

PRACTICA 2

INDICE

1. [Tar y scp](#)
2. [Rsync](#)
3. [SSH sin contraseña](#)
4. [Crontab](#)

1. TAR Y SCP

Lo primero que vamos a hacer es intentar enviar un directorio de se la maquina m1 a m2. Primero creamos la carpeta enviar_a_M2 en M1 y creamos 5 archivos dentro de esta, acto seguido usamos el comando:

- scp -r enviar_a_M2/ carlosgs@192.168.56.102:/home/carlosgs

Como podemos ver en las siguientes capturas la M1 envía los archivos a M2 y M2 los recibe correctamente.

```
M1-carlosgs (P2M1) [Corriendo] - Oracle VM VirtualBox
carlosgs@m1-carlosgs:~$ scp -r enviar_a_M2/ carlosgs@192.168.56.102:/home/carlosgs
carlosgs@192.168.56.102's password:
3.txt                                100%  6   11.3KB/s  00:00
2.txt                                100%  6   12.4KB/s  00:00
4.txt                                100%  6   13.7KB/s  00:00
5.txt                                100%  6   13.4KB/s  00:00
1.txt                                100%  6   19.2KB/s  00:00
carlosgs@m1-carlosgs:~$
```

```
M2-carlosgs (P2M2) [Corriendo] - Oracle VM VirtualBox
carlosgs@m1-carlosgs:~$ ls
enviar_a_M2  web
carlosgs@m1-carlosgs:~$ cd enviar_a_M2/
carlosgs@m1-carlosgs:~/enviar_a_M2$ ls
1.txt 2.txt 3.txt 4.txt 5.txt
carlosgs@m1-carlosgs:~/enviar_a_M2$
```

Ahora vamos a hacer lo mismo, pero creando un archivo.tgz y enviándolo por scp.

```
M1-carlosgs (P2M1) [Corriendo] - Oracle VM VirtualBox
carlosgs@m1-carlosgs:~$ tar -czvf archivo.tgz enviar_a_M2/
enviar_a_M2/
enviar_a_M2/3.txt
enviar_a_M2/2.txt
enviar_a_M2/4.txt
enviar_a_M2/5.txt
enviar_a_M2/1.txt
carlosgs@m1-carlosgs:~$ ls
archivo.tgz  enviar_a_M2  web
carlosgs@m1-carlosgs:~$ _
```

En las siguientes capturas vemos como M1 envía el archivo y M2 lo recibe. Con tar -x se descomprime el archivo y con scp -p se puede cambiar el puerto que usa.

```
M1-carlosgs (P2M1) [Corriendo] - Oracle VM VirtualBox
carlosgs@m1-carlosgs:~$ scp archivo.tgz carlosgs@192.168.56.102:/home/carlosgs/
carlosgs@192.168.56.102's password:
archivo.tgz                                100% 230  402.9KB/s  00:00
carlosgs@m1-carlosgs:~$ _
```

```
M2-carlosgs (P2M2) [Corriendo] - Oracle VM VirtualBox
carlosgs@m2-carlosgs:~$ ls
archivo.tgz  enviar_a_M2  hola
carlosgs@m2-carlosgs:~$
```

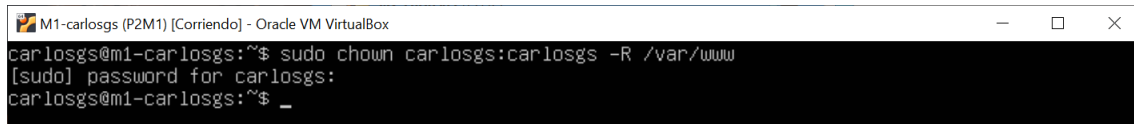
2. RSYNC

Lo primero que hacemos es instalarlo con el comando:

