



SERVIDORES WEB DE ALTAS PRESTACIONES

PRÁCTICA 2: Clonar la información de un sitio web

Autor

Miguel Ángel Pérez Díaz



Escuela Técnica Superior de Ingenierías Informática y
de Telecomunicación

—
Granada, 2020

1. Probar el funcionamiento de la copia de archivos por SSH

En este apartado vamos a intentar la copia de archivos de una máquina a otra mediante el acceso remoto con SSH. Para ello inicialmente vamos a tratar de crear un archivo tar.gz en la máquina M1 y dejarlo en M2, suponiendo que no disponemos de espacio en disco local de M1, podemos usar SSH para crearlo directamente en el equipo destino, en este caso M2.

En nuestro caso suponiendo que el disco está lleno y no habría espacio para crear un directorio de prueba, vamos a movernos hacia el directorio raíz en M1 y vamos a tratar de realizar una copia del directorio *HOME*.

Para ello vamos a redirigir con una pipe la salida del comando **TAR** al **SSH** hacia la máquina remota (M2):

- `tar czf - home | ssh miguel444@192.168.56.102 'cat > ~/archivo.tgz'`

MÁQUINA VIRTUAL M1:

```
miguel444@m1:/$ tar czf - home | ssh miguel444@192.168.56.102 'cat > ~/archivo.tgz'
miguel444@192.168.56.102's password:
miguel444@m1:/$ _
```

MÁQUINA VIRTUAL M2:

```
miguel444@m2:~$ ls
miguel444@m2:~$ ls
archivo.tgz
miguel444@m2:~$
```

Seguidamente trataremos de realizar la misma copia del directorio de M1 hacia M2, pero en este caso vamos a utilizar el comando **SCP** que hace uso de SSH para hacer copias seguras y encriptadas de archivos o directorios. Para ello utilizaremos directamente el comando:

- `scp -r home miguel444@192.168.56.102:/home/miguel444/`

MÁQUINA VIRTUAL M1:

```
miguel444@m1:/$ scp -r home miguel444@192.168.56.102:/home/miguel444/
miguel444@192.168.56.102's password:
.bashrc                                100% 3771      2.6MB/s   00:00
known_hosts                            100% 888       952.5KB/s 00:00
.sudo_as_admin_successful              100% 0         0.0KB/s   00:00
.bash_logout                           100% 220      283.0KB/s 00:00
.bash_history                           100% 50        65.1KB/s  00:00
motd.legal-displayed                  100% 0         0.0KB/s   00:00
.profile                               100% 807       1.1MB/s   00:00
miguel444@m1:/$ _
```

MÁQUINA VIRTUAL M2:

```
miguel444@m2:~$ ls
archivo.tgz  home
miguel444@m2:~$ ls home/
miguel444
miguel444@m2:~$ ls home/miguel444/
./                .bashrc          .ssh/
../               .cache/          .sudo_as_admin_successful
.bash_history     .gnupg/
.bash_logout      .profile
miguel444@m2:~$ ls home/miguel444/
miguel444@m2:~$
```

2. Clonado de una carpeta entre las dos máquinas

Para este apartado vamos a utilizar la herramienta **RSYNC** con la que podremos transferir y sincronizar archivos o directorios de manera eficiente entre una máquina local, un servidor remoto o cualquiera de estos. En nuestro caso trataremos de sincronizar el directorio /var/www entre las máquinas.

Inicialmente se ha intentado instalar la herramienta, pero parece ser que viene incorporada por defecto en esta versión de Linux:

```
miguel444@m2:~$ sudo apt-get install rsync
[sudo] password for miguel444:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
rsync ya está en su versión más reciente (3.1.2-2.1ubuntu1.1).
fijado rsync como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 14 no actualizados.
miguel444@m2:~$ _
```

Por último, antes de probar la herramienta tenemos que hacer que el usuario sea el dueño de la carpeta donde residen los archivos que hay en el espacio web (en ambas máquinas):

- `sudo chown miguel444:miguel444 -R /var/www`

```
miguel444@m1:/$ sudo chown miguel444:miguel444 -R /var/www/  
miguel444@m1:/$ _
```

