

# PRÁCTICA 2: CLONAR LA INFORMACIÓN DE UN SITIO WEB

David Armenteros Soto

## INDICE

1. Introducción: Objetivos y cuestiones a resolver
2. Configuración de ssh: acceso sin contraseña
3. Copiar archivos por ssh mediante tar y scp
4. Clonado de carpetas: utilización de rsync
5. Programar tareas con crontab

### 1. Introducción: Objetivos y cuestiones a resolver

El objetivo principal de la práctica es configurar las máquinas virtuales para trabajar en modo espejo, consiguiendo que una máquina secundaria mantenga siempre actualizada la información que hay en la máquina servidora principal. Las tareas a conseguir son las siguientes:

- Copiar archivos mediante **ssh**
- Clonar contenidos entre máquinas (**rsync**)
- Configurar ssh para acceder a máquinas remotas sin contraseña
- Establecer tareas en cron (**crontab**)

### 2. Configuración de ssh: acceso sin contraseña

En este apartado, explicaré como podemos conectar dos máquinas mediante ssh. Realizaré una configuración básica y una un poco más avanzada incorporando más opciones a los comandos utilizados.

- En primer lugar, tenemos que generar el par de claves (pública y privada para ello utilizaremos el comando **ssh-keygen**. La opción más simple se muestra a continuación:

```
daarso98@m2-daarso98:~$ ssh-keygen
generating public/private rsa key pair.
Enter file in which to save the key (/home/daarso98/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/daarso98/.ssh/id_rsa.
Your public key has been saved in /home/daarso98/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:pg976I8H505jQ20FJPda3kGf1xDGnwyxCtP/IpB7i0A daarso98@m2-daarso98
The key's randomart image is:
----[RSA 2048]-----+
  ..+.0=+ |
  +..+.0.0|
  .00.0.=0+|
  . ++0. =.|
  . S0.....|
  . 000.+ .|
  E0*+0 0 . .|
  .B+.. . .|
  .+=+ |
  ----[SHA256]-----+
```

Podemos indicar el tipo de cifrado que queremos con la opción **‘-t’** y el tamaño con la opción **‘-b’**

```
daarso98@m1-daarso98:~/.ssh$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/daarso98/.ssh/id_rsa):
/home/daarso98/.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/daarso98/.ssh/id_rsa.
Your public key has been saved in /home/daarso98/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:j8+KsgLeKbhvF6poFhEz90B+1M02JkrcYsCKz39u3PU daarso98@m1-daarso98
The key's randomart image is:
+---[RSA 4096]-----+
|.O
|O++ .
|.O+B
|*+.O
|OB.= S
|=+= . O.
|+.=..O .....
|O=++O +..O E
|*=000*O ..O
+----[SHA256]-----+
```

Por defecto, las claves públicas y privadas se guardarán en los ficheros **id\_rsa** y **id\_rsa.pub** que encontraremos en el directorio **~/.ssh**

```
daarso98@m2-daarso98:~$ ls -l ~/.ssh
total 16
-rw----- 1 daarso98 daarso98 2302 Mar 16 18:42 authorized_keys
-rw----- 1 daarso98 daarso98    0 Mar 16 18:22 id_rsa
-rw-r--r-- 1 daarso98 daarso98    0 Mar 16 18:22 id_rsa.pub
-rw-r--r-- 1 daarso98 daarso98  222 Mar 16 18:30 known_hosts
```

