
Servidores Web de Altas Prestaciones.

Práctica 2

Clonar la información de un Sitio Web.

Ricardo Ruiz Fernández de Alba

22/05/2023

Índice

Introducción	2
Tarea 1. Copiar archivos mediante SSH	2
Usando tubería PIPE	2
Usando SCP	3
Tarea 2. Clonar contenido entre máquinas	5
Configuraciones avanzadas de Rsync	7
Tarea 3. Configurar SSH para acceder sin contraseña	11
Tarea 4. Establecer una tarea con Cron	13
Utilizar Cron para automatizar Rsync	13
Demostración	13
Referencias	18

Introducción

En esta práctica, se abordarán varios aspectos relacionados con el uso de SSH y la automatización de tareas. Los objetivos principales son aprender a copiar archivos mediante SSH, clonar contenido entre máquinas, configurar SSH para acceder a máquinas remotas sin contraseña y establecer tareas periódicas utilizando cron.

Realizaremos las siguientes tareas básicas:

1. Probar la funcionalidad de copia de archivos a través de SSH.
2. Realizar el clonado de una carpeta entre las dos máquinas.
3. Configurar SSH para permitir el acceso sin solicitar contraseña.
4. Establecer una tarea programada en cron que se ejecute cada hora para mantener actualizado el contenido del directorio /var/www entre las dos máquinas.

