

Práctica 2 SWAP

En esta práctica se trata de aprender a clonar información entre nuestras dos máquinas y realizar una sincronización automática de los contenidos de `/var/www` entre ambas.

1. Copia de un directorio

Como se apunta en el guión de la práctica, existen varias formas de copiar archivos de una máquina a otra. En concreto se apuntan dos: una mediante SSH y otra mediante SCP, la cual hace copias seguras y encriptadas de los archivos que deseemos usando SSH.

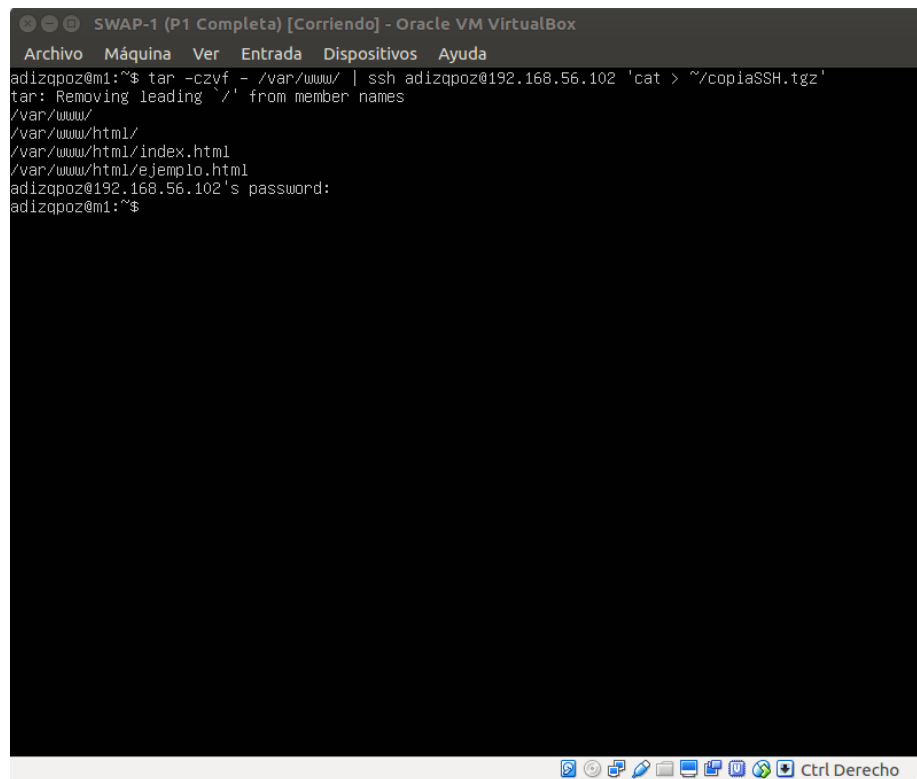
Debido a la simplicidad de su uso, priorizaremos el uso de SCP. Sin embargo usaremos ambos métodos en esta práctica, dado que se exponen ambos.

1.1. tar + SSH

Este método tiene como ventaja que es posible copiar la compresión de un directorio sin necesidad de crear el archivo comprimido en el equipo emisor, aspecto útil en caso de tener elevadas restricciones de almacenamiento en nuestro servidor. Para ello usaremos el siguiente comando:

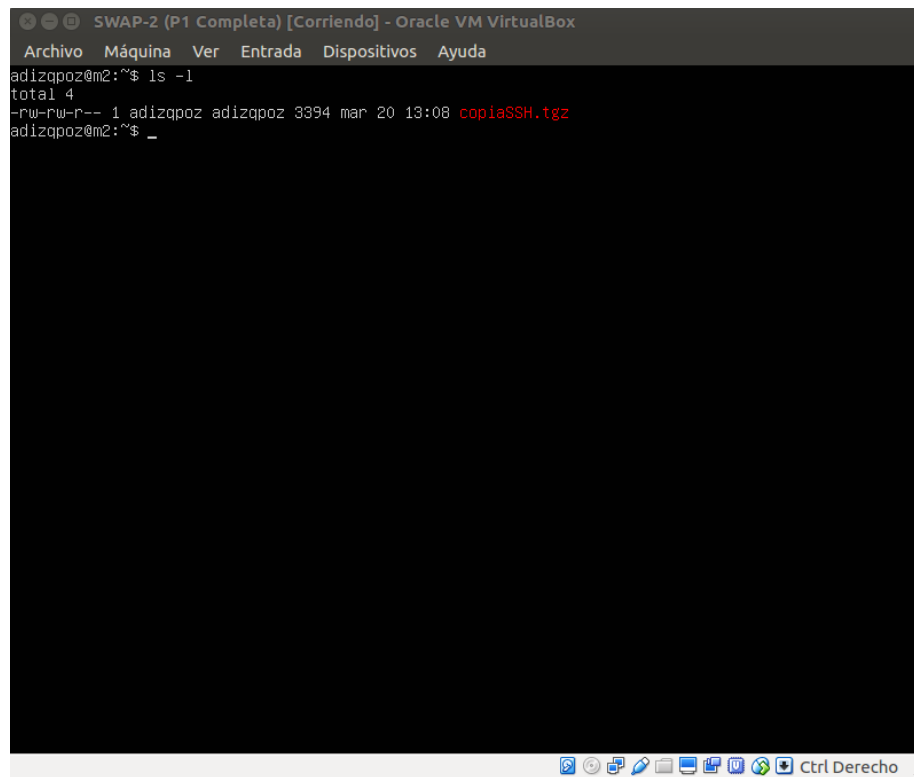
```
tar -czvf - directorio | ssh usuario@equiporemoto 'cat > ~/archivo.tgz'
```

En nuestro caso, *directorio* será `/var/www/`, *archivo* será `copiaSSH` y *equiporemoto* será "192.168.56.102", es decir, nuestra M2.



```
SWAP-1 (P1 Completa) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
adizqpoz@m1:~$ tar -czvf - /var/www/ | ssh adizqpoz@192.168.56.102 'cat > ~/copiaSSH.tgz'
tar: Removing leading '/' from member names
/var/www/
/var/www/html/
/var/www/html/index.html
/var/www/html/ejemplo.html
adizqpoz@192.168.56.102's password:
adizqpoz@m1:~$
```

Comprobamos en la máquina destino que el archivo ha llegado correctamente.