Otra forma más avanzada de usar el comando es con la opción **‘-f’** indicando el fichero en el que se generaran nuestras claves. En mi caso estos ficheros se llamarán **mykey** y **mykey.pub**

```
daarso98@m1-daarso98:~/.ssh$ ssh-keygen -t rsa -b 4096 -f mykey
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in mykey.
Your public key has been saved in mykey.pub.
The key fingerprint is:
SHA256:F0z1EctkX4g2KJqAoPiVR13f2cH9pDtcTWZ1AGvDSD0 daarso98@m1-daarso98
The key's randomart image is:
+---[RSA 4096]-----+
|.. . . . .++00B=*|
|O . .O O+..+EB.BX|
|O O..O .+.*O**=|
|. . .O O .. +|
|. S . .O|
|. +|
|. |
+----[SHA256]-----+
```

```
daarso98@m1-daarso98:~/.ssh$ ls -l
total 8
-rw----- 1 daarso98 daarso98  0 Mar 16 18:09 authorized_keys
-rw----- 1 daarso98 daarso98  0 Mar 16 18:09 id_rsa
-rw-r--r-- 1 daarso98 daarso98  0 Mar 16 18:09 id_rsa.pub
-rw-r--r-- 1 daarso98 daarso98  0 Mar 16 18:09 known_hosts
-rw----- 1 daarso98 daarso98 3247 Mar 16 18:10 mykey
-rw-r--r-- 1 daarso98 daarso98  746 Mar 16 18:10 mykey.pub
daarso98@m1-daarso98:~/.ssh$
```

- A continuación, deberemos copiar la clave pública en el equipo remoto añadiéndola al fichero `~/.ssh/authorized_keys`, para ello utilizaremos el comando `ssh-copy-id`. La forma más sencilla es la siguiente:

```
daarso98@m1-daarso98:~/.ssh$ ssh-copy-id -p 2222 daarso98@192.168.56.105
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/daarso98/.ssh/id_rsa.pub"
The authenticity of host '[192.168.56.105]:2222 ([192.168.56.105]:2222)' can't be established.
ECDSA key fingerprint is SHA256:GnBGkU1Saiss2Xo4skQ2Y417rYvnpqVm9H4f91oihXw.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are al
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
daarso98@192.168.56.105's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh -p '2222' 'daarso98@192.168.56.105'"
and check to make sure that only the key(s) you wanted were added.
```

En caso de haber generado las claves en el paso anterior indicando el fichero, como es de esperar también tenemos que indicar el fichero a `ssh-copy-id` con la opción `-i`

```
daarso98@m1-daarso98:~/.ssh$ ssh-copy-id -p 2222 -i ~/.ssh/mykey daarso98@192.168.56.105
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/daarso98/.ssh/mykey.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are al
eady installed

/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist on the remote syste
.
(if you think this is a mistake, you may want to use -f option)
```

- Para conseguir una conexión por ssh sin contraseña tenemos que deshabilitar **PasswordAuthentication** del archivo de configuración `etc/ssh/sshd_config`

```
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no
```

## ■ Finalmente probamos que la conexión funciona correctamente

```
daarso98@m1-daarso98:/etc/ssh$ ssh -p 2222 daarso98@192.168.56.105
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-136-generic x86_64)

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

System information as of Tue Mar 16 18:01:10 UTC 2021

System load:  0.0          Processes:           96
Usage of /:   43.5% of 8.79GB Users logged in:      1
Memory usage: 34%         IP address for enp0s3: 10.0.2.15
Swap usage:   0%          IP address for enp0s8: 192.168.56.105

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

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Mar 16 17:56:23 2021 from 192.168.56.104
daarso98@m2-daarso98:~$ exit
logout
Connection to 192.168.56.105 closed.
```

Nuevamente, indicamos con la opción ‘-i’ el fichero de la clave, en caso de haber seguido los pasos anteriores avanzados.

```
daarso98@m1-daarso98:~/ssh$ ssh -p 2222 -i ~/ssh/mykey daarso98@192.168.56.105
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-136-generic x86_64)

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

System information as of Tue Mar 16 18:27:59 UTC 2021

System load:  0.03         Processes:           96
Usage of /:   43.5% of 8.79GB Users logged in:      1
Memory usage: 34%         IP address for enp0s3: 10.0.2.15
Swap usage:   0%          IP address for enp0s8: 192.168.56.105

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

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Mar 16 18:27:53 2021 from 192.168.56.104
daarso98@m2-daarso98:~$ exit
logout
Connection to 192.168.56.105 closed.
```