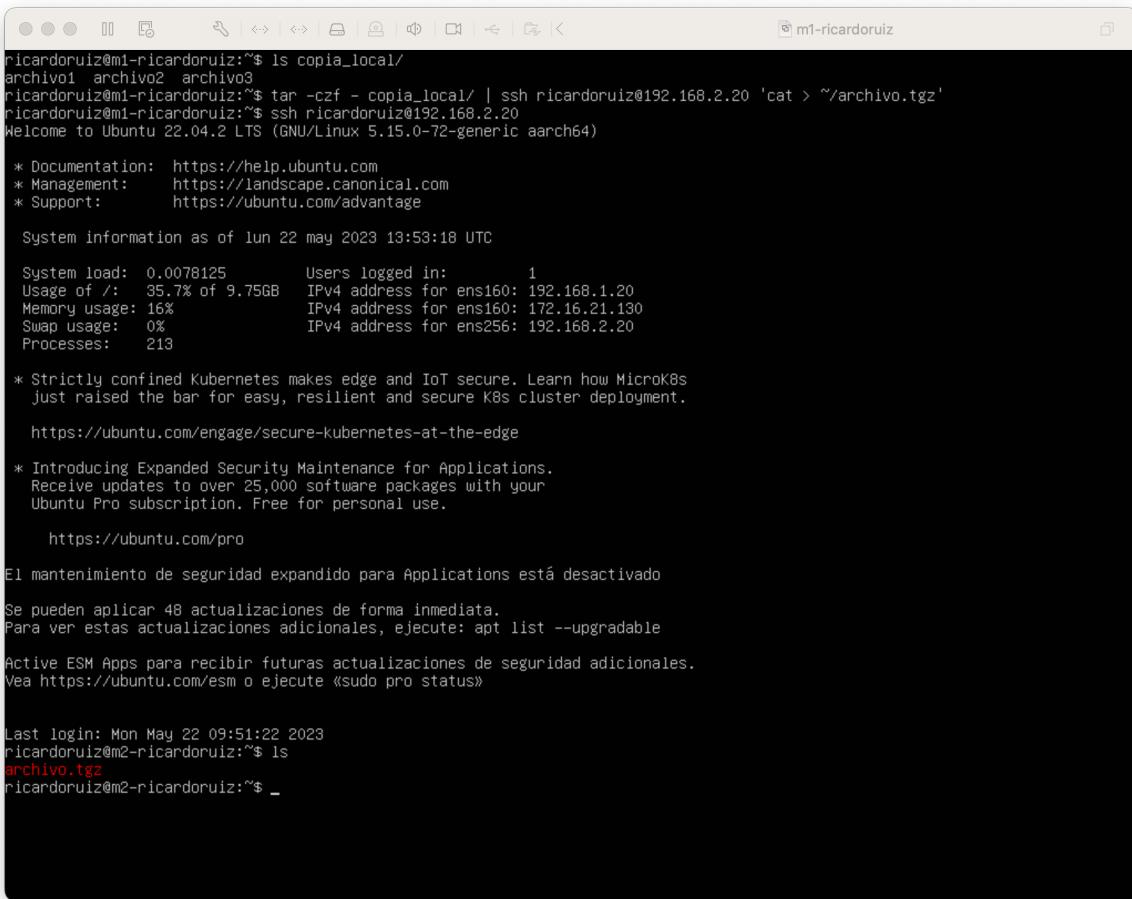
Tarea 1. Copiar archivos mediante SSH

Usando tubería PIPE

Supongamos que no tenemos suficiente espacio en disco local para crear un archivo tar.gz. Podemos crearlo directamente en el equipo remoto mediante SSH.:

```
1 ricardoruiz@m1-ricardoruiz $ tar -czf - copia_local/ | ssh  
ricardoruiz@192.167.2.20 'cat > ~/archivo.tgz'
```

De esta manera, se creará el archivo tar.tgz en el equipo remoto.



```
ricardoruiz@m1-ricardoruiz:~$ ls copia_local/
archivo1 archivo2 archivo3
ricardoruiz@m1-ricardoruiz:~$ tar -czf - copia_local/ | ssh ricardoruiz@192.168.2.20 'cat > ~/archivo.tgz'
ricardoruiz@m1-ricardoruiz:~$ ssh ricardoruiz@192.168.2.20
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-72-generic aarch64)

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

 System information as of lun 22 may 2023 13:53:18 UTC

 System load: 0.0078125    Users logged in: 1
 Usage of /: 35.7% of 9.75GB  IPv4 address for ens160: 192.168.1.20
 Memory usage: 16%          IPv4 address for ens160: 172.16.21.190
 Swap usage: 0%            IPv4 address for ens256: 192.168.2.20
 Processes: 213

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
 just raised the bar for easy, resilient and secure K8s cluster deployment.
 https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Introducing Expanded Security Maintenance for Applications.
 Receive updates to over 25,000 software packages with your
 Ubuntu Pro subscription. Free for personal use.

 https://ubuntu.com/pro

El mantenimiento de seguridad expandido para Applications está desactivado
Se pueden aplicar 48 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»
```

Last login: Mon May 22 09:51:22 2023
ricardoruiz@m2-ricardoruiz:~\$ ls
archivo.tgz
ricardoruiz@m2-ricardoruiz:~\$ _

