alias WEB:

  webuser ALL=WEB

- Lưu và đóng file bằng lệnh :wq.

3. Tiếp theo, đăng nhập vào HĐH bằng tài khoản webuser:

sudo su - webuser

- Khởi động lại dịch vụ web:

  sudo systemctl restart httpd.service

- Thử đọc file sudoers mới tạo là web_admin:

  sudo cat /etc/sudoers.d/web_admin

- Do lệnh cat không được liệt kê trong dòng lệnh alias của file web_admin , vì vậy user webuser không thể sử dụng sudo để đọc file này.

XX. Thực hành cấu hình bảo mật truy cập SSH cho một Linux Server

Mở ứng dụng terminal của bạn và truy cập vào Linux Centos của bạn.

- **Lưu ý**: Để thực hiện bài thực hành này cần phải disable SELinux.

  ssh root@<YOUR_IP>

- Nhập mật khẩu của bạn tại dấu nhắc.

1. Cấu hình sshd để sử dụng Sockets.

- Xác minh là sshd.socket unit chưa được bật lên enable trong hệ điều hành.
  systemctl status sshd.socket
- Setup một at job để dừng dịch vụ sshd.service và khởi động sshd.socket.
  sudo at now + 3 minutes
- Nhập mật khẩu của bạn tại dấu nhắc. Thêm nội dung sau:
  at> systemctl stop sshd.service
  at> systemctl start sshd.socket
- Nhấn Ctrl + D để kết thúc việc cấu hình at.

- Xác nhận là sshd.socket unit đã được kích hoạt và đang chạy.
  systemctl status sshd.socket
- Cho phép enable sshd.socket được bật lên vĩnh viễn và tắt disable dịch vụ sshd.service.
  sudo systemctl enable sshd.socket
  sudo systemctl disable sshd.service

2. Cài đặt và cấu hinh sử dụng TCP wrappers trên Centos 8 :

- Cài đặt gói epel-release từ kho lưu trữ EPEL (Extra Packages for Enterprise Linux). Để có thể truy cập vào các gói phụ trợ và bổ sung từ kho lưu trữ EPEL trên CentOS/RHEL:

  dnf install epel-release

- Cài đặt gói tcp_wrappers, một công cụ cho phép bạn kiểm soát quyền truy cập các dịch vụ mạng:

  dnf install tcp_wrappers

- Copy file dịch vụ sshd@.service vào /etc/systemd/system/ :

cp /usr/lib/systemd/system/sshd@.service /etc/systemd/system/

3. **Tiếp theo chúng ta soạn thảo file "sshd@.service" để cho phép TCP wrappers quản lý dịch vụ ssh server:**

vi /etc/systemd/system/sshd@.service

- Thay đổi dòng cấu hình" ExecStart=-/usr/sbin/sshd -i $OPTIONS $CRYPTO_POLICY " thành như sau:

ExecStart=@-/usr/sbin/tcpd /usr/sbin/sshd -i $OPTIONS $CRYPTO_POLICY

- Thiết lập TCP Wrappers để chỉ cho phép truy cập từ xa vào SSH.Chỉnh sửa file /etc/hosts.allow:

sudo vim /etc/hosts.allow

- **Thêm dòng sau vào file:**

sshd2 sshd: ALL

- Chỉnh sửa file /etc/hosts.deny.

sudo vim /etc/hosts.deny

- Thêm dòng sau vào file:

ALL: ALL

- Thoát khỏi phiên SSH.

exit

- Kết nối lại với phiên secure shell.

ssh root@your_IP

- Nhập mật khẩu của bạn tại dấu nhắc.

- Như vậy Bài thực hành này cung cấp hướng dẫn về cách sử dụng TCP Wrappers và systemd socket để cấu hình bảo mật cho SSH trên CentOS. TCP Wrappers là một công cụ được sử dụng để giới hạn truy cập vào các dịch vụ mạng bằng cách cho phép hoặc từ chối các kết nối dựa trên địa chỉ IP hoặc hostname. Systemd socket là một cơ chế kích hoạt các dịch vụ mạng khi được yêu cầu, giúp tiết kiệm tài nguyên hệ điều hành, Vậy các bạn hãy thực hành một vài lần theo hướng dẫn để chúng ta hiểu được kiến thức tốt hơn.