Llegados a este punto, nos planteamos como poder conectarnos vía ssh sin contraseña sin utilizar el comando `ssh-copy-id`, es decir realizandolo de manera manualmente

```
daarso98@m1-daarso98:~/.ssh$ cat ~/.ssh/id_rsa.pub | ssh -p 2222 daarso98@192.168.56.105 "cat >> ~/.ssh/authorized_keys"
```

```
daarso98@m1-daarso98:~$ cat ~/.ssh/mykey.pub | ssh -p 2222 daarso98@192.168.56.105 "cat >> ~/.ssh/authorized_keys"
```

Manualmente podemos enviar el contenido de la clave publica al archivo de `authorized_keys` via ssh

### 3. Copiar archivos por ssh mediante tar y scp

En esta sección, vamos a crear un fichero **tar.gz** en un equipo para enviarlo por **ssh** a otro. Para ello utilizamos el comando **tar**, con las opciones:

- **'-c'**: crea un nuevo archivo .tar
- **'-z'**: representa la compresión gzip
- **'-f'**: para indicar el nombre del archivo

```
daarso98@m1-daarso98:~$ tar czf - directorio | ssh -p 2222 -i ~/.ssh/mykey daarso98@192.168.56.105 "cat > ~/archivo.tgz"
```

Como resultado, si nos dirigimos a la otra máquina observamos que efectivamente se ha realizado correctamente:

```
daarso98@m2-daarso98:~$ ls -l
total 28
-rw-rw-r-- 1 daarso98 daarso98 121 Mar 16 18:49 archivo.tgz
```

Tambien es posible utilizando **scp** (hace uso de ssh). Utilizo el comando **'-P'** para indicar el puerto (ssh utiliza **'-p'**)

```
daarso98@m1-daarso98:~$ tar -czvf archivo2.tgz directorio
directorio/
```

```
daarso98@m1-daarso98:~$ scp -P 2222 -i ~/.ssh/mykey archivo2.tgz daarso98@192.168.56.105:~/archivo2.tgz
archivo2.tgz 100% 121 60.1KB/s 00:00
```

```
daarso98@m2-daarso98:~$ ls -l
total 32
-rw-rw-r-- 1 daarso98 daarso98 121 Mar 16 18:49 archivo.tgz
-rw-rw-r-- 1 daarso98 daarso98 121 Mar 16 19:02 archivo2.tgz
```

Directamente podemos hacer uso de la opción ‘-r’ para indicar el directorio que queremos copiar. En mi caso el nombre del directorio es directorio

```
daarso98@m1-daarso98:~$ scp -P 2222 -i ~/.ssh/mykey -r directorio daarso98@192.168.56.105:~/director
io
a.txt 100% 0 0.0KB/s 00:00
```

Como resultado, vemos como se ha copiado el directorio y el fichero a.txt

```
daarso98@m2-daarso98:~$ ls -l
total 36
-rw-rw-r-- 1 daarso98 daarso98 121 Mar 16 18:49 archivo.tgz
-rw-rw-r-- 1 daarso98 daarso98 121 Mar 16 19:02 archivo2.tgz
-rw-rw-r-- 1 daarso98 daarso98 416 Mar 12 11:29 cookie.txt
drwxrwxrwx 2 daarso98 daarso98 4096 Mar 16 19:08 directorio
-rw-rw-r-- 1 daarso98 daarso98 6748 Mar 12 11:28 imagen.png
-rw-rw-r-- 1 daarso98 daarso98 6748 Mar 12 11:29 logo3w.png
drwxrwxr-x 2 daarso98 daarso98 4096 Mar 3 21:40 prueba
```

```
daarso98@m2-daarso98:~/directorio$ ls -l
total 0
-rw-rw-r-- 1 daarso98 daarso98 0 Mar 17 19:24 a.txt
```

