



SERVIDORES WEB DE ALTAS PRESTACIONES

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

*César Muñoz Reinoso*

Curso 2022-2023

17 de marzo de 2023

# Índice

<b>1. Copiar archivos locales en un equipo remoto</b>	<b>2</b>
1.1. Comando con pipe . . . . .	2
1.2. Comando separados con scp . . . . .	2
1.3. Copiar archivo de equipo remoto a equipo local . . . . .	3
<b>2. Rsync</b>	<b>4</b>
2.1. Sincronización carpeta /var/www . . . . .	4
2.2. Sincronización con exclusiones en carpeta /var/www . . . . .	4
<b>3. Acceso SSH sin contraseña</b>	<b>5</b>
3.1. Uso de ssh-copy-id con rsa . . . . .	5
3.2. Uso de ssh-copy-id con dsa . . . . .	7
3.3. Envío de clave con scp . . . . .	7
<b>4. Programar tareas con Crontab</b>	<b>8</b>
4.1. Actualización de /www/var . . . . .	8
4.2. Eliminación de archivos en /tmp . . . . .	8

# 1. Copiar archivos locales en un equipo remoto

## 1.1. Comando con pipe

Para mostrar las diferentes formas que existen de copiar ficheros entre equipos, creamos un archivo de texto. Primero ejecutamos dos comandos unidos con un pipe,

```
tar -czvf archivocomprimido.tgz archivoss.txt
```

, que comprime el documento de texto y lo manda por ssh al equipo remoto con dirección ip 192.168.56.3 y puerto 2200, ya que en la anterior práctica la cambiamos a dicho puerto.

```
ssh -p 2200 cesarmunoz@192.168.56.3 'cat > /archivocomprimido.tgz'
```

```
cesarmunoz@m1-cesarmunoz:~$ echo 'Informacion para el archivo ssh' > archivoss.txt
cesarmunoz@m1-cesarmunoz:~$ tar -czvf archivosscomprimido.tgz archivoss.txt | ssh -p 2200 cesarmunoz@192.168.56.3 'cat > ~/archivocomprimido.tgz'
cesarmunoz@m1-cesarmunoz:~$
```

```
cesarmunoz@m2-cesarmunoz:~$ ls
archivocomprimido.tgz
cesarmunoz@m2-cesarmunoz:~$ _
```

## 1.2. Comando separados con scp

También podemos comprimir el archivo con un comando independiente, y luego enviarlo al equipo remoto con scp

```
tar -czvf archivocomprimido.tgz archivoss.txt
```

En este caso con scp volvemos a especificar el puerto ssh (2200) y después de la IP, le decimos donde queremos que copie el archivo comprimido.

```
scp -P 2200 archivocomprimido2.tgz
cesarmunoz@192.168.56.3:/home/cesarmunoz/directorio_destino_ssh
```

```
cesarmunoz@m1-cesarmunoz:~$ tar -czvf archivosscomprimido2.tgz archivoss.txt
archivoss.txt
cesarmunoz@m1-cesarmunoz:~$ scp -P 2200 archivosscomprimido2.tgz cesarmunoz@192.168.56.3:/home/cesarmunoz/directorio_destino_ssh
archivosscomprimido2.tgz
100% 160 146.2KB/s 00:00
```

```
cesarmunoz@m2-cesarmunoz:~$ mkdir directorio_destino_ssh
cesarmunoz@m2-cesarmunoz:~$ cd directorio_destino_ssh/
cesarmunoz@m2-cesarmunoz:~/directorio_destino_ssh$ pwd
/home/cesarmunoz/directorio_destino_ssh
cesarmunoz@m2-cesarmunoz:~/directorio_destino_ssh$ ls
archivosshcomprimido2.tgz
```

### 1.3. Copiar archivo de equipo remoto a equipo local

Si queremos copiar un archivo de un equipo remoto al equipo local ejecutamos scp con la direccion del archivo y la ruta de destino, en nuestro caso en la que nos encontramos actualmente.

```
cesarmunoz@m2-cesarmunoz:~/directorio_destino_ssh$ touch archivo_transferencia.txt
cesarmunoz@m2-cesarmunoz:~/directorio_destino_ssh$ ls
archivosshcomprimido2.tgz  archivo_transferencia.txt
cesarmunoz@m2-cesarmunoz:~/directorio_destino_ssh$ _
```

```
cesarmunoz@m1-cesarmunoz:~$ scp -P 2200 cesarmunoz@192.168.56.3:/home/cesarmunoz/directorio_destino_
ssh/archivo_transferencia.txt .
archivo_transferencia.txt                                100%   0   0.0KB/s   00:00
cesarmunoz@m1-cesarmunoz:~$
```

## 2. Rsync

### 2.1. Sincronización carpeta /var/www

Para poder sincronizar el contenido de /var/www de M2 en el equipo M1 hacemos uso de la orden rsync. En ella utilizamos el parámetro `-delete` para eliminar archivos extraños desde el lado receptor, `-backup-dir=DIR` para establecer un directorio en el que se almacenarán las copias de seguridad de los archivos.