- `sudo apt-get install rsync.`

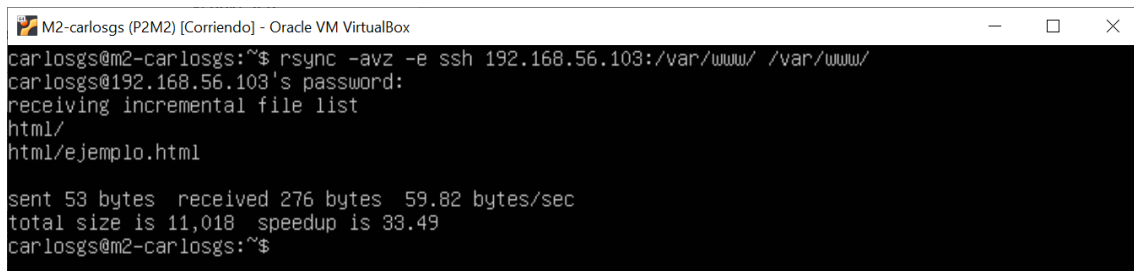
Una vez hecho esto tenemos que hacer que el usuario sea el dueño de la carpeta que queremos sincronizar en ambas maquinas. Para esto usamos el comando:

- `sudo chown carlosgs:carlosgs -R /var/www`

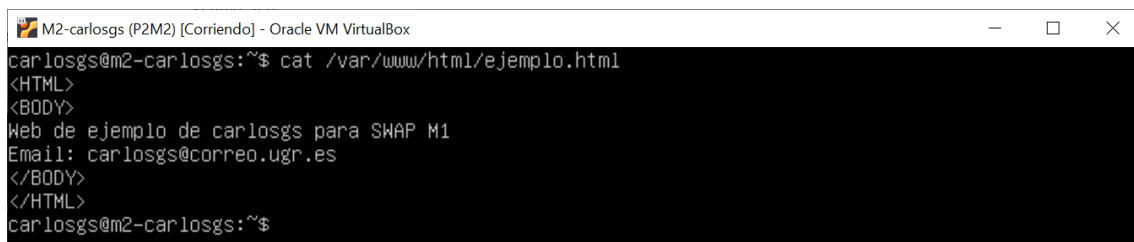
A terminal window titled 'M1-carlosgs (P2M1) [Corriendo] - Oracle VM VirtualBox'. The prompt is 'carlosgs@m1-carlosgs:~\$'. The user enters 'sudo chown carlosgs:carlosgs -R /var/www'. The prompt changes to '[sudo] password for carlosgs:'. The user enters a password (indicated by underscores). The prompt returns to 'carlosgs@m1-carlosgs:~\$'.

Ahora vamos a sincronizar las 2 carpetas (M1 tiene una copia idéntica de la carpeta de M2) usando rsync y los parámetros -a para activar el modo archivo, -v para mostrar el proceso de manera visual, -z para comprimir los datos durante la transferencia y el -e que sirve en nuestro caso para poder añadir la carpeta del ssh

- `rsync -avz -e ssh 192.168.56.103:/var/www/ /var/www/`

A terminal window titled 'M2-carlosgs (P2M2) [Corriendo] - Oracle VM VirtualBox'. The prompt is 'carlosgs@m2-carlosgs:~\$'. The user enters 'rsync -avz -e ssh 192.168.56.103:/var/www/ /var/www/'. The prompt changes to 'carlosgs@192.168.56.103's password:'. The user enters a password. The output shows 'receiving incremental file list', 'html/', and 'html/ejemplo.html'. Then it shows 'sent 53 bytes received 276 bytes 59.82 bytes/sec', 'total size is 11,018 speedup is 33.49', and the prompt returns to 'carlosgs@m2-carlosgs:~\$'.

Podemos ver que la maquina dos tiene los archivos de m1

A terminal window titled 'M2-carlosgs (P2M2) [Corriendo] - Oracle VM VirtualBox'. The prompt is 'carlosgs@m2-carlosgs:~\$'. The user enters 'cat /var/www/html/ejemplo.html'. The output shows the content of the file: '<HTML>', '<BODY>', 'Web de ejemplo de carlosgs para SWAP M1', 'Email: carlosgs@correo.ugr.es', '</BODY>', and '</HTML>'. The prompt returns to 'carlosgs@m2-carlosgs:~\$'.