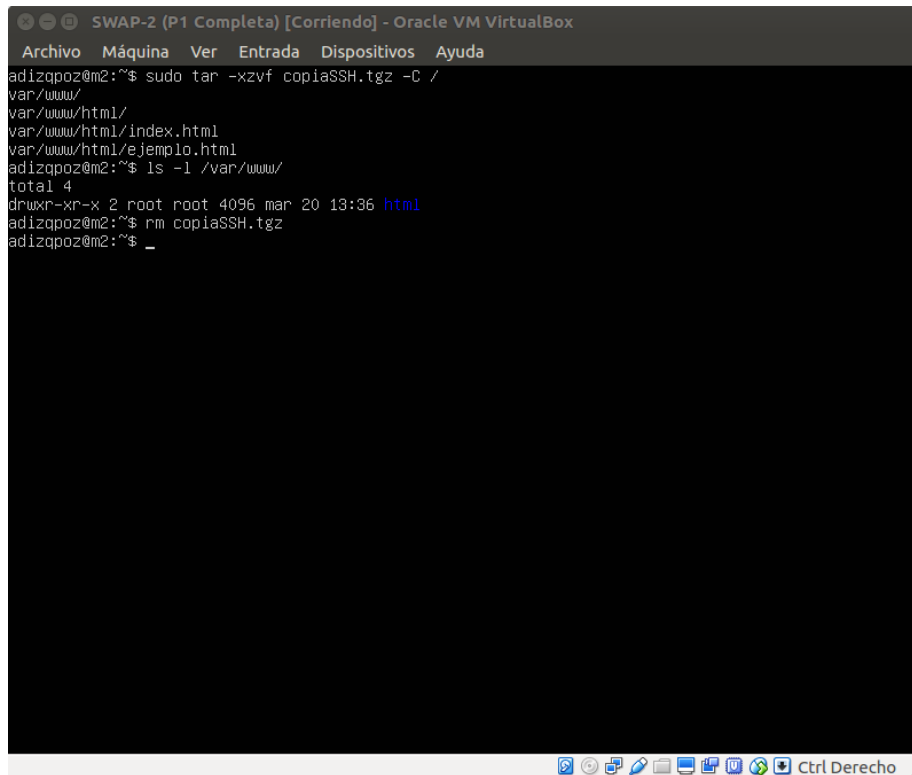


```
SWAP-2 (P1 Completa) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
adizqp0z@m2:~$ ls -l
total 4
-rw-rw-r-- 1 adizqp0z adizqp0z 3394 mar 20 13:08 copiaSSH.tgz
adizqp0z@m2:~$ _
```

Y por último extraemos el contenido del archivo comprimido en la ruta correspondiente con el siguiente comando:

```
tar -xzf archivo.tgz -C /
```

Siendo nuestro archivo el mismo que hemos traído desde M1. Aplicando el comando y comprobando que todo se ha ejecutado correctamente vemos lo siguiente:



```
SWAP-2 (P1 Completa) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
adizqpoz@m2:~$ sudo tar -xzf copiaSSH.tgz -C /
var/www/
var/www/html/
var/www/html/index.html
var/www/html/ejemplo.html
adizqpoz@m2:~$ ls -l /var/www/
total 4
drwxr-xr-x 2 root root 4096 mar 20 13:36 html
adizqpoz@m2:~$ rm copiaSSH.tgz
adizqpoz@m2:~$ _
```

Para comprobar que se ha copiado correctamente nos fijamos en la fecha de última modificación que nos aparece al ejecutar ls sobre el directorio `/var/www/`. Además, hemos borrado el archivador una vez hemos comprobado que todo se ha realizado correctamente.

SCP

Este método, como hemos adelantado, permite realizar copias seguras y encriptadas de los archivos y directorios que se quiera transferir de un equipo a otro a través de SSH.

Esta alternativa nos ofrece igual protección ante ataques que la anterior, pero requiere menor uso de instrucciones que el anterior método, ya que con un solo comando realizamos exactamente la misma labor que con la serie de comandos y comprobaciones que usamos en el anterior apartado.