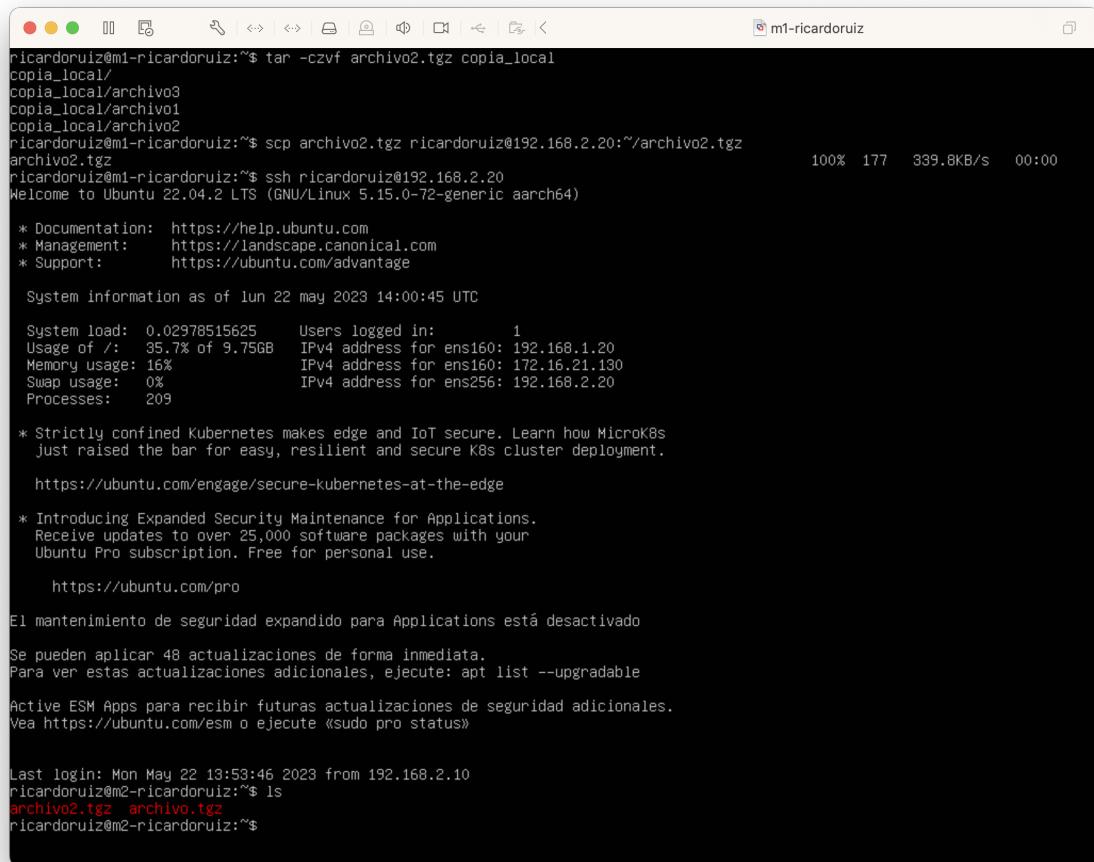
Figura 1: Copia mediante SSH

También es posible utilizar SCP, que utiliza SSH para realizar copias seguras y encriptadas de archivos o directorios. Podemos crear un archivo tar.gz localmente y luego copiarlo al equipo remoto utilizando SCP. Los comandos serían los siguientes:

Usando SCP

También podemos realizar esta tarea usando SCP que utiliza SSH para hacer copias seguras y encriptadas de archivos o directorios.

```
1 ricardoruiz@m1-ricardoruiz $ tar -czvf archivo2.tgz copia_local
2 ricardoruiz@m1-ricardoruiz $ scp archivo.tgz ricardoruiz@192
.168.2.20:~/archivo2.tgz
```