Además, si usamos `--delete` podemos borrar los archivos eliminados en la carpeta de origen y con `--exclude` podemos establecer que carpetas y archivos no queremos que se sincronicen.

- `rsync -avz --delete -e ...`
- `rsync -avz --exclude=**/stats --exclude=**/error ...`

3. SSH sin contraseña

Para hacer esto vamos a crear una clave publica y privada en M2 y vamos a copiar manualmente la clave en M1. Para esto primero utilizamos el comando:

- `ssh-keygen -b 4096 -t rsa`

Con el `-b` elegimos el numero de bytes de la clave y con `-t` el tipo de clave.

```
M2-carlos (P2M2) [Corriendo] - Oracle VM VirtualBox
carlos@m2-carlos:~/.ssh$ ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/carlos/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/carlos/.ssh/id_rsa.
Your public key has been saved in /home/carlos/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:8E1m4372d53qcbEWU3u3bKS0//0iBDiF53BCdLcsnKo carlos@m2-carlos
The key's randomart image is:
+---[RSA 4096]-----+
|      oo.. .      |
|      +0++ .      |
|      . 00 o .      |
|      ooB00 o      |
|      S.o. . *o     |
|      . . 00= @     |
|      E  ..0+.0+    |
|      .. 0.o        |
|      .+ o*         |
+---[SHA256]-----+
carlos@m2-carlos:~/.ssh$
```

Como podemos observar ya tenemos los archivos que contienen nuestra clave primaria y privada.

```
M2-carlos (P2M2) [Corriendo] - Oracle VM VirtualBox
carlos@m2-carlos:~/.ssh$ ls
authorized_keys id_rsa id_rsa.pub known_hosts
carlos@m2-carlos:~/.ssh$
```

Ahora vamos a usar `scp` para enviar el archivo `id_rsa.pub` (contiene la clave publica) a M1 y lo añadimos a `authorized keys` usando