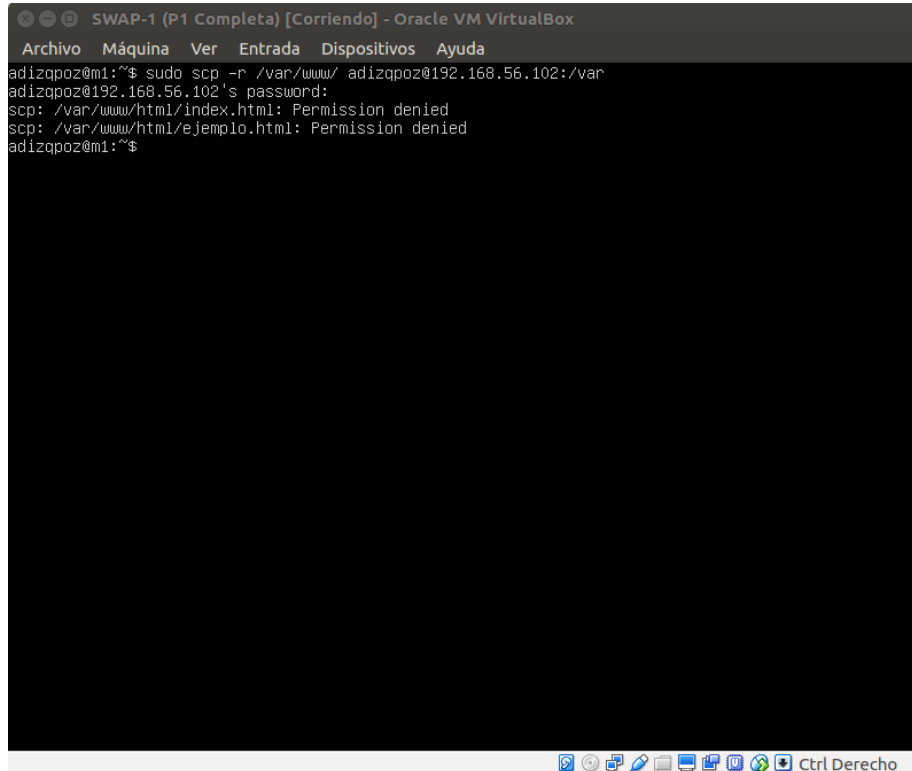
Este comando es el siguiente:

```
scp -r directorio usuario@equiporemoto:/directorio
```

En nuestro caso, el comando que hemos de usar es el siguiente:

```
sudo scp -r /var/www/ adizqpoz@192.168.56.102:/var
```

Y la salida que nos proporciona es la siguiente:



```
SWAP-1 (P1 Completa) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
adizqpoz@m1:~$ sudo scp -r /var/www/ adizqpoz@192.168.56.102:/var
adizqpoz@192.168.56.102's password:
scp: /var/www/html/index.html: Permission denied
scp: /var/www/html/ejemplo.html: Permission denied
adizqpoz@m1:~$
```

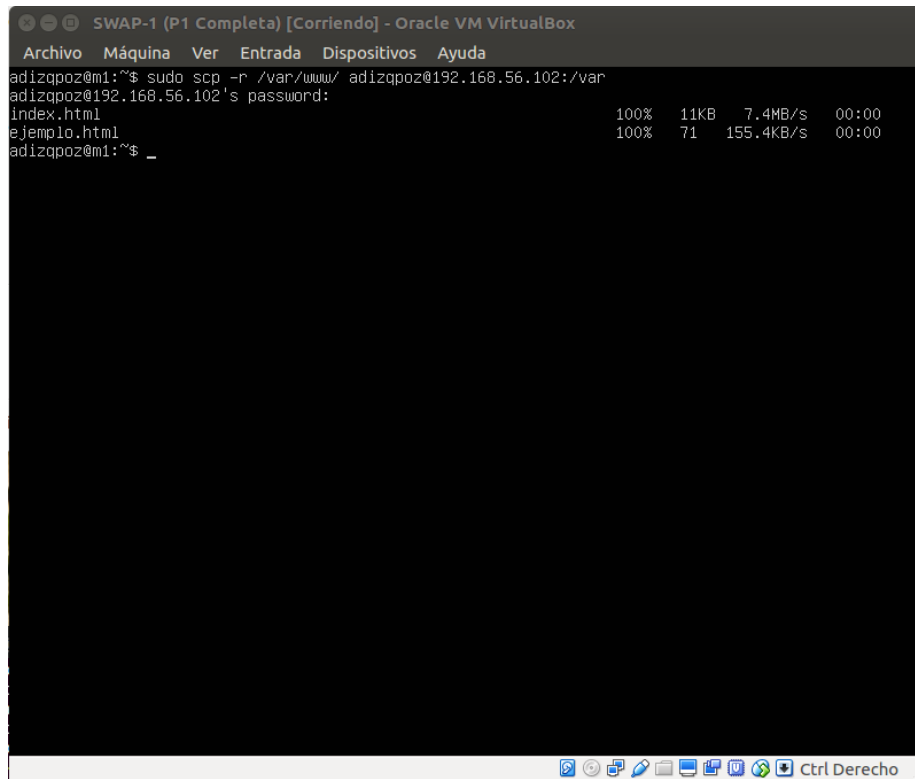
Como vemos, el usuario de la máquina M2 no es capaz de copiar la carpeta de la máquina 1 porque no tiene los permisos suficientes para llevar esa labor a cabo.

Se podría conectar con el usuario root de la otra máquina. Sin embargo eso sería muy peligroso, e incluso debería estar deshabilitado en la configuración de SSH. Sin embargo, esa labor no la abordaremos por el momento.

