

SERVIDORES WEB

DE ALTAS PRESTACIONES

José David Torres de las Morenas

Práctica 2

Vamos a crear un tar con ficheros locales en un equipo remoto:

Maquina 1 (máquina servidora):

```
ubuntu@ubuntu:~$ tar czf - /home/ubuntu/carpeta/ | ssh 192.168.157.129 'cat > ~/tar.tgz'
tar: Removing leading '/' from member names
The authenticity of host '192.168.157.129 (192.168.157.129)' can't be established.
ECDSA key fingerprint is SHA256:qwmPgR51I7jQxqol4/Pnq0lv4FdmFKhkxP9xAq/m514.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.157.129' (ECDSA) to the list of known hosts.
ubuntu@192.168.157.129's password:
ubuntu@ubuntu:~$
```

Máquina 2 (máquina cliente):

Vemos que el fichero se ha enviado (recibido en la maquina cliente) correctamente.

```
ubuntu@ubuntu:~$ ls
tar.tgz
```

Ahora vamos a instalar la herramienta rsync, utilizando la siguiente instrucción:

```
sudo apt-get install rsync
```

Para probar que funciona vamos a copiar una carpeta de un directorio de la segunda máquina en otro de la primera máquina.

```
ubuntu@ubuntu:/etc/ssh$ sudo rsync -avz --delete -e ssh root@192.168.157.128:/var/www /var/www
root@192.168.157.128's password:
receiving incremental file list
www/
www/carpeta/
www/html/
www/html/index.html
www/rsync/

sent 63 bytes received 3,406 bytes 991.14 bytes/sec
total size is 11,321 speedup is 3.26
ubuntu@ubuntu:/etc/ssh$
```

Vemos que se ha enviado correctamente:

```
ubuntu@ubuntu:/var/www$ ls  
html rsync
```

Acceso sin contraseña para ssh

Lo primero que vamos a hacer es en la máquina cliente ejecutar la siguiente instrucción para generar una nueva ssh key:

```
sudo ssh-keygen -t dsa
```

Tras esto, podemos enviar la configuración a la máquina servidora:

```
ubuntu@ubuntu:~$ ssh-copy-id -i .ssh/id_dsa.pub root@192.168.157.128  
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".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 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  
root@192.168.157.128's password:  
  
Number of key(s) added: 1  
  
Now try logging into the machine, with: "ssh 'root@192.168.157.128'"  
and check to make sure that only the key(s) you wanted were added.  
ubuntu@ubuntu:~$ _
```

Activar Windows
Ir a Configuración de PC par

Ahora probamos a loguearnos desde la máquina cliente:

```
ubuntu@ubuntu:~$ ssh 192.168.157.128 -l root  
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage
```

Vemos que ya no necesitamos introducir la contraseña, e incluso podemos ejecutar comandos en ella:

```
ubuntu@ubuntu:~$ ssh 192.168.157.128 -l root uname -a  
Linux ubuntu 4.4.0-31-generic #50-Ubuntu SMP Wed Jul 13 00:07:12 UTC 2016 x86_64 x86_64 x86_64 GNU/L  
inux
```

Programar tareas con crontab

Vamos a editar el archivo de crontab con la siguiente instrucción:

```
sudo nano /etc/crontab
```

Añadimos la siguiente línea para programar la tarea:

```
GNU nano 2.5.3      File: 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 )
#
0 * * * * root    rsync -avz --delete -e ssh root@192.168.157.128:/var/www /var/www
```