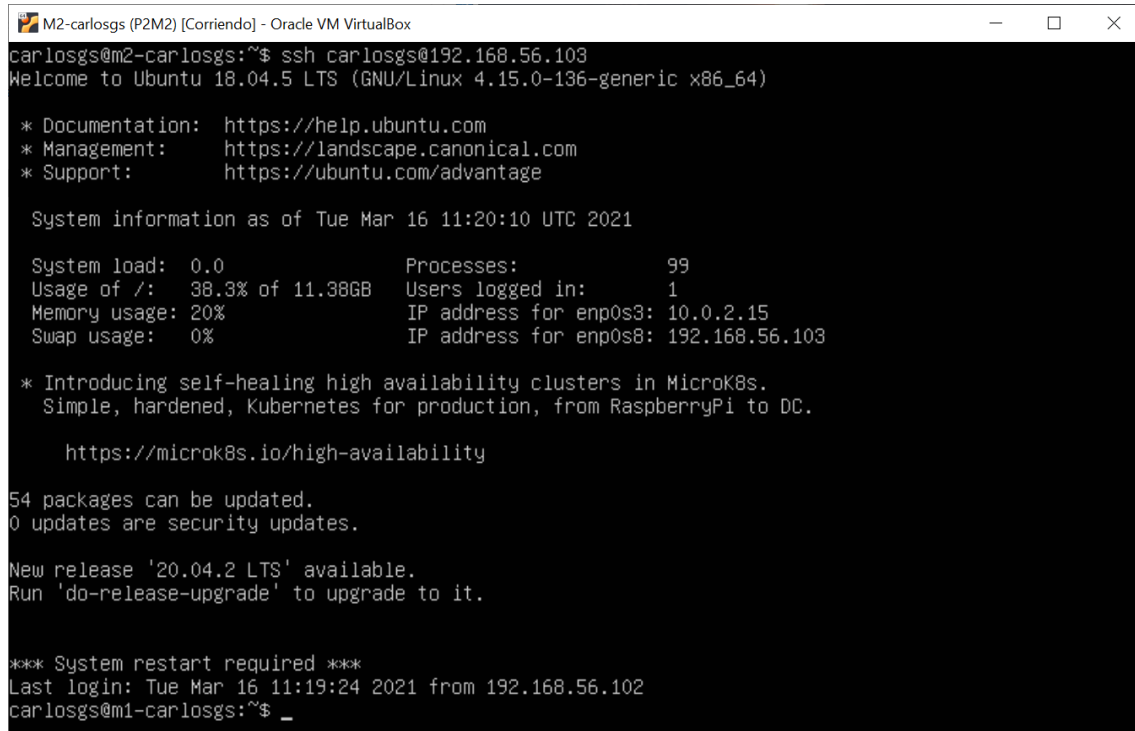
- `cat id_rsa.pub >> /home/carlos/.ssh/authorized_keys`

```
M1-carlos (P2M1) [Corriendo] - Oracle VM VirtualBox
carlos@m1-carlos:~/.ssh$ cat authorized_keys
carlos@m1-carlos:~/.ssh$ cd ..
carlos@m1-carlos:~$ cat id_rsa.pub >> /home/carlos/.ssh/authorized_keys
carlos@m1-carlos:~/.ssh$ cd .ssh/
carlos@m1-carlos:~/.ssh$ cat
^C
carlos@m1-carlos:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACnVcZuixdVCoIuwvpfp/hW/Ywcv7TfV6AraeaPH6NmAew35r51hf2KN32CPT6
sJmkDo6EwQ0da140pvQTpes0lwJaXvGmB5GLUAmVZtUpsMbrzcnhXPJu88kX1sMg1xKMee1xkfIFoy9NIxFa1RfcKS75pr2NhkQS
QQcQ2zV0eM5eB18g+F1QVD4xwa7QhhBEyCSVGAH/pmQyUvzdT3VPQyoI223x6frapMU0io/6uBKoiUrfMcKrf1fT0KAHD1rvC5BBi
FqQhYsJFmViEzK4AJDBDK/4r85b43P6Kc2+OS/1/0//JAjS2h2k.jLYG1d3a1h034ovaUPMugUL1EXfDJP8Rj1mu3NRnttIkrS+9I
0xSrR10752PibBHUUvDm6NS3NwNc8IM95+ahy0ucwN+ICAgtqDTFOFEM7GN16jToo2cGEavL5v15da1IwQ6R74op1fYtmprzeS
4dLUsGytqzLW+1qWnDzCoBrrnVbF2XT9U5kVZQ1bshNPER11wBVE+KCzoy8Dk2IFcx04XAhVwjr1CLWT9jNoYJXdrLW+59EgT63
5PBcyr1YqyCV9a+KeRKp+7BI5p4ur4b0Z5ICQvvfA9X9yhFdc/M3eBILyxivVR2Eo25xc47TECQQQfm68RgMJGibyWhT5pk3Tf/S
RtJ0MnsNAcDP4vQt0SpAwQ== carlos@m2-carlos
carlos@m1-carlos:~/.ssh$ ^C
carlos@m1-carlos:~/.ssh$ _
```

Una vez hecho esto nos debería de dejar hacer `ssh` y `rsync` sin pedirnos contraseña, vamos a probarlo.