La otra opción, mucho más viable, sería otorgar la propiedad del directorio donde alojamos nuestro sitio web al usuario al cual podemos conectar por SSH. Esto se hace con la siguiente orden:

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

Una vez hecho esto, intentamos de nuevo ejecutar el comando anterior y obtenemos esto como resultado:

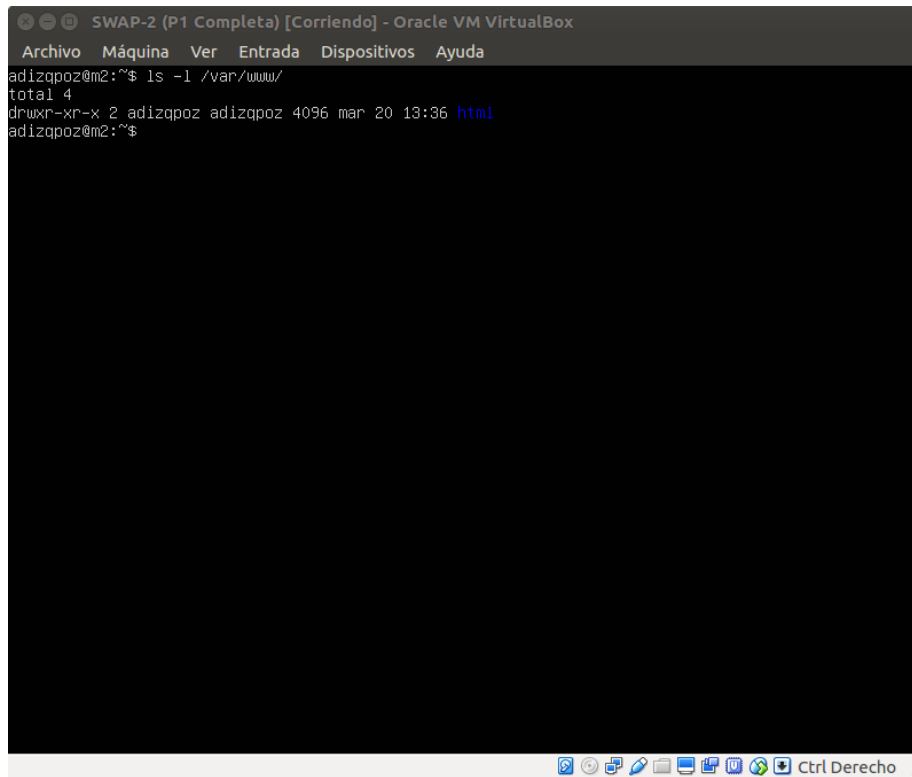


The screenshot shows a terminal window titled "SWAP-1 (P1 Completa) [Corriendo] - Oracle VM VirtualBox". The terminal displays the following commands and output:

```
adizqpoz@m1:~$ sudo scp -r /var/www/ adizqpoz@192.168.56.102:/var
adizqpoz@192.168.56.102's password:
index.html          100% 11KB  7.4MB/s  00:00
ejemplo.html        100%  71  155.4KB/s 00:00
adizqpoz@m1:~$ _
```

The terminal window has a menu bar with "Archivo", "Máquina", "Ver", "Entrada", "Dispositivos", and "Ayuda". The bottom status bar shows various icons and the text "Ctrl Derecho".

Observamos que la transfeencia se realiza con éxito. Sin embargo, es buena práctica verificar que nuestro objetivo se ha cumplido. Por ello, comprobamos en la máquina receptora guiándonos por la fecha de la última actualización.



```
SWAP-2 (P1 Completa) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
adizqpoz@m2:~$ ls -l /var/www/
total 4
drwxr-xr-x 2 adizqpoz adizqpoz 4096 mar 20 13:36 html
adizqpoz@m2:~$
```

Como suponíamos tras la anterior imagen, la copia de nuestra carpeta `/var/www/` ha sido un éxito.

2. Clonar directorios con `rsync`

A pesar de que aparentemente funcionan bien los dos métodos anteriormente utilizados, para grandes cantidades de información, y para filtrar qué archivos deseamos que se clonen y qué archivos no, una herramienta más óptima para ello es *rsync*.

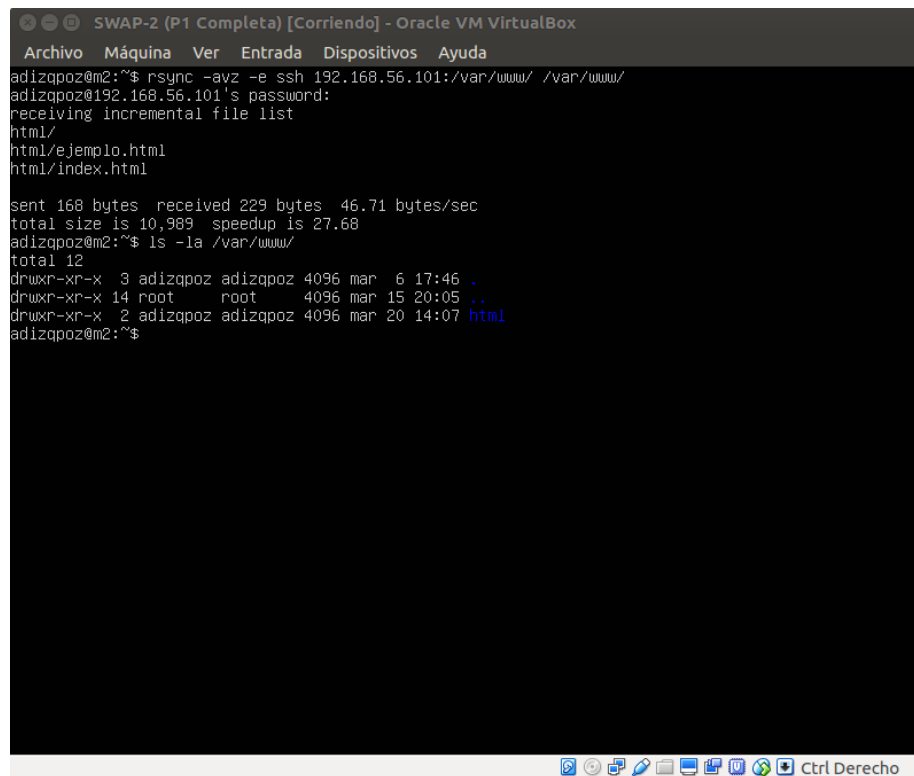
Se indica cómo instalar el programa en nuestros servidores. Sin embargo, ya lo tenemos instalado, así que esa labor ya no es necesaria.