Los parámetros `-azvP` sirven para comprimir y enviar archivos, activar el verbose y mostrar el progreso de envío. Con el parámetro `-e` le decimos que queremos enviarlo por ssh, por el puerto 2200 y sincronizarlo con la carpeta /var/www/

```
rsync -delete -backup-dir=/var/www/html/backup -azvP -e 'ssh -p 2200' /var/www/
cesarmunoz@192.168.56.3:/var/www/
```

```
cesarmunoz@m1-cesarmunoz:~$ rsync --delete -backup --backup-dir=/var/www/html/backup -avzP -e 'ssh -p 2200' /var/www/ cesarmunoz@192.168.56.3:/var/www/
sending incremental file list
html/
html/backup/

sent 226 bytes received 20 bytes 164,00 bytes/sec
total size is 10.772 speedup is 43,79
```

### 2.2. Sincronización con exclusiones en carpeta /var/www

Si queremos que ciertas carpetas o archivos no se sincronicen con rsync, empleamos el parametro `-exclude=PATTERN` para establecer los archivos con un cierto patrón o nombre. Ahora el archivo `error.html` no se sincronizará en la máquina M2.

```
rsync -azvP -exclude=.error.html 'ssh -p 2200' /var/www/
cesarmunoz@192.168.56.3:/var/www/
```

```
cesarmunoz@m1-cesarmunoz:~$ touch /var/www/html/error.html
cesarmunoz@m1-cesarmunoz:~$ rsync -avzP --exclude="error.html" -e 'ssh -p 2200' /var/www/ cesarmunoz@192.168.56.3:/var/www/
sending incremental file list
html/

sent 187 bytes received 17 bytes 136,00 bytes/sec
total size is 10.772 speedup is 52,80
```

## 3. Acceso SSH sin contraseña

### 3.1. Uso de ssh-copy-id con rsa

Al igual que en la práctica anterior, para acceder por ssh sin contraseña al equipo remoto creamos la clave pública y lo enviamos con ssh-copy-id.

```
ssh-keygen -t rsa
```

```
cesarmunoz@m1-cesarmunoz:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/cesarmunoz/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/cesarmunoz/.ssh/id_rsa
Your public key has been saved in /home/cesarmunoz/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:aAVRgv4FJK1j/PmnrZ/zrL6L+GiJkymEkTwnPVDrCXw cesarmunoz@m1-cesarmunoz
The key's randomart image is:
+---[RSA 3072]-----+
|    ...+=0.    |
| ..  0.00      |
|..+0E  .  ..   |
|0+++B  0.      |
|0+++.00S       |
| . .  .+       |
| .   + 0       |
| . = 00.000    |
| . 00.=*0*0    |
+-----[SHA256]-----+
```

```
cesarmunoz@m1-cesarmunoz:~$ ssh-copy-id cesarmunoz@192.168.56.3
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/cesarmunoz/.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 alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys
cesarmunoz@192.168.56.3's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'cesarmunoz@192.168.56.3'"
and check to make sure that only the key(s) you wanted were added.
```

```
cesarmunoz@m1-cesarmunoz:~$ ssh cesarmunoz@192.168.56.3
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-67-generic x86_64)

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

System information as of dom 12 mar 2023 17:11:02 UTC

System load:  0.0               Processes:            110
Usage of /:   58.8% of 8.02GB    Users logged in:     1
Memory usage: 58%              IPv4 address for enp0s3: 10.0.2.15
Swap usage:   0%               IPv4 address for enp0s8: 192.168.56.3

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 16 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Last login: Sun Mar 12 17:07:19 2023 from 192.168.56.2
cesarmunoz@m2-cesarmunoz:~$ _
```

### 3.2. Uso de ssh-copy-id con dsa

También podemos utilizar una clave de otro de tipo, en este caso utilizamos una clase DSA de 1024 bits y lo enviamos también con ssh-copy-id.

```
cesarmunoz@m1-cesarmunoz:~/ssh$ ssh-keygen -b 1024 -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/cesarmunoz/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/cesarmunoz/.ssh/id_dsa
Your public key has been saved in /home/cesarmunoz/.ssh/id_dsa.pub
The key fingerprint is:
SHA256:d5zCkKrpqnozmEmUzN1kbDdS6BwHSxYfur8+0QNevSA cesarmunoz@m1-cesarmunoz
The key's randomart image is:
+----[DSA 1024]-----+
|  .==0                |
|  o0++ .              |
| o Xo=.. +            |
| * +.E + + . .       |
| . .. * S = +         |
| . . = 0 0 0          |
| .+ o.. .             |
| + +. ..              |
| +o.+oo.              |
+-----[SHA256]-----+
cesarmunoz@m1-cesarmunoz:~/ssh$ ssh-copy-id -p 2200 cesarmunoz@192.168.56.3
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/cesarmunoz/.ssh/id_dsa.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

Number of key(s) added: 1

Now try logging into the machine, with: "ssh -p '2200' 'cesarmunoz@192.168.56.3'"
and check to make sure that only the key(s) you wanted were added.
```