XXI. Thực hành mã hóa một File sử dụng GPG

Lưu ý: Để thực hiện bài lab, các bạn cần phải tạo 02 users, ví dụ: user01 và user02

1. Tạo một GPG key cho user01. Sử dụng lệnh:

gpg --full-generate-key

- Chấp nhận các mục mặc định tại mỗi dấu nhắc, xong nhấn enter:

Please select what kind of key you want:

(1) RSA and RSA (default)

(2) DSA and Elgamal

(3) DSA (sign only)

(4) RSA (sign only)

Your selection?

RSA keys may be between 1024 and 4096 bits long.

What keysize do you want? (2048)

Requested keysize is 2048 bits

Please specify how long the key should be valid.

0 = key does not expire

<n> = key expires in n days

<n>w = key expires in n weeks

<n>m = key expires in n months

<n>y = key expires in n years

Key is valid for? (0)

Key does not expire at all

For Real name, enter cloud_user, and use cloud_user@localhost for the Email address. We can leave the comment field blank by just pressing Enter, and press o at the end for OK:

Real name: user01

Email address: user01@localhost

Comment:

You selected this USER-ID:

"user01 <user01@localhost>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O

- Chúng ta sử dụng mật khẩu: 12345678 cho mật khẩu của passphrase, các bạn cần xác nhận mật khẩu thêm 1 lần.Bây giờ thì key đã được tạo ra, và bạn sẽ gửi đến user02, do đó bạn cần export nó để user02 có thể sử dụng để giải mã các files . Chúng ta thực hiện export key bằng lệnh sau:

gpg -a -o user01.key --export <KEY_ID>

- Trong lệnh đó, hãy sử dụng ID tham chiếu public key từ đầu ra của quá trình tạo khóa. Nó sẽ là một chuỗi ngẫu nhiên và dòng chứa nó (trong đầu ra tạo khóa)

sudo chmod -R 777 $(tty)

- **Trông như thế này:**

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /home/user01/.gnupg/trustdb.gpg: trustdb created
gpg: key 76B8DAF3 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
pub 2048R/76B8DAF3 2022-11-04
Key fingerprint = 6C4A 9741 0522 DEF9 9487 0383 EAFE 0E6E 76B8 DAF3
uid user01 <user01@localhost>
sub 2048R/C0171E1D 2022-11-04

[user01@server01 ~]$ gpg -a -o user01.key --export 76B8DAF3

- **Bây giờ chúng ta sẽ sử dụng lệnh mail để gửi một email với file public key đính kèm**
  "user01.key " tới user02:

[user01@server01 ~]$ mail -s "get your key" -a user01.key user02@localhost

Please keep this! I'll give you the passphrase then.

.

EOT

2. Cấu hình GPG cho user02

- Sử dụng ssh để login vào user02:

  ssh user02@localhost

- Giống như chúng ta đã thực hiện với tài khoản user01, chúng ta sẽ tạo GPG key cho user02, Enter để chấp nhận các giá trị mặc định cho mỗi lời nhắc.
- **Sự khác biệt duy nhất sẽ có tên của user02 và địa chỉ Email của user02@localhost:**

gpg --gen-key

[user02@server01 ~]$ gpg --full-generate-key

gpg (GnuPG) 2.0.22; Copyright (C) 2013 Free Software Foundation, Inc.

This is free software: you are free to change and redistribute it.

There is NO WARRANTY, to the extent permitted by law.

gpg: directory `/home/user02/.gnupg' created

gpg: new configuration file `/home/user02/.gnupg/gpg.conf' created

gpg: WARNING: options in `/home/user02/.gnupg/gpg.conf' are not yet active during this run

gpg: keyring `/home/user02/.gnupg/secring.gpg' created

gpg: keyring `/home/user02/.gnupg/pubring.gpg' created

Please select what kind of key you want:

(1) RSA and RSA (default)

(2) DSA and Elgamal

(3) DSA (sign only)

(4) RSA (sign only)

Your selection?

RSA keys may be between 1024 and 4096 bits long.

What keysize do you want? (2048)

Requested keysize is 2048 bits

Please specify how long the key should be valid.