También se nos indica cambiar el propietario del directorio principal del servidor web, lo cual hemos hecho en el paso anterior para poder realizar correctamente esas tareas, así que lo que nos queda en esta sección es ejecutar el comando necesario para ejecutar nuestra tarea, el cual es el siguiente:

```
rsync -avz -e ssh 192.168.56.101:/var/www/ /var/www/
```

Este comando realiza una transferencia que respeta fielmente la estructura del directorio fuente, incluyendo enlaces simbólicos, y es comprimido para realizar la

transferencia de forma más eficiente. Su resultado y comprobación es el siguiente:



```
SWAP-2 (P1 Completa) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
adizqpoz@m2:~$ rsync -avz -e ssh 192.168.56.101:/var/www/ /var/www/
adizqpoz@192.168.56.101's password:
receiving incremental file list
html/
html/ejemplo.html
html/index.html

sent 168 bytes  received 229 bytes  46.71 bytes/sec
total size is 10,989  speedup is 27.68
adizqpoz@m2:~$ ls -la /var/www/
total 12
drwxr-xr-x  3 adizqpoz adizqpoz 4096 mar  6 17:46 .
drwxr-xr-x 14 root      root    4096 mar 15 20:05 ..
drwxr-xr-x  2 adizqpoz adizqpoz 4096 mar 20 14:07 html
adizqpoz@m2:~$
```

Como observamos, todo se ha realizado correctamente.

Aunque no se haga una demostración de esto debido a la simplicidad de nuestro servidor web, merece la pena reseñar una funcionalidad de este programa que se apunta en el guión de prácticas de esta práctica.

Esta funcionalidad es la de omitir la copia de ciertos archivos o subdirectorios. Es útil para no realizar la copia de logs del servidor web, que son datos individuales de cada servidor, y que deben ser revisados por separado. El ejemplo de comando que se expone es el siguiente:

```
rsync -avz --delete --exclude=\\*/stats --exclude=\\*/error --exclude=\\*/files/pictures
```

Siempre que se desee alguna otra funcionalidad siempre se puede consultar el manual.

3. Configuración de SSH

En realidad, nuestra intención como administradores de este sistema no debería consistir en realizar una copia de un equipo a otro de forma puntual, sino que

un script sea capaz de realizar copias periódicas de un equipo a otro.

Para ello necesitamos que ambos equipos puedan tener acceso sin necesidad de introducir contraseña. La solución para ello pasa por habilitar un mecanismo de clave pública/privada, de forma que la clave pública sirva como contraseña para el equipo que debe acceder al dueño de la clave privada pueda hacerlo sin necesidad de una contraseña.

Para ello utilizamos el siguiente comando:

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

El método utilizado para el cifrado de las claves será el método RSA, ya que el RSA1 utiliza un protocolo de SSH anterior, y por tanto, menos seguro, y el método DSA está orientado principalmente a la firma electrónica de documentos, y no puede generar pares de claves de tamaño mayor a 1024 bits.

El resultado del uso de este comando es el siguiente:

```
adizqpoz@m2:~$ ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/adizqpoz/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/adizqpoz/.ssh/id_rsa.
Your public key has been saved in /home/adizqpoz/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Gmbkh6x7+0xhn70/WbrOKY6KluN31k69xm6Qr3k9u1k adizqpoz@m2
The key's randomart image is:
+---[RSA 4096]-----+
|
|  .
|  = .
|  B .
|  o +
|  . . S .
|  o   o. .
|  o +.. .=. E
|  =+.=.=0X000
|  +*0+=B@00++
+----[SHA256]-----+
```

Posteriormente, copiaremos la clave pública con un comando específico para esta labor, que simplifica significativamente la ejecución de la orden respecto a los métodos de copia de archivos anteriormente mostrados en este documento. El comando mencionado es el siguiente:

Al ejecutarlo, obtenemos esta salida:

```

adizqpoz@m2:~$ ssh-copy-id 192.168.56.101
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/adizqpoz/.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
adizqpoz@192.168.56.101's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh '192.168.56.101'"
and check to make sure that only the key(s) you wanted were added.