Una vez preparado todo lo necesario vamos a probar la herramienta, para ello como hemos dicho anteriormente vamos a clonar la carpeta con el contenido del servidor web principal, ejecutaremos en la máquina 2 (secundaria):

- `rsync -avz -e ssh 192.168.56.12:/var/www/ /var/www/`

```
miguel444@m2:~$ rsync -avz -e ssh 192.168.56.12:/var/www/ /var/www/  
miguel444@192.168.56.12's password:  
receiving incremental file list  
./  
html/  
html/ejemplo.html  
html/index.html  
  
sent 169 bytes  received 294 bytes  71.23 bytes/sec  
total size is 10,987  speedup is 23.73  
miguel444@m2:~$ _
```

Tras esto podemos comprobar que el directorio /var/www queda clonado de forma idéntica en ambas máquinas (de M1 se queda copiado en M2). Para ello vamos a observar que los directorios y ficheros copiados mantienen los mismos permisos que en la máquina origen.

```
miguel444@m1:/$ sudo chown miguel444:miguel444 -R /var/www/  
miguel444@m1:/$ ls -la /var/www  
total 12  
drwxr-xr-x  3 miguel444 miguel444 4096 mar 10 17:20 .  
drwxr-xr-x 14 root      root      4096 mar 10 17:14 ..  
drwxr-xr-x  2 miguel444 miguel444 4096 mar 10 17:20 html  
miguel444@m1:/$ _
```

Máquina Virtual M1

Máquina Virtual M2

```
miguel444@m2:~$ rsync -avz -e ssh 192.168.56.12:/var/www/ /var/www/  
miguel444@192.168.56.12's password:  
receiving incremental file list  
./  
html/  
html/ejemplo.html  
html/index.html  
  
sent 169 bytes  received 294 bytes  71.23 bytes/sec  
total size is 10,987  speedup is 23.73  
miguel444@m2:~$ ls -la /var/www  
total 12  
drwxr-xr-x  3 miguel444 miguel444 4096 mar 10 17:20 .  
drwxr-xr-x 14 root      root      4096 mar 10 17:15 ..  
drwxr-xr-x  2 miguel444 miguel444 4096 mar 10 17:20 html  
miguel444@m2:~$ _
```

También se ha probado otra funcionalidad que nos ofrece la herramienta **RSYNC** con la que podemos excluir archivos en el proceso de copia, de manera que no se sobrescriban los archivos que no queramos.

- `rsync -avz --delete --exclude=**/stats --exclude=**/error --exclude=**/files/pictures -e ssh maquina1:/var/www/ /var/www/`

```
miguel444@m2:~$ rsync -avz --delete --exclude=**/stats --exclude=**/error --exclude=**/files/pictures -e ssh 192.168.56.12:/var/www/ /var/www/
miguel444@192.168.56.12's password:
receiving incremental file list
deleting html/ejemplo2.html

sent 72 bytes  received 142 bytes  47.56 bytes/sec
total size is 10,987  speedup is 51.34
miguel444@m2:~$
```

De esta forma estaremos haciendo la copia completa del directorio /var/www pero excluyendo /var/www/error, /var/www/stats y /var/www/files/pictures

La opción --delete indica que aquellos ficheros que se hayan eliminado en la máquina origen, también se borren en la máquina destino (para que el clonado sea perfecto).

La opción --exclude indica que ciertos directorios o ficheros no deben copiarse (p.ej., archivos de log). De esta forma, en la orden de ejemplo anterior, cuando indicamos --exclude=**/error significa "no copies lo que hay en /var/www/error" ya que corresponde a mensajes de la máquina original (y la segunda máquina ya generará sus propios mensajes).