0 = key does not expire

<n> = key expires in n days

<n>w = key expires in n weeks

<n>m = key expires in n months

<n>y = key expires in n years

Key is valid for? (0)

Key does not expire at all

Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: user02

Email address: user02@localhost

Comment:

You selected this USER-ID:

"user02 <user02@localhost>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o

- Sử dụng cùng mật khẩu 12345678 cho passphrase.
- Khi chúng ta đã tạo key cho user02, chúng ta mở ứng dụng email mutt và lưu public key được gửi từ tài khoản user01:

mutt

- Di chuyển Mũi tên lên và xuống để xem mail của user01, sau đó nhấn Enter. Nhấn v để xem file đính kèm và nhấn s để lưu file vào thư mục home của user02. Cuối cùng, nhấn q để thoát khỏi Mutt. Bây giờ, để nhập import public key từ user01 vào key ring của user02, hãy chạy lệnh sau:

  gpg --import user01.key

- Chúng ta có thể chạy lệnh dưới để xem nội dung key ring của user02:

  gpg --list-keys

- Hãy đăng xuất khỏi tài khoản của user02: exit

3. Tạo, mã hóa 1 file văn bản và gửi file này đến user02

- Khi chúng ta ký điện tử vào một file, chúng ta đang sử dụng khóa GPG riêng của chúng ta để đảm bảo rằng file này là của chúng ta. Người dùng nhận được file sẽ sử dụng bản sao public key mà chúng ta đã gửi, để xác nhận là chúng ta đã ký file. Hãy tạo thử một file mã hóa:

  echo "Please confirm this file" > note.txt

- Bây giờ, chúng ta sẽ sử dụng private key của user01 để ký file:

  gpg --clearsign note.txt

- Hãy nhớ là chúng ta cần sử dụng passphrase mà chúng ta đã tạo trước đó (12345678). Bây giờ sẽ có một file note.txt.asc trong thư mục chính của user01. Chúng ta có thể chạy lệnh ls để xem file này. Khi chúng ta đã tạo xong file, hãy gửi nó qua email đến user02@localhost:

  mail -s "Check this one" -a note.txt.asc user02@localhost

  Please confirm this secret file for me?

4. Xác nhận chữ ký của file mã hóa được gửi qua email:

- Đăng nhập lại vào localhost với user02:

  ssh user02@localhost

- Sử dụng ứng dụng email mutt và giống như trước đây, xem và lưu file đính kèm của email mới. Bây giờ, hãy xác nhận file note.txt.asc đã được gửi qua email:

  gpg --verify note.txt.asc

- Chúng ta sẽ nhận được cảnh báo về chữ ký không được bên thứ ba xác nhận và việc này không có vấn đề gì. Việc quan trọng là dòng sau từ đầu ra:

  gpg: Good signature from user01 <user01@localhost>"

- Đây là những gì một file xác minh được hiển thị. Tiếp theo, mã hóa một bản sao của file /etc/fstab như sau:

```
cp /etc/fstab ~

gpg -a -r user01 -e ~/fstab
```

- Bạn sẽ thấy một cảnh báo chung được hiển thị về việc key có thể không thuộc về người được nêu tên. Chúng ta đã biết rằng key này là từ user01, vì vậy chỉ cần nhấn y tại dấu nhắc. Xác nhận là có một file có tên fstab.asc trong thư mục chính của user02 (bằng cách chạy lệnh ls). Tạo một email mới tới user01 và đính kèm file này:

```
mail -s "Mail is ok" -a fstab.asc user01@localhost

Can you decrypt this?

.
```

- Đăng xuất khỏi tài khoản của user02:

```
exit
```

5. Giải mã file đính kèm:

- Bây giờ, trong tài khoản user01, hãy mở ứng dụng email mutt và lưu file đính kèm fstab.asc từ email mới. Giải mã file fstab.asc đã lưu bằng lệnh gpg và nhập passphrase cho khóa của user01 khi được nhắc:

```
gpg fstab.asc
```

- Và, hãy xác nhận là chúng ta có thể đọc nội dung của file được giải mã:

```
cat fstab
```

XXII. Thực hành Tạo và trao đổi SSH Keys để truy cập từ xa vào Server an toàn bảo mật

- Tạo Khóa SSH trên Máy chủ 1 và Máy chủ 2

- Tạo Key trên Server 1

1. **Trong terminal của bạn, hãy đăng nhập vào Máy chủ 1.**

```
ssh user01@your_IP server1;
```