```

Y para comprobar que hemos hecho esta serie de acciones correctamente, intentaremos conectar con el servidor M1.

```

adizqpoz@m2:~$ ssh 192.168.56.101
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

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

System information as of Fri Mar 20 19:30:40 UTC 2020

System load:  0.0               Processes:    95
Usage of /:   36.1% of 9.78GB   Users logged in: 1
Memory usage: 63%              IP address for enp0s3: 10.0.2.15
Swap usage:   0%               IP address for enp0s8: 192.168.56.101

 * Latest Kubernetes 1.18 beta is now available for your laptop, NUC, cloud
   instance or Raspberry Pi, with automatic updates to the final GA release.

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

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

   https://multipass.run/

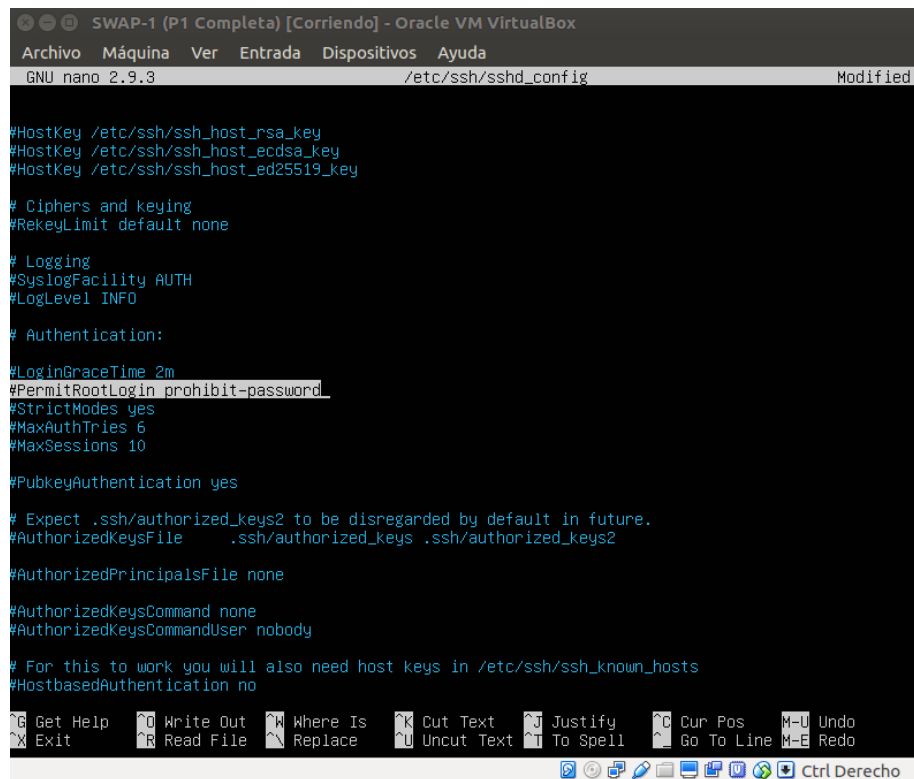
Pueden actualizarse 0 paquetes.
0 actualizaciones son de seguridad.

Last login: Fri Mar 20 11:31:42 2020

```

Como vemos, podemos acceder con éxito al servidor principal mediante SSH sin uso de contraseña. También se podría, de forma adicional, revisar la configuración de SSH para, por ejemplo, asegurarnos de que no se puede acceder al equipo identificándose como *root* o inhabilitar el acceso mediante contraseña para que únicamente puedan acceder los equipos con los que se tenga un acceso mediante clave pública/privada, por ejemplo.

Para ello accedemos al archivo */etc/ssh/sshd_config* y comprobamos los campos pertinentes a los aspectos que acabamos de señalar.



```
SWAP-1 (P1 Completa) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 2.9.3 /etc/ssh/sshd_config Modified

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

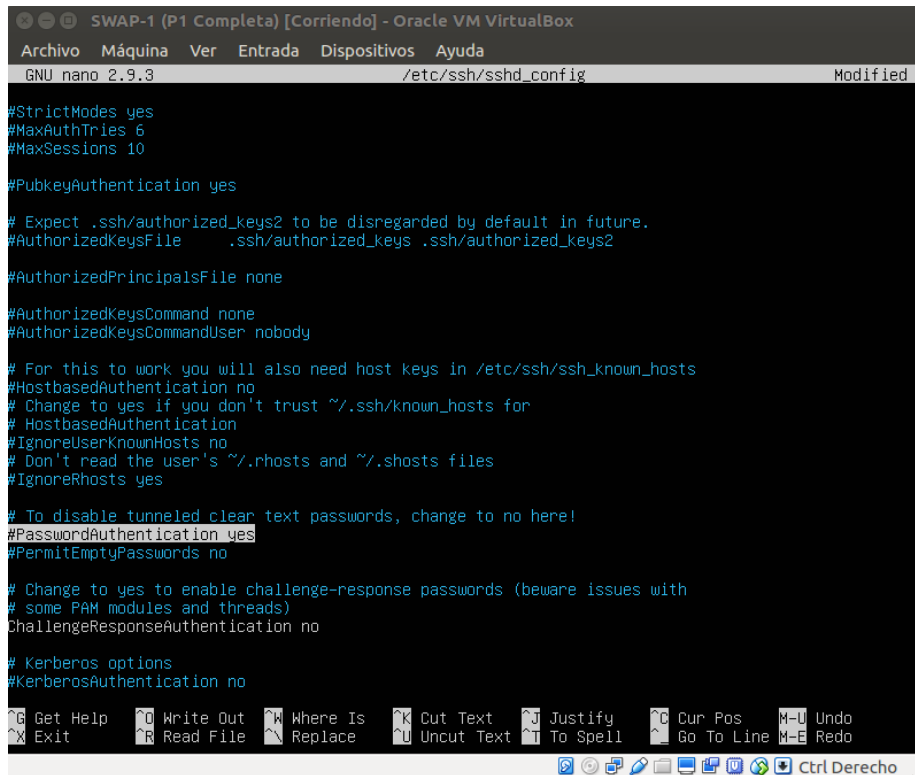
#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text   ^J Justify    ^C Cur Pos   M-U Undo
^X Exit      ^R Read File  ^N Replace   ^U Uncut Text ^T To Spell   ^G Go To Line M-E Redo
Ctrl Derecho
```

Como podemos observar, por defecto ya se impide el acceso al root mediante contraseña, pero sí que se podría mediante clave pública/privada. Para cambiar esto, debemos descomentar la línea y asignar a esa variable el valor "no".



```
SWAP-1 (P1 Completa) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 2.9.3 /etc/ssh/sshd_config Modified

#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# 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 yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Kerberos options
#KerberosAuthentication no

Get Help  Write Out  Where Is  Cut Text  Justify  Cur Pos  M-U Undo
Exit      Read File  Replace   Uncut Text  To Spell  Go To Line  M-B Redo
Ctrl Derecho
```