Primero hacemos un

- ssh carlosgs@192.168.56.103



```
M2-carlosgs (P2M2) [Corriendo] - Oracle VM VirtualBox
carlosgs@m2-carlosgs:~$ ssh carlosgs@192.168.56.103
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 11:20:10 UTC 2021

System load:  0.0               Processes:    99
Usage of /:   38.3% of 11.38GB   Users logged in: 1
Memory usage: 20%              IP address for enp0s3: 10.0.2.15
Swap usage:   0%                IP address for enp0s8: 192.168.56.103

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

   https://microk8s.io/high-availability

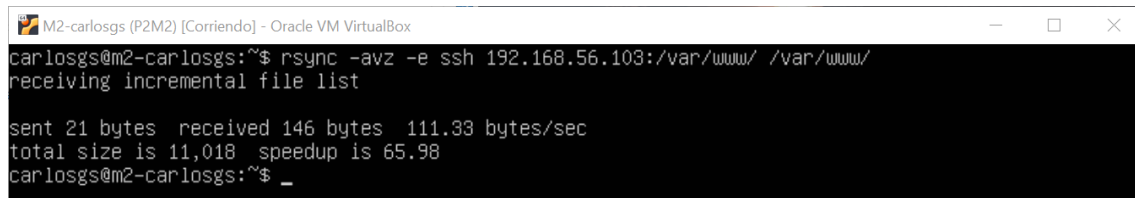
54 packages can be updated.
0 updates are security updates.

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

*** System restart required ***
Last login: Tue Mar 16 11:19:24 2021 from 192.168.56.102
carlosgs@m1-carlosgs:~$ _
```

Vemos que entramos sin problema, probemos ahora con:

- rsync -avz -e ssh 192.168.56.103:/var/www/ /var/www/



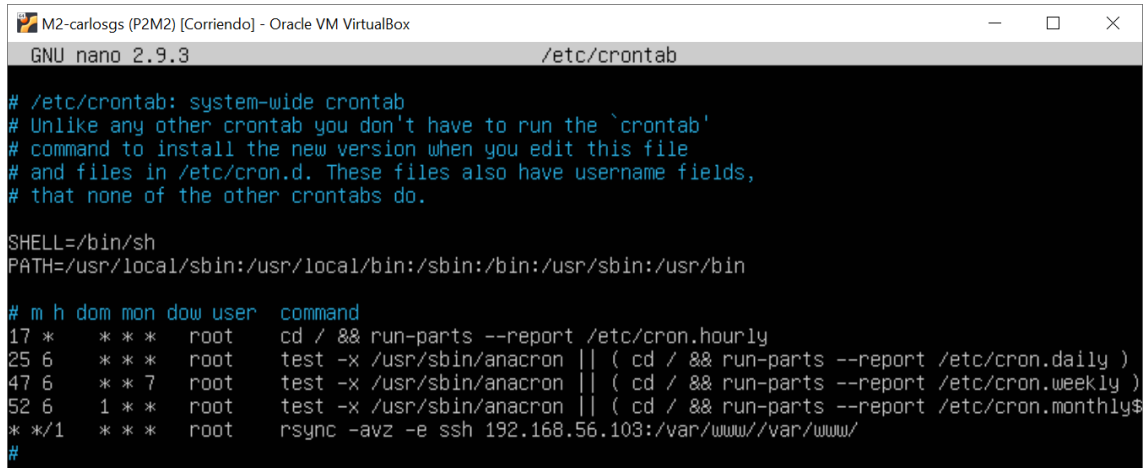
```
M2-carlosgs (P2M2) [Corriendo] - Oracle VM VirtualBox
carlosgs@m2-carlosgs:~$ rsync -avz -e ssh 192.168.56.103:/var/www/ /var/www/
receiving incremental file list

sent 21 bytes  received 146 bytes  111.33 bytes/sec
total size is 11,018  speedup is 65.98
carlosgs@m2-carlosgs:~$ _
```

Vemos que funciona perfectamente.

4. Crontab

Para programar el rsync en segundo plano con crontab, modificamos el archivo “etc/crontab” aunque también podríamos usar crontab -e para modificar una tarea.



The screenshot shows a terminal window titled "M2-carlosrgs (P2M2) [Corriendo] - Oracle VM VirtualBox". The terminal is running the GNU nano 2.9.3 editor, editing the file /etc/crontab. The content of the file is as follows:

```
# /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 )
* */1 * * * root    rsync -avz -e ssh 192.168.56.103:/var/www/ /var/www/
#
```

He añadido la línea:

```
- * */1 * * * root    rsync -avz -e ssh 192.168.56.103:/var/www/ /var/www/
```

La cual nos dice que en cualquier minuto, pasada una hora, ejecuta el comando rsync... con el usuario root.