- Liệt kê nội dung của thư mục hiện tại.

```
ls -la
```

- Thay đổi thư mục .ssh.

```
cd .ssh
```

- Liệt kê nội dung của thư mục .ssh.

ls -la

1.1.Tạo key cho Máy chủ 1.

ssh-keygen

- Nhấn Enter ở ba lời nhắc tiếp theo. Liệt kê lại nội dung của thư mục .ssh.

  ls -la

- Liệt kê nội dung của file id_rsa.pub.

  cat id_rsa.pub

- copy kết quả của lệnh này vào clipboard.

1.2.Tạo Key trên Server 2

- Trong terminal của máy chủ 1, hãy đăng nhập vào máy chủ 2.

  ssh user02@yourIP server2;

- Thay đổi thư mục .ssh.

  cd .ssh/

- Liệt kê nội dung của thư mục .ssh.

  ls -la

- Cài đặt trình soạn thảo văn bản nano.

  Sudo yum install nano

- Nhập mật khẩu của bạn tại dấu nhắc. Mở file authorized_keys trong nano.

  nano authorized_keys

- Thêm key mà chúng ta vừa tạo vào file. Nhấn Ctrl + X. Nhấn Y rồi Enter để lưu các thay đổi.
  Trao đổi key SSH giữa các máy chủ
- Trong màn hình terminal của Máy chủ 2 của bạn, hãy tạo một khóa mới.

  ssh-keygen

- Nhấn Enter cho ba lời nhắc tiếp theo. Liệt kê nội dung của thư mục hiện hành.

  ls -la

- Liệt kê nội dung của file id_rsa.pub.

  cat id_rsa.pub

- Copy kết quả của lệnh này vào clipboard. Nhập exit để đăng xuất khỏi Máy chủ 2(Có nghĩa là quay lại terminal của máy chủ 1)

- Cài đặt nano.

  Sudo yum install nano

- Nhập y để tiếp tục. Liệt kê nội dung của thư mục hiện hành.

  ls -la

- Mở file authorized_keys trong nano.

  nano authorized_keys

- Thêm khóa chúng ta vừa tạo vào file. Nhấn Ctrl + X. Nhấn Y rồi Enter để lưu các thay đổi.

2. Kiểm tra cấu hình

- Kiểm tra đăng nhập vào Máy chủ 2 từ Máy chủ 1 mà không cần mật khẩu.

  ssh user02@IP của máy chủ 2

- Kiểm tra đăng nhập vào Máy chủ 1 từ Máy chủ 2 mà không cần mật khẩu.

  ssh user01@IP của máy chủ 1

XXIII. Thực hành quản lý Docker container

- Chúng ta cần phải đăng nhập với quyền root , vì vậy hãy chạy "sudo -i" ngay khi đăng nhập.

1. Chạy image hello-world và busybox

- Đầu tiên chúng ta kiểm tra xem Docker đã được cài đặt và dịch vụ docker.service đã được kích hoạt chưa:

  docker run hello-world

  docker run --name hi hello-world

- Nếu chúng ta không chắc image đang ở đâu, hoặc chính xác là tên gọi là gì, chúng ta có thể tìm kiếm:

  docker search busybox

- Chúng ta muốn image được đưa lên đầu danh sách và có thể tải nó về server bằng lệnh:

  docker pull docker.io/busybox

- Bây giờ chúng ta có thể chạy image này. Chúng ta sẽ đặt tên nó là "busy" và thiết lập thông số cho phép chúng ta tương tác với container trong terminal:

  docker run --name busy -it busybox /bin/sh

- Nhập "exit" để thoát.