Las herramientas tar y scp nos ofrecen multitud de **opciones más avanzadas** disponibles para el usuario. Por ejemplo, podemos crear un archivo en formato **tar.bz2** con las nuevas opciones ‘-v’ que muestra una descripción detallada del progreso y ‘-j’ que es la que indica que la compresión se va a realizar en bzip2

```
daarso98@m1-daarso98:~$ tar -cvjf archivo.tar.bz2 directorio
directorio/
directorio/a.txt
```

Podemos descomprimir un archivo. La opción ‘-x’ indica que hay que extraer el fichero, además nos muestra que el contenido es directorio/

```
daarso98@m1-daarso98:~$ tar -xvf archivo2.tgz
directorio/
```

## 4. Clonado de carpetas: utilización de rsync

La herramienta **rsync** nos va a permitir clonar carpetas, de tal forma que una máquina será el espejo de la otra. Para empezar a trabajar con esta herramienta, se instalará de los repositorios de ubuntu haciendo uso del comando **sudo apt-get install rsync**.

En primer lugar, haremos que el usuario sea dueño de la carpeta donde residen los archivos con el siguiente comando:

```
daarso98@m1-daarso98:~$ sudo chown daarso98:daarso98 -R /var/www
```

Para probar el funcionamiento vamos a clonar una carpeta, en este caso **/var/www/**, la máquina 2 hará de espejo de la máquina 1. Utilizaremos las siguientes opciones:

- **'-a'**: copiar archivos y directorios recursivamente
- **'-v'**: modo verboso
- **'-z'**: comprime datos de los archivos al transferir
- **'-e'**: para ejecutar comando de forma remota

```
daarso98@m2-daarso98:~$ rsync -avz -e 'ssh -p 2222 -i ~/.ssh/mykey' 192.168.56.104:/var/www/ /var/w
www/
receiving incremental file list
./
html/
html/ejemplo.html
html/index.html
```

Como vemos se ha clonado el directorio, incluyendo los ficheros *ejemplo.html* e *index.html* que antes no aparecían

```
daarso98@m2-daarso98:~$ ls -la /var/www/html/
total 24
drwxr-xr-x 2 daarso98 daarso98 4096 Mar  3 21:10 .
drwxr-xr-x 3 daarso98 daarso98 4096 Mar  3 20:55 ..
-rw-r--r-- 1 daarso98 daarso98   97 Mar  3 21:09 ejemplo.html
-rw-r--r-- 1 daarso98 daarso98 10918 Mar  3 20:59 index.html
```

La herramienta rsnc nos permite especificar que directorios copiar y cuales ignorar con las opciones:

- **--delete**: indica que aquellos ficheros que se hayan eliminado en la máquina origen, también se borren en la de destino.
- **--exclude**: para ignorar un archivo o directorio

Para realizar una prueba he creado dos archivos [a.html](#) y [b.html](#) en el directorio [/var/www/html](#) , voy a ignorar uno de ellos.

```
daarso98@m2-daarso98:~$ rsync -avz --delete --exclude=**/html/a.html -e 'ssh -p 2222 -i ~/.ssh/mykey' 192.168.56.104:/var/www/ /var/www
receiving incremental file list
html/
html/b.html

sent 67 bytes  received 214 bytes  187.33 bytes/sec
total size is 11,015  speedup is 39.20
```

```
daarso98@m2-daarso98:~$ ls -la /var/www/html
total 24
drwxr-xr-x 2 daarso98 daarso98 4096 Mar 17 15:45 .
drwxr-xr-x 3 daarso98 daarso98 4096 Mar  3 20:55 ..
-rw-rw-r-- 1 daarso98 daarso98    0 Mar 17 15:45 b.html
-rw-r--r-- 1 daarso98 daarso98   97 Mar  3 21:09 ejemplo.html
-rw-r--r-- 1 daarso98 daarso98 10918 Mar  3 20:59 index.html
```

