Práctica 1

Jesús Rodríguez Heras
Arantzazu Otal Alberro

19 de marzo de 2019

1. Instalación de máquinas virtuales mediante Vagrant

1. **Crear una máquina virtual usando Vagrant.**
   Para descargar la máquina virtual usamos `vagrant init hashicorp/precise64` y para lanzarla `vagran up`.

2. **Actualizar el listado de los paquetes.**
   Para actualizar el listado de paquetes usamos `sudo apt-get update`.

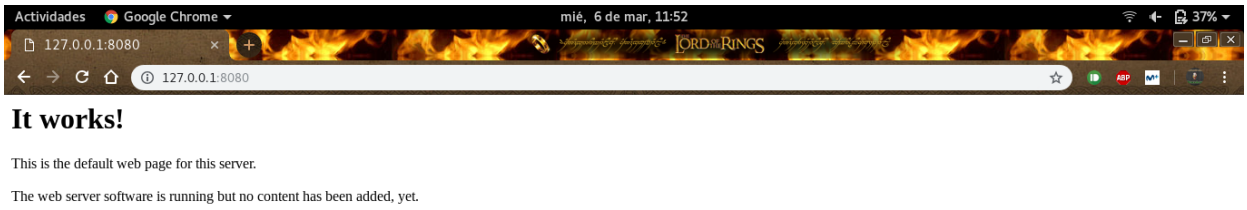
3. **Instalar el Apache:**
   Para la instalación de apache `sudo apt-get install -y apache2`.

4. **Redirigir el puerto 80 para accederse localmente.**
   Para redirigir el puerto 80[1] lo indicamos en el archivo `Vagrantfile` de la siguiente forma:

```
1  Vagrant.configure("2") do |config|
2  config.vm.box = "hashicorp/precise64"
3  config.vm.network :forwarded_port, guest: 80, host: 80
4  end
```

   Para entrar en la máquina virtual usamos `vagrant ssh`.

   Para comprobar que funciona correctamente, buscamos la dirección `127.0.0.1:80`[2] en nuestro navegador, y nos redirigirá a la página de Apache de la máquina virtual.



[1] Cuidado con el puerto 80 porque nos puede salir un error en el que nos dice que debemos usar un puerto superior al 1024. Por lo tanto, usaremos el 8080 en el archivo `Vagrantfile`, quedando `... guest:80, host: 8080` en la línea 3 del código anteriormente mostrado.

[2] `127.0.0.1:8080` en caso de que no nos haya dejado usar el puerto 80.

2. Comandos básicos

1. **Mostrar todos los archivo txt del sistema.**
   `find / -type f -name "*.txt"`

2. **Modificar el ejercicio anterior para que no se muestren los errores de acceso.**
   `find / -type f -name "*.txt"|& grep -v "Permission denied"`

3. **Mostrar el número de archivos txt del sistema.**
   `find / -type f -name "*.txt"|& grep -v "Permission denied"| wc -l`

4. **Mostrar cuantos usuarios no pueden iniciar sesión**
   `cat /etc/passwd | grep nologin | wc -l`

5. **Mostrar el tipo de inicio de sesión de los usuarios que están dados de alta en el sistema, así como el número de cada uno de estos usuarios. La información se tiene que mostrar de forma ordenada por el número de usuarios.**
   `cat /etc/passwd | cut -d : -f 7 | sort | uniq -c | sort -n`

3. Procesos

1. **Encuentra usando find y perm los programas con el setuid activado. ¿Cuáles son?**
   `find / -perm -4000 2>/dev/null`

2. **Identificar los 3 procesos que se ejecutan con permisos de root que requieren más memoria.**
   `ps aux -sort -rss | grep "root" | head -n 3`

3. **Monitorizar (con watch) la memoria libre. Deberá mostrarse cada 10 segundos la actual memoria libre (únicamente ese valor, sin ningún texto adicional).**
   `watch -n 10 'cat /proc/meminfo | grep "MemFree"| cut -d : -f 2'`

4. Gestión de usuarios

1. **Crear un usuario llamado con nuestras iniciales. El usuario tendrá home, bash y podrá iniciar sesión.** `sudo useradd -m -s /bin/bash JAROHA`

2. **Crear un usuario llamado *webuser* sin bash, ni home y que no pueda iniciar sesión.**
   `sudo useradd -M -s /bin/false webuser`

3. **Crear un usuario llamado antonio que se pueda identificar como webuser (usando sudo) pero no como root.** `sudo useradd -m -s /bin/bash antonio`
   Luego, debemos cambiar en el archivo /etc/sudoers añadiendo al usuario antonio de la siguiente forma:
   `antonio ALL=(webuser) ALL`

5. /proc

1. **Identificar tres archivos y explicar su contenido.**

   - `meminfo:` Nos muestra información de la memoria. Por ejemplo: memoria libre, memoria ocupada, etc.
   - `version:` Nos muestra información sobre la versión del sistema.
   - `partitions:` Nos muestra información a cerca de las particiones de nuestro sistema.

2. **Identificar al menos dos archivos que permitan escritura.**
   Con el comando `ls -l` podemos fijarnos en los permisos que tiene cada fichero. Dos ejemplos de archivos que permiten escritura son: `timer_stats` y `mtrr`

6. Bash scripting

1. **Realizar un programa que permita cambiar el nombre de los archivos de un directorio dado como argumento de entrada. El nuevo nombre será un número secuencial, manteniendo la extensión.**

```
#!/bin/bash
archivos=$(ls "$1")
n=0
for i in ${archivos[@]}
do
    ext=$(echo $i | cut -d . -f 2)
    nombre="${n}.$ext"
    mv "${1}/$i" "${1}/$nombre"
    let n=n+1
done
```

2. **Se pide realizar un programa que realice una copia de los directorios que cuelguen de /importante/ en media/backup, siguiendo las directivas:**

   - **Cada directorio que cuelga de /importante/ deberá de guardarse en un fichero `.tgz`.**
   - **Cada fichero de backup deberá de tener el nombre del directorio seguido de la fecha. La fecha estará en formato año, mes, día. Por ejemplo, para copiar el directorio `docencia` deberá tener el nombre `docencia_20140404.tgz`.**

```
#!/bin/bash

cd ~/importante/

fecha=$(date +%Y%m%d)

for i in */; do
    tar -cvzf /media/backup/"${i%/}_$fecha.tgz" "$i";
done
```

3. **Proponer una modificación al programa anterior, de tal modo de que parra cada directorio se guarden sólo los 5 últimos ficheros. Es decir, al copiar el sexto fichero asociado a un mismo directorio se borrará la copia más a ntigua.**

```
#!/bin/bash

cd ~/importante/

fecha=$(date +%Y%m%d)

for i in */; do
    tar -cvzf /media/backup/"${i%/}_$fecha.tgz" "$i";
done

n_ficheros=$(ls /media/backup | wc -l)

if [ $n_ficheros -eq 6 ]; then
    rm -r $(ls -tr /media/backup | head -1)
fi
```

4. **Utilizar el crontab para ejecutar el programa de backup cada día a las 5 de la mañana.** Para ejecutar el crontab usamos `crontab -e`.

   A continuación, seleccionamos el editor que más nos guste, en nuestro caso, `nano` (opción 2).

   Por último escribimos lo siguiente en el archivo de configuración de `crontab`:

   `0 5 * * * /home/vagrant/e3.sh`.

   Al guardar, ya se queda activado y listo para ejecutarse a las 5 de la mañana.