Por otro lado, como ya sabíamos, se puede acceder a M1 mediante contraseña, y esto también puede deshabilitarse de la misma manera que en la opción anterior. En caso de que necesitemos que obtenga M1 una nueva clave pública de otro servidor en el futuro, debemos volver a habilitar esa opción. Por ese motivo, por el momento, no vamos a cambiar esa opción. El momento adecuado sería cuando el sistema esté completo, a falta de las posibles mejoras. Cuando se necesite conectar alguna máquina nueva, volveremos a habilitar el acceso mediante contraseña, y una vez transferida la clave pública, se volvería a deshabilitar el acceso al servidor mediante contraseña.

Cada vez que realicemos un cambio en este archivo debemos reiniciar ssh mediante la siguiente orden:

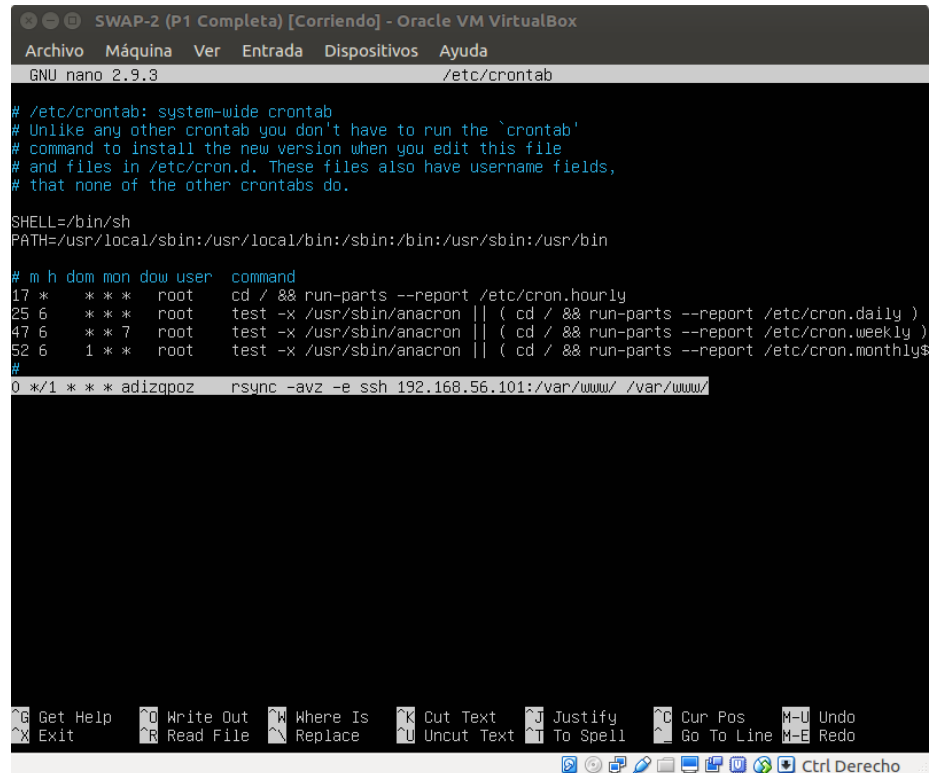
```
service sshd restart
```

4. Automatización de tareas

El motivo por el que habilitamos un sistema de claves pública/privada es porque el servidor puede encargarse por sí solo de mantener su información actualizada, y si hemos de introducir una contraseña cada vez que el sistema se actualice, la automatización de esta acción es baja.

Por ello, usaremos el demonio *cron*, cuyo cometido es lanzar procesos automáticamente, sea en un momento determinado, o periódicamente.

Para ello, debemos modificar el archivo `/etc/crontab/` agregando una nueva línea al mismo. Como lo que deseamos es que cada hora se ejecute el comando `rsync -avz 192.168.56.101:/var/www/ /var/www/`, para hacer una copia desde M1 hasta M2, insertaremos la siguiente línea:



```
SWAP-2 (P1 Completa) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 2.9.3 /etc/crontab

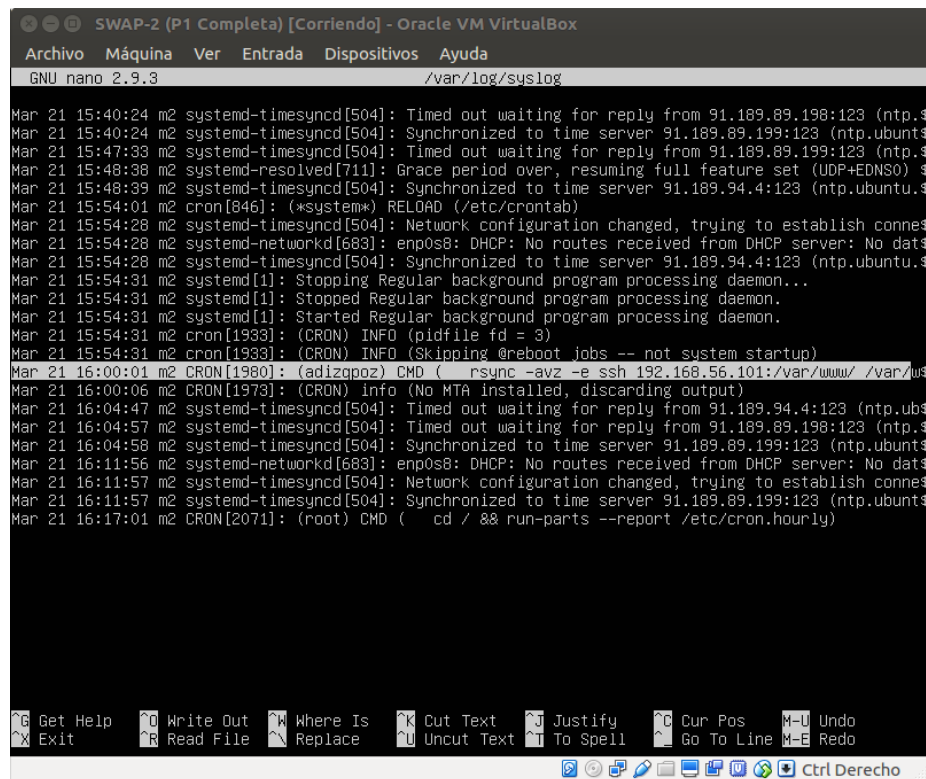
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

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 */1 * * * * adizqpoz rsync -avz -e ssh 192.168.56.101:/var/www/ /var/www/

Get Help  Write Out  Where Is  Cut Text  Justify  Cur Pos  M-U Undo
Exit      Read File  Replace   Uncut Text  To Spell  Go To Line  M-E Redo
Ctrl Derecho
```