A continuación, utilizaré la opción **-include** para incluir el fichero a.html que anteriormente había ignorado.

```
daarso98@m2-daarso98:~$ rsync -avz --delete --include=**/html/a.html -e 'ssh -p 2222 -i ~/.ssh/mykey' 192.168.56.104:/var/www/ /var/www
receiving incremental file list
html/a.html

sent 64 bytes  received 241 bytes  610.00 bytes/sec
total size is 11,015  speedup is 36.11
daarso98@m2-daarso98:~$ ls -la /var/www/html/
total 24
drwxr-xr-x 2 daarso98 daarso98 4096 Mar 17 15:45 .
drwxr-xr-x 3 daarso98 daarso98 4096 Mar  3 20:55 ..
-rw-rw-r-- 1 daarso98 daarso98    0 Mar 17 15:45 a.html
-rw-rw-r-- 1 daarso98 daarso98    0 Mar 17 15:45 b.html
-rw-r--r-- 1 daarso98 daarso98   97 Mar  3 21:09 ejemplo.html
-rw-r--r-- 1 daarso98 daarso98 10918 Mar  3 20:59 index.html
```

Otra de las opciones más utilizadas es **--remove-source-files** , elimina los archivos de la máquina de origen. En este caso he creado en la máquina 1 un directorio llamado [rsync\\_dir](#) con dos archivos [backup.txt](#) y [backup2.txt](#) y he clonado en un directorio de la máquina 2 llamado [directorio](#)

```
daarso98@m2-daarso98:~/directorio$ rsync -avz --remove-source-files -e 'ssh -p 2222 -i ~/.ssh/mykey' 192.168.56.104:/home/daarso98/rsync_dir/ /home/daarso98/directorio/
receiving incremental file list
backup.txt
backup2.txt

sent 78 bytes  received 228 bytes  204.00 bytes/sec
total size is 85  speedup is 0.28
daarso98@m2-daarso98:~/directorio$ ls -l
total 4
-rw-rw-r-- 1 daarso98 daarso98 85 Mar 17 16:10 backup.txt
-rw-rw-r-- 1 daarso98 daarso98  0 Mar 17 16:10 backup2.txt
```



Como era de esperar el directorio se queda vacío:

```
daarso98@m1-daarso98:~/rsync_dir$ ls -l
total 0
```

Podemos utilizar también **--min-size** para indicar que solo quiero clonar archivos a partir de un determinado tamaño

```
daarso98@m2-daarso98:/var/www/html$ ls -l
total 0
```

```
daarso98@m2-daarso98:/var/www/html$ rsync -avz --min-size=1KB -e 'ssh -p 2222 -i ~/.ssh/mykey' 192.168.56.104:/var/www/ /var/www
receiving incremental file list
html/
html/index.html

sent 47 bytes  received 3,367 bytes  2,276.00 bytes/sec
total size is 11,015  speedup is 3.23
daarso98@m2-daarso98:/var/www/html$ ls -la
total 20
drwxr-xr-x 2 daarso98 daarso98 4096 Mar 17 15:45 .
drwxr-xr-x 3 daarso98 daarso98 4096 Mar  3 20:55 ..
-rw-r--r-- 1 daarso98 daarso98 10918 Mar  3 20:59 index.html
```

Solo index.html tiene un tamaño mínimo de 1KB, el fichero ejemplo.html , a.html y b.html no cumplían esa característica

## 5. Programar tareas con crontab

Para programar tareas con **crontab** debemos editar el fichero **/etc/crontab** o utilizar el comando **crontab -e**. Recordemos que el fichero tiene la siguiente estructura:

Minuto	Hora	DiaDelMes	Mes	DiaDeLaSemana	Usuario	Comando
--------	------	-----------	-----	---------------	---------	---------

```
GNU nano 2.9.3 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 )
#
```

```

GNU nano 2.9.3                                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 )
30 0 * * * root    /sbin/shutdown -h now
30 15 * * 4 root    rm /tmp/*
*/3 * * * * root    rm /tmp/*
#

```

He añadido las tres tareas explicadas en el guión:

- (1) 30 0 \* \* \* root /sbin/shutdown -h now
- (2) 30 15 \* \* 4 root rm /tmp/\*
- (3) \*/3 \* \* \* \* root rm /tmp/\*