3. Configuración de ssh para acceder sin que solicite contraseña.

Como hemos observado en la práctica anterior y en esta, cada vez que intentamos acceder mediante acceso remoto **SSH** nos pide la contraseña del usuario remoto. En trabajos donde se automatice el proceso de clonado y sincronización entre máquinas, este proceso puede quedar bloqueado debido a la petición de contraseña. En este apartado vamos a intentar solventar este problema:

Para ello usaremos la autenticación clave pública-privada, inicialmente debemos generar dichas claves en las máquinas:

- `ssh-keygen -b 4096 -t rsa`

```
miguel444@m2:~$ ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/miguel444/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/miguel444/.ssh/id_rsa.
Your public key has been saved in /home/miguel444/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:X6MHb1lg2AodW2mzZ1ti0Ao0UteofRyE4T8ij4r8LjU miguel444@m2
The key's randomart image is:
+---[RSA 4096]-----+
|      .o= +o.      |
|      o.XB +      |
|      . +==* .      |
|      ..o+oB .      |
|      S o=*+      |
|      E * B.o      |
|      . . . B .      |
|      ... . . .      |
|      o++          |
+-----[SHA256]-----+
miguel444@m2:~$
```

Ahora trataremos de copiar la clave pública de la máquina M2 en la máquina M1. Se trata de usar el comando `ssh-copy-id`, que viene integrado con el comando `ssh`. Lo que haremos es copiarla a la máquina principal M1 (a la que queremos acceder luego desde la secundaria M2) desde la máquina M2:

- `ssh-copy-id 192.168.56.12`

```
miguel444@m2:~$ ssh-copy-id 192.168.56.12
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/miguel444/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install all the new keys
miguel444@192.168.56.12's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh '192.168.56.12'"
and check to make sure that only the key(s) you wanted were added.

miguel444@m2:~$ _
```

Por lo que ahora finalmente podemos acceder de M2 a M1 sin necesidad de introducir la contraseña cada vez que utilicemos el servicio **SSH**:

- `ssh 192.168.56.12 (máquina M1)`

```

miguel444@m2:~$ ssh 192.168.56.12
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

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

System information as of Fri Mar 20 11:11:28 UTC 2020

System load:  0.0               Processes:            92
Usage of /:   33.7% of 9.78GB   Users logged in:     1
Memory usage: 64%              IP address for enp0s3: 10.0.2.15
Swap usage:   0%               IP address for enp0s8: 192.168.56.12

Pueden actualizarse 18 paquetes.
4 actualizaciones son de seguridad.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection
or proxy settings

Last login: Fri Mar 20 10:34:32 2020
miguel444@m1:~$ _

```

Otra funcionalidad que nos ofrece SSH es la de ejecutar comandos en remoto, para ello simplemente deberemos añadirlo al final del comando **SSH** para conectarnos:

- `ssh 192.168.56.12 uname -a` (máquina M1)

```

miguel444@m2:~$ ssh 192.168.56.12 uname -a
Linux m1 4.15.0-88-generic #88-Ubuntu SMP Tue Feb 11 20:11:34 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
miguel444@m2:~$ _

```

4. Establecer una tarea en cron que se ejecute cada hora para mantener actualizado el contenido del directorio /var/www entre las dos máquinas.

Cron es un administrador procesos en segundo plano que ejecuta procesos en el instante indicado en el fichero crontab.

Cron se ejecuta en background y revisa cada minuto la tabla del fichero /etc/crontab en búsqueda de tareas que se deban ejecutar (si ha llegado su momento). Podemos agregar nuevas tareas a cron para automatizar algunos procesos (p.ej. copias de seguridad).

Para ello, debemos editar el archivo /etc/crontab añadiéndole el comando con el que podemos mantener actualizado el contenido del directorio /var/www entre las dos máquinas.

Como hemos visto anteriormente podemos utilizar la herramienta **RSYNC** para realizar esta tarea, para ello como se ha dicho anteriormente es necesario modificar el fichero `/etc/crontab` y añadir el comando utilizado en los apartados anteriores:

Como se pide que se ejecute cada hora marcaremos todos los campos con un asterisco menos el de los minutos que marcaremos 01, indicando que se active cada hora de todos los días del mes y días de la semana en el primer minuto:

- `01 * * * * root rsync -avz -e ssh 192.168.56.12:/var/www/ /var/www/`

```
GNU nano 2.9.3 /etc/crontab Modified
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
01 * * * * root    rsync -avz -e ssh 192.168.56.12:/var/www/ /var/www/
#
```

Así quedará de forma automatizada mediante cron la sincronización del directorio entre las dos máquinas, realizando en este caso la máquina M2 la tarea de sincronización.