2. **Hiển thị container đang chạy**
   - Chúng ta có thể xem các containers đang chạy sử dụng lệnh "docker ps". Nếu chúng ta muốn xem tất cả các container của chúng ta, ngày cả là các containers này có đang chạy hay không, thì chúng ta chạy lệnh sau:

     docker ps -a

   - Chúng ta sẽ thấy một số container có trong hệ thốnng.Vậy Hãy xóa bỏ container "hi":

     docker rm hi

   - Chạy lại "docker ps -a" để kiểm tra nó đã bị xóa chưa.
   - Hiển thị các images
   - Để hiển thị images trong kho lưu trữ cục bộ trong server, chúng ta có thể chạy lệnh này:

     docker images

3. **Nếu chúng ta không cần image nữa, chúng ta có thể loại bỏ nó, thực hiện các bước sau:**
   - Dừng tất cả các container đang sử dụng image "hello-world" bằng cách chạy lệnh:

     docker stop $(docker ps -a -q --filter ancestor=hello-world)

   - Xóa tất cả các container đang sử dụng image "hello-world" bằng cách chạy lệnh:

     docker rm $(docker ps -a -q --filter ancestor=hello-world)

   - Xóa image "hello-world" bằng lệnh:

     docker rmi -f hello-world

   - Trong trường hợp này, chúng ta đã xóa image "hello-world". Kiểm tra xem nó đã bị xóa hay chưa, bằng cách chạy lệnh sau:

     docker images

   - Nếu image này đã ko còn được hiển thị, kết quả sẽ không còn chứa image "hello-world" nữa.

4. **Tạo một Container apache2 dựa trên Image httpd:2.4 và ánh xạ localhost: 8080 đến Cổng 80 của Container**
   - Chúng ta sẽ sử dụng lệnh "docker run" để tạo một container có tên là "apache2", dựa trên image "httpd:2.4". Chúng ta cũng sẽ ánh xạ cổng 8080 trên localhost thành cổng 80 trên container bằng cách sử dụng "-p 8080:80".

     docker run --name apache2 -p 8080:80 httpd:2.4

- Để thoát khỏi container đang chạy , nhấn Ctrl + c. Sau đó để kiểm tra nó có tồn tại hay không, chạy:

  docker ps -a

- Để khởi động container, chúng ta có thể sử dụng lệnh:

  docker start apache2

- Bây giờ chúng ta sẽ có thể thấy container này với lệnh "docker ps" (không có tùy chọn "-a").

5. **Thêm thông tin**
   - Nếu chúng ta cần xử lý sự cố hoặc muốn biết có vấn đề gì đang xảy ra với container, chúng ta có thể xem các nhật ký logs và thống kê của container bằng hai lệnh sau:

     docker logs apache2

     docker stats apache2

   - Chúng ta phải nhấn Ctrl + c để thoát khỏi lệnh thứ hai.
   - Container này dự kiến cung cấp một trang web, vì vậy hãy kiểm tra
     lynx -dump http://localhost:8080
6. **Trong Container apache2, chúng ta sẽ cập nhật file index.html mặc định, sau đó xác nhận và lưu các thay đổi**

   - Để thực thi một lệnh bên trong 1 container đang chạy, chúng ta phải chạy lệnh sau đây :

     docker exec -it apache2 bash

   - Trong container, chúng ta sẽ được làm việc dưới quyền root và đang ở trong thư mục /usr/local/apache2. Chạy lệnh "ls" để xem có gì ở trong thư mục này, sau đó chạy "cd htdocs" để vào thư mục trang web index.html.
   - Bây giờ chúng ta muốn thay đổi trang web để tất cả người dùng có thể truy cập xem container apache2 :

     echo 'apache2 container' > index.html

   - Sau đó, chúng ta có thể chạy lệnh exit để thoát khỏi container, và kiểm tra xem thay đổi của chúng ta có tác dụng hay không bằng cách chạy lệnh sau:

     lynx -dump http://localhost:8080

   - Nếu chúng ta khởi động lại container ở bước này, thay đổi của chúng ta sẽ bị mất, do đó chúng ta sẽ commit nó.
   - Vậy chúng ta sẽ commit các thay đổi và khởi động lại container apache2 để xác nhận là các thay đổi đã được áp dụng. Bằng cách sử dụng lệnh:

```
docker commit -m 'Updated index.html' apache2
```

- Chúng ta đã commit các thay đổi cho container. Sau đó, chúng ta sẽ dừng container bằng lệnh:

```
docker stop apache2
```

- Và kiểm tra xem container đã stop bằng lệnh:

```
docker ps -a
```

- Tiếp theo, chúng ta khởi động lại container bằng lệnh:

```
docker start apache2
```

- Và cuối cùng, chúng ta sẽ kiểm tra website mà container đang cung cấp bằng lệnh:

```
lynx -dump http://localhost:8080
```