Esperamos hasta que comience la siguiente hora y comprobamos que el comando se ha ejecutado al comenzar la hora buscando al final del archivo `/var/log/syslog`, que contiene los logs del sistema.

A screenshot of a terminal window titled "SWAP-2 (P1 Completa) [Corriendo] - Oracle VM VirtualBox". The window shows the GNU nano 2.9.3 editor editing the file /var/log/syslog. The log contains various system messages, including time synchronization, network configuration changes, and cron job executions. The bottom of the window features a menu bar with options like "Get Help", "Write Out", "Where Is", "Cut Text", "Justify", "Cur Pos", "Undo", "Exit", "Read File", "Replace", "Uncut Text", "To Spell", "Go To Line", "Redo", and a toolbar with icons for file operations and a "Ctrl Derecho" button.

```
SWAP-2 (P1 Completa) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 2.9.3 /var/log/syslog
Mar 21 15:40:24 m2 systemd-timesyncd[504]: Timed out waiting for reply from 91.189.89.198:123 (ntp.$
Mar 21 15:40:24 m2 systemd-timesyncd[504]: Synchronized to time server 91.189.89.199:123 (ntp.ubunt$
Mar 21 15:47:33 m2 systemd-timesyncd[504]: Timed out waiting for reply from 91.189.89.199:123 (ntp.$
Mar 21 15:48:38 m2 systemd-resolved[711]: Grace period over, resuming full feature set (UDP+EDNS0) $
Mar 21 15:48:39 m2 systemd-timesyncd[504]: Synchronized to time server 91.189.94.4:123 (ntp.ubuntu.$
Mar 21 15:54:01 m2 cron[846]: (*system*) RELOAD (/etc/crontab)
Mar 21 15:54:28 m2 systemd-timesyncd[504]: Network configuration changed, trying to establish connes
Mar 21 15:54:28 m2 systemd-networkd[683]: enp0s8: DHCP: No routes received from DHCP server: No dat$
Mar 21 15:54:28 m2 systemd-timesyncd[504]: Synchronized to time server 91.189.94.4:123 (ntp.ubuntu.$
Mar 21 15:54:31 m2 systemd[1]: Stopping Regular background program processing daemon...
Mar 21 15:54:31 m2 systemd[1]: Stopped Regular background program processing daemon.
Mar 21 15:54:31 m2 cron[1933]: (CRON) INFO (pidfile fd = 3)
Mar 21 15:54:31 m2 cron[1933]: (CRON) INFO (Skipping @reboot jobs -- not system startup)
Mar 21 16:00:01 m2 CRON[1980]: (adizgpoz) CMD ( rsync -avz -e ssh 192.168.56.101:/var/www/ /var/www$
Mar 21 16:00:06 m2 CRON[1973]: (CRON) info (No MTA installed, discarding output)
Mar 21 16:04:47 m2 systemd-timesyncd[504]: Timed out waiting for reply from 91.189.94.4:123 (ntp.ub$
Mar 21 16:04:57 m2 systemd-timesyncd[504]: Timed out waiting for reply from 91.189.89.198:123 (ntp.$
Mar 21 16:04:58 m2 systemd-timesyncd[504]: Synchronized to time server 91.189.89.199:123 (ntp.ubunt$
Mar 21 16:11:56 m2 systemd-networkd[683]: enp0s8: DHCP: No routes received from DHCP server: No dat$
Mar 21 16:11:57 m2 systemd-timesyncd[504]: Network configuration changed, trying to establish connes
Mar 21 16:11:57 m2 systemd-timesyncd[504]: Synchronized to time server 91.189.89.199:123 (ntp.ubunt$
Mar 21 16:17:01 m2 CRON[2071]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)

Get Help  Write Out  Where Is  Cut Text  Justify  Cur Pos  M-U Undo
Exit      Read File  Replace  Uncut Text  To Spell  Go To Line  M-E Redo

Ctrl Derecho
```

Como podemos observar, el comando se ha ejecutado correctamente. Con esto ya hemos cumplido con el objetivo de esta práctica, poder sincronizar periódicamente de manera automática el contenido del servidor web M2 con el de M1.

Autor: Adrián Izquierdo Pozo

Si desea ver el archivo Markdown puede verlo en mi repositorio de Github