Vamos a comprobar que funciona crontab, gracias a la tarea (3) que elimina ficheros de /tmp cada 3 minutos. Veremos que el archivo temporal.txt ha desaparecido

```

daarso98@m1-daarso98:/tmp$ ls
netplan_0sny2mmx  netplan_o0n1r1xm
netplan_20gsi6ku  netplan_pcoqoyv1
netplan_33cvihae  netplan_pcsbjebj
netplan_5hpo5fm0  netplan_pqsph01_
netplan_844pj_eg  netplan_qmciju6t
netplan__dq0mczn  netplan_t9c6245m
netplan_bxwcco90  netplan_vhm2e1_2
netplan_cqn15h2e  netplan_xrtslvao
netplan_e1a1ywi3  systemd-private-f3f8a13e674948caa7068d6a681f3cb6-apache2.service-HCX3kX
netplan_jd1pzaxh  systemd-private-f3f8a13e674948caa7068d6a681f3cb6-systemd-resolved.service-ER
netplan_nq5owcxw  systemd-private-f3f8a13e674948caa7068d6a681f3cb6-systemd-timesyncd.service-x
daarso98@m1-daarso98:/tmp$ touch temporal.txt

```

```

daarso98@m1-daarso98:~$ ls -l /tmp/
total 88
drwx----- 2 root root 4096 Mar 11 19:30 netplan_0sny2mmx
drwx----- 2 root root 4096 Mar 11 19:52 netplan_20gsi6ku
drwx----- 2 root root 4096 Mar 11 19:30 netplan_33cvihae
drwx----- 2 root root 4096 Mar 10 16:36 netplan_5hpo5fm0
drwx----- 2 root root 4096 Mar 11 19:30 netplan_844pj_eg
drwx----- 2 root root 4096 Mar 10 16:40 netplan__dq0mczn
drwx----- 2 root root 4096 Mar 11 19:38 netplan_bxwcco90
drwx----- 2 root root 4096 Mar 10 16:31 netplan_cqn15h2e
drwx----- 2 root root 4096 Mar 11 19:30 netplan_e1a1ywi3
drwx----- 2 root root 4096 Mar 10 16:13 netplan_jd1pzaxh
drwx----- 2 root root 4096 Mar 10 16:28 netplan_nq5owcxw
drwx----- 2 root root 4096 Mar 11 19:32 netplan_o0n1r1xm
drwx----- 2 root root 4096 Mar 10 16:31 netplan_pcoqoyv1
drwx----- 2 root root 4096 Mar 11 19:31 netplan_pcsbjebj
drwx----- 2 root root 4096 Mar 10 16:27 netplan_pqsph01_
drwx----- 2 root root 4096 Mar 10 16:42 netplan_qmciju6t
drwx----- 2 root root 4096 Mar 10 16:42 netplan_t9c6245m
drwx----- 2 root root 4096 Mar 11 19:39 netplan_vhm2e1_2
drwx----- 2 root root 4096 Mar 11 19:27 netplan_xrtslvao
drwx----- 3 root root 4096 Mar 10 15:36 systemd-private-f3f8a13e674948caa7068d6a681f3cb6-apache2.s
vice-HCX3kX
drwx----- 3 root root 4096 Mar 10 15:36 systemd-private-f3f8a13e674948caa7068d6a681f3cb6-systemd-r
solved.service-ER43m0
drwx----- 3 root root 4096 Mar 10 15:36 systemd-private-f3f8a13e674948caa7068d6a681f3cb6-systemd-t

```

Voy a realizar otro ejemplo añadiendo:

(4) 0 0 \* \* \* root /home/daarso98/Hola.sh >> /home/daarso98/Hola.txt

```
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 )
0 0 * * * root    /sbin/shutdown -h now
30 15 * * 4 root    rm /tmp/*
*/3 * * * * root    rm /tmp/*
0 0 * * * root    /home/daarso98/Hola.sh >> /home/daarso98/Hola.txt
#
```

