

Práctica 2

El objetivo de esta práctica es mejorar la comunicación entre las dos máquinas virtuales, copiando archivos por ssh y clonando contenidos, además de establecer tareas con cron.

En la primera parte de la práctica crearemos un archivo tar con ficheros o directorios en un equipo remoto, para ello usaremos el comando:

1. `tar czf - directorio | ssh usuario@equiporemoto 'cat > ~/archivo.tgz'`
2. `tar -czvf directorio archivo.tgz scp archivo.tgz usuario@equiporemoto:~/archivo.tgz`

o directamente:

1. `scp -r directorio usuario@equiporemoto:/directorio`

```
root@UbuntuServer:/home/laguilarg99# mkdir P2
root@UbuntuServer:/home/laguilarg99# ls
P2
root@UbuntuServer:/home/laguilarg99# cd P2
root@UbuntuServer:/home/laguilarg99/P2# mkdir SSH
root@UbuntuServer:/home/laguilarg99/P2# ls
SSH
root@UbuntuServer:/home/laguilarg99/P2# cd SSH/
root@UbuntuServer:/home/laguilarg99/P2/SSH# touch Prueba{1,2,3}
root@UbuntuServer:/home/laguilarg99/P2/SSH# ls
Prueba1 Prueba2 Prueba3
```

En M1 tenemos la siguiente estructura de archivos:

`/home/laguilarg99/P2/SSH/Prueba{1,2,3}`

Enviaremos el directorio SSH a la máquina M2 ejecutando cualquiera de los comandos citados anteriormente, es importante dar permisos de escritura al grupo otros en la carpeta donde deseamos crear el archivo tar.

M1:

```
root@UbuntuServer:/home/laguilarg99/P2# tar czf - SSH | ssh laguilarg99@192.168.56.102 'cat > ~/P2/SSH.
tgz'
laguilarg99@192.168.56.102's password:
```

M2:

```
root@UbuntuServer:/home/laguilarg99# ls P2
SSH.tgz
```

En la segunda parte de la práctica nos centraremos en el proceso de instalación de **Rsync** y como usarlo para copiar grandes cantidades de datos. Instalamos dicho programa en ambas máquinas:

```
root@UbuntuServer:/home/laguilarg99# sudo apt-get install rsync
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se actualizarán los siguientes paquetes:
  rsync
1 actualizados, 0 nuevos se instalarán, 0 para eliminar y 196 no actualizados.
Se necesita descargar 329 kB de archivos.
Se utilizarán 0 B de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu xenial-updates/main amd64 rsync amd64 3.1.1-3ubuntu1.3 [329 kB]
Descargados 329 kB en 1s (277 kB/s)
(Leyendo la base de datos ... 61962 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../rsync_3.1.1-3ubuntu1.3_amd64.deb ...
Desempaquetando rsync (3.1.1-3ubuntu1.3) sobre (3.1.1-3ubuntu1.2) ...
Procesando disparadores para systemd (229-4ubuntu21.4) ...
Procesando disparadores para ureadahead (0.100.0-19) ...
Procesando disparadores para man-db (2.7.5-1) ...
Configurando rsync (3.1.1-3ubuntu1.3) ...
```

Ahora clonamos la carpeta /var/www en la máquina 1, desde la máquina 2 usando el comando:

a. `rsync -avz -e ssh ipmaquina1:/var/www/ /var/www/`

```
laguilarg99@UbuntuServer:~$ rsync -avz -e ssh 192.168.56.101:/var/www/ /var/www/
laguilarg99@192.168.56.101's password:
receiving incremental file list
./
html/
html/index.html

sent 156 bytes received 329 bytes 138.57 bytes/sec
total size is 11,324 speedup is 23.35
```

Habiendo ejecutado antes en ambas máquinas el comando:

a. `sudo chown usuario:usuario -R /var/www`

```
laguilarg99@UbuntuServer:~$ sudo chown laguilarg99:laguilarg99 -R /var/www
```

En la tercera parte de la práctica aprenderemos a acceder por ssh a una máquina y a otra sin necesidad de introducir la contraseña. Para esto se usa la autenticación con claves pública-privada.

Tras ejecutar:

- a. `ssh-keygen -b 4096 -t rsa`
- b. `ssh-copy-id 192.168.56.101`

Podremos conectarnos a la máquina 1 sin necesidad de claves:

```
laguilarg99@UbuntuServer:~$ ssh 192.168.56.101
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-131-generic x86_64)

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

Pueden actualizarse 201 paquetes.
144 actualizaciones son de seguridad.

New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Mar 16 14:44:34 2020 from 192.168.56.101
```

En la última parte de la práctica simplemente, programaremos tareas con crontab, tal y como especifica la práctica estableceremos una tarea en cron que se ejecute cada hora para actualizar el directorio /var/www entre máquinas.

Para que se pueda ejecutar el comando de rsync correctamente sin que pida la contraseña cada vez que dicho comando tenga lugar debemos de crear una conexión ssh sin contraseñas entre las máquinas usando los comandos descritos en la parte anterior.

El script tiene la siguiente forma:

```
laguilarg99@UbuntuServer:~/P2$ cat copiaWEB.sh
#!/bin/bash
rsync -avz -e ssh 192.168.56.102:/var/www/ /var/www/
```

Y el crontab quedaría de la siguiente manera (editado al ejecutar crontab -e):

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
@hourly ~/P2/copiaWEB.sh
```