### 3.3. Envío de clave con scp

No siempre es necesario utilizar el comando ssh-copy-id. Ahora utilizamos el comando scp aprendido en la practica, copiando la clave del archivo authorized\_keys del equipo local en el mismo archivo en el equipo remoto.

```
cesarmunoz@m1-cesarmunoz:~/ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDR4e38p10e1KdoUJgao+7b6mWbnpnIp/QY9ANUm2UraS0NaUr.jQ5YdwuDDdfr1
N5/DvEbGGfXH6cm4y3yWrFzW5X1IhBNzXjQDeyaQKbDg48LkewR8649e21I7DenI1RP0wxkfYmJqCCM+8FFjLpaGLj42srsE1bhu
G+zoLdjJ9TbC25G6y6p2Y1tiPCjx2TzmXp05mJRFe6UhrTUQ1hKAisMHcT93FGJaG9GexbH6vr5ieay9JJLgi2Kwk2VRGY/MkbDZ
Iuw1bt0pm9H3oJn/AYJKPqvJ1jJb6t+TKR1x9xE0ny8rtIH+B54g3XF5Tx7DV7mvyBnfP4W7CM/OWxQp1Ib1T41N2Tx1QJGBRNC5
cKk6hFvztMkIgwef6TOIJUe54aKu9Njc/Ruc7eJBJSaX1+f16Lq762IHRN/LuX07P2hS9XLAFF/oPnkksq+NtoVPj/sBW5R4D8Ke
7Vkhw3j864wYNZEStoGGxsZTnW8yNcpWDsWCbWk5+KDwzAyRkGE= cesarmunoz@m1-cesarmunoz
cesarmunoz@m1-cesarmunoz:~/ssh$ scp -P 2200 /home/cesarmunoz/.ssh/authorized_keys cesarmunoz@192.16
8.56.3:/home/cesarmunoz/.ssh
authorized_keys                                100% 578    874.8KB/s   00:00
```



## 4. Programar tareas con Crontab

### 4.1. Actualización de /www/var

Podemos hacer uso de crontab para administrar procesos en segundo plano siguiendo una planificación. En el script sincronizar.sh, está el comando utilizado anteriormente para sincronizar las carpetas /var/www/ de ambos equipos. Le damos permisos de ejecución a root y editamos el archivo /etc/crontab. Añadimos una línea donde

```
0 * * * * root /home/cesarmunoz/sincronizar.sh
```

significa que root ejecute todos los días a cada hora en punto, en segundo plano el script sincronizar.sh.

```
GNU nano 6.2                                sincronizar.sh *
#!/bin/bash
rsync -avzP -e 'ssh -p 2200' /var/www/html/swap.html cesarmunoz@192.168.56.3:/var/www/html/swap.html
```

### 4.2. Eliminación de archivos en /tmp

Crontab también es muy útil por ejemplo para limpiar un directorio antes de cada clase de prácticas. Le damos permisos de ejecución a root y editamos el archivo /etc/crontab. Añadimos una línea donde

```
30 17 * * 3 root /home/cesarmunoz/limpiar_tmp.sh
```

significa que root ejecute todos los miércoles a las 17:30, en segundo plano el script limpiar\_tmp.sh.

```
GNU nano 6.2                                limpiar_tmp.sh
#!/bin/bash
rm -R /tmp
_
```

```
cesarmunoz@m1-cesarmunoz:~$ chmod 777 sincronizar.sh limpiar_tmp.sh
cesarmunoz@m1-cesarmunoz:~$ ls -l
total 24
-rw-rw-r-- 1 cesarmunoz cesarmunoz 160 mar 15 17:18 archivoss comprimido2.tgz
-rw-rw-r-- 1 cesarmunoz cesarmunoz 160 mar 15 17:15 archivoss comprimido.tgz
-rw-rw-r-- 1 cesarmunoz cesarmunoz 32 mar 15 17:15 archivoss.txt
-rw-rw-r-- 1 cesarmunoz cesarmunoz 0 mar 15 17:25 archivo_transferencia.txt
-rw-rw-r-- 1 cesarmunoz cesarmunoz 131 mar 12 16:16 cookie.txt
-rwxrwxrwx 1 cesarmunoz cesarmunoz 24 mar 16 20:05 limpiar_tmp.sh
-rwxrwxrwx 1 cesarmunoz cesarmunoz 115 mar 16 19:41 sincronizar.sh
```

```
GNU nano 6.2 /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
# You can also override PATH, but by default, newer versions inherit it from the environment
#PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
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 )
0 * * * * root /home/cesarmunoz/sincronizacion.sh
30 17 * * 3 root /home/cesarmunoz/limpiar_tmp.sh
#
```