Ejecuta una vez al día el script Hola.sh y redirige la salida al fichero Hola.txt

```
daarso98@m2-daarso98:~$ cat Hola.txt
Hola Mundo
Hola Mundo
Hola Mundo
Hola Mundo
Hola Mundo
```

Podemos planificar tareas con el comando **crontab -e** y ver las tareas con **contrab -l**

```
(5)* * * * * rsync -avz -e 'ssh -p 2222 -i ~/.ssh/mykey' 192.168.56.104:/var/www/  
/var/www/
```

Clona el directorio indicado de la máquina 1 a la máquina 2 cada minuto.

```
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
* * * * * rsync -avz -e 'ssh -p 2222 -i ~/.ssh/mykey' 192.168.56.104:/var/www/ /var/www/  
  
crontab: installing new crontab
```

```
daarso98@m2-daarso98:~$ crontab -l  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
* * * * * rsync -avz -e 'ssh -p 2222 -i ~/.ssh/mykey' 192.168.56.104:/var/www/ /var/www/
```

Aquí tenemos la comprobación:

```
daarso98@m2-daarso98:/var/www/html$ ls -l
total 16
-rw-rw-r-- 1 daarso98 daarso98  0 Mar 17 15:45 a.html
-rw-rw-r-- 1 daarso98 daarso98  0 Mar 17 15:45 b.html
-rw-r--r-- 1 daarso98 daarso98  97 Mar  3 21:09 ejemplo.html
-rw-r--r-- 1 daarso98 daarso98 10918 Mar  3 20:59 index.html
daarso98@m2-daarso98:/var/www/html$ rm a.html
daarso98@m2-daarso98:/var/www/html$ ls -l
total 16
-rw-rw-r-- 1 daarso98 daarso98  0 Mar 17 15:45 b.html
-rw-r--r-- 1 daarso98 daarso98  97 Mar  3 21:09 ejemplo.html
-rw-r--r-- 1 daarso98 daarso98 10918 Mar  3 20:59 index.html
daarso98@m2-daarso98:/var/www/html$ ls -l
total 16
-rw-rw-r-- 1 daarso98 daarso98  0 Mar 17 15:45 a.html
-rw-rw-r-- 1 daarso98 daarso98  0 Mar 17 15:45 b.html
-rw-r--r-- 1 daarso98 daarso98  97 Mar  3 21:09 ejemplo.html
-rw-r--r-- 1 daarso98 daarso98 10918 Mar  3 20:59 index.html
daarso98@m2-daarso98:/var/www/html$
```

Si quisieramos realizar esta tarea cada hora, tal y como indica el guión simplete escribir la siguiente entrada en crontab:

```
(6)0 * * * * rsync -avz -e 'ssh -p 2222 -i ~/.ssh/mykey' 192.168.56.104:/var/www/
/var/www/
```

Para verificar que para un usuario existe el archivo crontab podemos consultar en el directorio **/var/spool/cron/crontabs/**

```
daarso98@m2-daarso98:/var/www/html$ sudo ls -l /var/spool/cron/crontabs/
total 4
-rw----- 1 daarso98 crontab 1179 Mar 18 10:26 daarso98
```

La opción **crontab -r** elimina el contrab actual del usuario

```
* * * * * rsync -avz -e 'ssh -p 2222 -i ~/.ssh/mykey' 192.168.56.104:/var/www/ /var/www/
0 0 * * * touch z.txt

crontab: installing new crontab
daarso98@m2-daarso98:/etc$ crontab -r
daarso98@m2-daarso98:/etc$ crontab -l
no crontab for daarso98
```

Por defecto, crontab no tiene ningún tipo de restricción en cuanto a usuarios sin embargo existe la opción de restringir el acceso incluyendo usuario en los ficheros **/etc/cron.d/cron.allow** **/etc/cron.d/cron.deny** . Si modificamos estos ficheros es posible que tengamos que utilizar la opción **'-u'** para indicar el usuario.