

# Práctica 2

---

**GitHub:** Lauragg

Comenzando por dónde lo dejamos en la práctica anterior, en esta práctica deberemos:

- Probar el funcionamiento de la copia de archivos por *ssh*.
- Clonado de una carpeta entre las dos máquinas.
- Configuración de *ssh* para acceder sin que solicite contraseña.
- Establecer una tarea en *cron* que se ejecute cada hora para mantener actualizado el contenido del directorio `/var/www` entre las dos máquinas.

Continuaremos, siguiendo las instrucciones del guión.

## Copia de archivos por ssh.

---

Creamos la carpeta con varios archivos y la enviamos:

```
lauragg@m2:~$ ls
imagen.png
lauragg@m2:~$ ls
imagen.png  tar.tgz
```

```
lauragg@m1:~$ tar czf - copiaP2 | ssh 192.168.56.200 'cat > ~/tar.tgz'
lauragg@192.168.56.200's password:
lauragg@m1:~$
```

Comprobamos que ha llegado con éxito:

```
lauragg@m1:~$ ls
imagen.png
lauragg@m1:~$ mkdir copiaP2
lauragg@m1:~$ touch copiaP2/ejemplo.txt copiaP2/copia.html copiaP2/archivos.php
lauragg@m1:~$ ls copiaP2
archivos.php  copia.html  ejemplo.txt
```

## Clonado de una carpeta entre dos máquinas (con rsync)

---

Lo primero que debemos de hacer es instalar *rsync* en ambas máquinas, utilizando el comando

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

Crearemos un archivo en la carpeta `/var/www` que será la carpeta que clonaremos, para comprobar que efectivamente se ha realizado bien.

```
lauragg@m1:/var/www$ cat html/prueba_clon.html
<html>
    <body>
        Se ha realizado correctamete la clonación.
    </body>
</html>
lauragg@m1:/var/www$ _
```

A continuación, procedemos a clonar el contenido en nuestra máquina m2. Para ello, primero deberemos darle permisos a nuestro usuario en la carpeta que queremos clonar en la propia m2 y, después, proceder.

```
lauragg@m2:/var/www$ sudo chown lauragg:lauragg -R /var/www/
lauragg@m2:/var/www$ rsync -avz -e ssh 192.168.56.100:/var/www/ /var/www/
lauragg@192.168.56.100's password:
receiving incremental file list
rsync: send_files failed to open "/var/www/html/.ejemplo.html.swp": Permission denied
html/
html/ejemplo.html
html/prueba_clon.html

sent 85 bytes  received 518 bytes  134.00 bytes/sec
total size is 23,355  speedup is 38.73
rsync error: some files/attrs were not transferred (see previous errors) (code 23) at
generator=3.1.2]
lauragg@m2:/var/www$ ls html
ejemplo.html  index.html  prueba_clon.html
lauragg@m2:/var/www$ cat html/prueba_clon.html
<html>
    <body>
        Se ha realizado correctamete la clonación.
    </body>
</html>
lauragg@m2:/var/www$
```

Nótese que tenemos un archivo temporal que no ha sido clonado, nuevamente esto es por la falta de permisos en la máquina m1 y podríamos solucionarlo procediendo a dar dichos permisos. Sin embargo, como no estamos interesados en dichos archivos temporales lo dejamos sin cambios.

## Acceso ssh sin contraseña

---

Ahora buscamos poder acceder a nuestra máquina m1 desde la máquina m2 sin necesidad de introducir ninguna contraseña, para esto utilizaremos el sistema de claves públicas y privadas. Para ello generamos primero las claves en nuestra m2:

```

lauragg@m2:~$ ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/lauragg/.ssh/id_rsa):
/home/lauragg/.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/lauragg/.ssh/id_rsa.
Your public key has been saved in /home/lauragg/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Vs309jqEmrIyriz0D0obcA0tGcs0S/01EbK77CaGmxg lauragg@m2
The key's randomart image is:
+---[RSA 4096]-----+
|
| . . . 0
| 0 . 0 . 0
| . * 0 . 0
| * + 0 + S . +
| = 0 0 . 0 + . . . .
| E 0 . . = 0 . .
| B + 0 0 + . 0 0 . .
| = X * 0 + + . 0 . .
+-----[SHA256]-----+
lauragg@m2:~$

```

Después, indicamos a nuestra m1 mediante `ssh-copy-id` que vamos a subir nuestra clave pública y al no poner argumentos estamos indicando que la tenemos guardada en nuestro pc en la ruta por defecto.

```

lauragg@m2:~$ ssh-copy-id 192.168.56.100
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/lauragg/.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
lauragg@192.168.56.100's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh '192.168.56.100'"
and check to make sure that only the key(s) you wanted were added.

lauragg@m2:~$ _

```

Finalmente, comprobamos que podemos acceder:

```
lauragg@m2:~$ ssh 192.168.56.100
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-91-generic x86_64)

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

System information as of Wed Apr  1 16:15:32 UTC 2020

System load:  0.0                       Processes:            96
Usage of /:   44.2% of 9.78GB           Users logged in:     1
Memory usage: 33%                       IP address for enp0s3: 10.0.2.15
Swap usage:   0%                       IP address for enp0s8: 192.168.56.100

 * Kubernetes 1.18 GA is now available! See https://microk8s.io for docs or
   install it with:

       sudo snap install microk8s --channel=1.18 --classic

 * Multipass 1.1 adds proxy support for developers behind enterprise
   firewalls. Rapid prototyping for cloud operations just got easier.

       https://multipass.run/

15 packages can be updated.
0 updates are security updates.

Last login: Wed Apr  1 15:49:55 2020 from 192.168.56.200
lauragg@m1:~$
```

## Utilizar cron para mantener actualizado el contenido de ambas carpetas.

Modificamos el archivo `/etc/crontab` y después reiniciamos el servicio con `sudo service cron restart`.

