

# **Práctica 2: Clonar la información de un sitio web**

---

*Juan Manuel Castillo Nievas*

---

<b>1. Funcionamiento de la copia de archivos de ssh</b>	<b>2</b>
<b>2. Clonar una carpeta entre las dos máquinas</b>	<b>3</b>
<b>3. Acceso sin contraseña para ssh</b>	<b>4</b>
<b>4. Actualizar el contenido de /var/www cada hora</b>	<b>6</b>

## **1. Funcionamiento de la copia de archivos de ssh**

---

Para crear un directorio tar.gz directamente en m2 sin que se guarde en m1 se utiliza la siguiente instrucción:

```
jumacasni@m1:~$ tar czf - /var/www/html | ssh jumacasni@192.168.121.200 'cat > ~/archivo.tgz'
tar: Removing leading '/' from member names
jumacasni@192.168.121.200's password:
jumacasni@m1:~$ _
```

Y en m2 comprobamos que nos ha llegado el archivo comprimido *archivo.tgz*.

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

## 2. Clonar una carpeta entre las dos máquinas

---

Para clonar archivos entre las dos máquinas se va a utilizar la herramienta *rsync*. En mi caso, esta herramienta ya venía instalada en ambas máquinas.

Primero he hecho que el usuario sea el dueño de la carpeta */var/www*. Para ello, he ejecutado la siguiente instrucción en cada máquina:

```
jumacasni@m1:~$ sudo chown jumacasni:jumacasni -R /var/www
[sudo] password for jumacasni:
jumacasni@m1:~$
```

```
jumacasni@m2:~$ sudo chown jumacasni:jumacasni -R /var/www
[sudo] password for jumacasni:
jumacasni@m2:~$ _
```

A continuación voy a clonar la carpeta */var/www* de la máquina principal (m1) en la máquina secundaria (m2). Para ello, ejecuto la siguiente instrucción en m2:

```
jumacasni@m2:~$ rsync -avz -e ssh 192.168.121.199:/var/www /var/www
jumacasni@192.168.121.199's password:
receiving incremental file list
www/
www/html/
www/html/ejemplo.html
www/html/index.html

sent 74 bytes received 3,422 bytes 776.89 bytes/sec
total size is 10,986 speedup is 3.14
jumacasni@m2:~$
```

Como se puede comprobar, me ha pedido la clave de m1. Al mirar lo que hay en el directorio */var/www*, se puede ver que efectivamente se ha clonado con éxito:

```
jumacasni@m2:~$ ls -la /var/www
total 12
drwxr-xr-x 3 jumacasni jumacasni 4096 mar 17 15:58 .
drwxr-xr-x 14 root      root      4096 mar 17 15:56 ..
drwxr-xr-x 3 jumacasni jumacasni 4096 mar  4 19:52 www
jumacasni@m2:~$
```

### 3. Acceso sin contraseña para ssh

---

El objetivo es hacer un script que automáticamente haga el rsync, pero para ello hay que acceder sin contraseña a ssh.

En m2 he generado la clave con la siguiente instrucción:

```
jumacasni@m2:~$ ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jumacasni/.ssh/id_rsa):
/home/jumacasni/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jumacasni/.ssh/id_rsa.
Your public key has been saved in /home/jumacasni/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:1B3UbSYRpE+vNpAUqAqkUORJTd33Yn0W15NX8L4ynoA jumacasni@m2
The key's randomart image is:
+---[RSA 4096]-----+
| 0+0.00+++ ..=|
|. + .0..0+ = ..0|
|+ 0. + . = 0 0+|
|0 . + * .0 0 0.|
|.. . = S... 0 .|
|. . . . .|
|      +E . 0 .|
|. . 0 +|
|      0|
+-----[SHA256]-----+
jumacasni@m2:~$
```

El campo *passphrase* queda vacío porque añade seguridad a la clave privada pero si queremos que las máquinas accedan sin contraseña hay que dejarlo vacío.

Ahora hay que copiar la clave pública generada en m1. Para ello, en m2 se ejecuta el comando *ssh-copy-id*:

```

jumacasni@m2:~$ ssh-copy-id 192.168.121.199
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/jumacasni/.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
jumacasni@192.168.121.199's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh '192.168.121.199'"
and check to make sure that only the key(s) you wanted were added.

jumacasni@m2:~$ ssh 192.168.121.199
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 Tue Mar 17 16:09:20 UTC 2020

System load:  0.06               Processes:            90
Usage of /:   43.3% of 9.78GB    Users logged in:     1
Memory usage: 31%               IP address for enp0s3: 10.0.2.15
Swap usage:   0%                IP address for enp0s8: 192.168.121.199

Pueden actualizarse 14 paquetes.
0 actualizaciones son de seguridad.

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

Last login: Tue Mar 17 15:36:24 2020
jumacasni@m1:~$

```

En la captura de pantalla se puede ver además que al hacer de nuevo *ssh* no nos ha pedido ninguna contraseña.

## 4. Actualizar el contenido de /var/www cada hora

---

Para este apartado se utiliza la orden *crontab*. Hay que editar el archivo */etc/crontab* en la máquina m2 y añadir la línea:

```
00 * * * * root rsync -avz -e ssh 192.168.121.199:/var/www/ /var/www/
```

Esta línea indica que en el minuto 00 de cada hora (da igual hora, día, mes o día de la semana) se ejecute la instrucción *rsync* que es la misma que se ha utilizado al principio de esta práctica para clonar la carpeta */var/www*.

```
GNU nano 2.9.3 /etc/crontab
# /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 )
#
00 * * * * root    rsync -avz -e ssh 192.168.121.199:/var/www/ /var/www/
#
```

```
jumacasni@m2:~$ cat /etc/crontab
# /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 )
)
00 * * * * root    rsync -avz -e ssh 192.168.121.199:/var/www/ /var/www/
#
jumacasni@m2:~$
```