The terminal window shows the following sequence of commands and output:

```
ricardoruiz@m1-ricardoruiz:~$ tar -czvf archivo2.tgz copia_local
copia_local/
copia_local/archivo3
copia_local/archivo1
copia_local/archivo2
ricardoruiz@m1-ricardoruiz:~$ scp archivo2.tgz ricardoruiz@192.168.2.20:~/archivo2.tgz
ricardoruiz@m1-ricardoruiz:~$ ssh ricardoruiz@192.168.2.20
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-72-generic aarch64)

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

 System information as of lun 22 may 2023 14:00:45 UTC

 System load:  0.02978515625   Users logged in:          1
 Usage of /:   35.7% of 9.75GB   IPv4 address for ens160: 192.168.1.20
 Memory usage: 16%              IPv4 address for ens160: 172.16.21.130
 Swap usage:  0%                IPv4 address for ens256: 192.168.2.20
 Processes:   209

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
 just raised the bar for easy, resilient and secure K8s cluster deployment.
 https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Introducing Expanded Security Maintenance for Applications.
 Receive updates to over 25,000 software packages with your
 Ubuntu Pro subscription. Free for personal use.
 https://ubuntu.com/pro

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 48 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

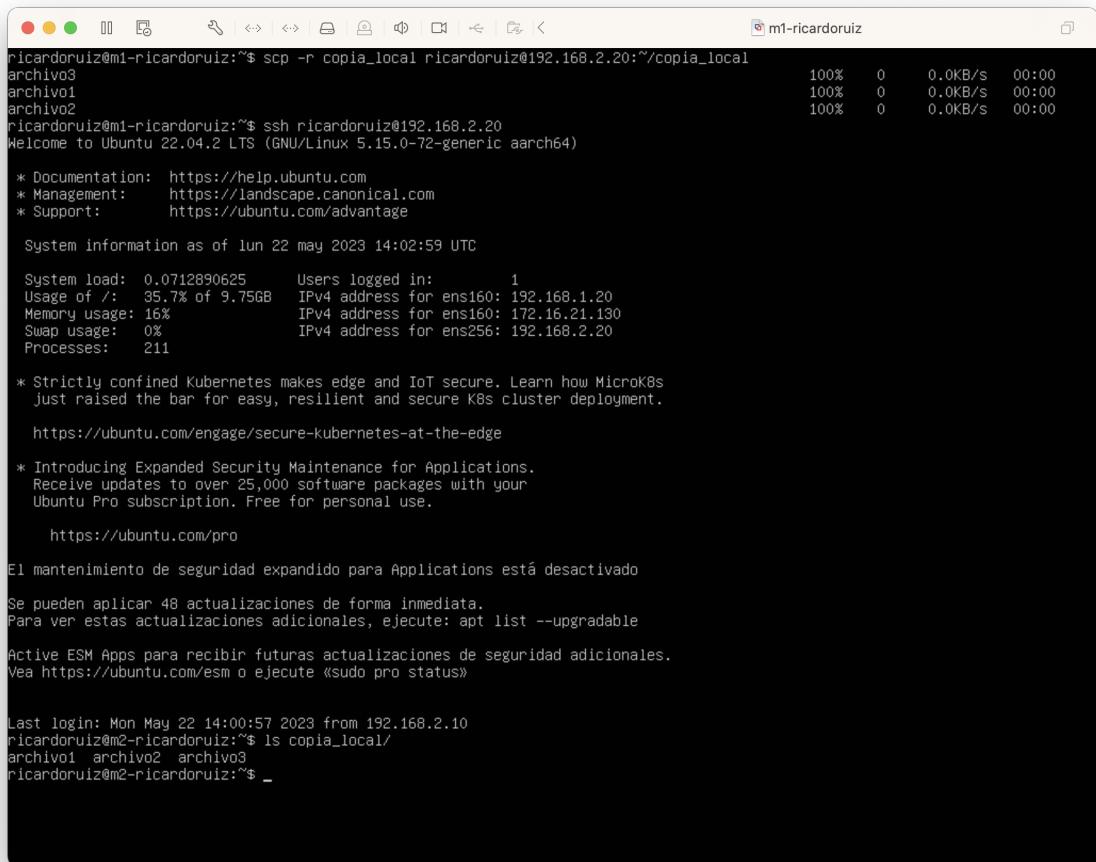
Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Last login: Mon May 22 13:53:46 2023 from 192.168.2.10
ricardoruiz@m2-ricardoruiz:~$ ls
archivo2.tgz  archivo.tgz
ricardoruiz@m2-ricardoruiz:~$
```

Figura 2: Copia mediante SCP

O enviar el propio directorio sin comprimir:

```
1 $ scp -r copia_local ricardoruiz@m2-ricardoruiz:~/copia_local
```



```
ricardoruiz@m1-ricardoruiz:~$ scp -r copia_local ricardoruiz@192.168.2.20:~/copia_local
archivo3                                         100%   0     0.0KB/s  00:00
archivo1                                         100%   0     0.0KB/s  00:00
archivo2                                         100%   0     0.0KB/s  00:00
ricardoruiz@m1-ricardoruiz:~$ ssh ricardoruiz@192.168.2.20
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-72-generic aarch64)

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

 System information as of lun 22 may 2023 14:02:59 UTC

 System load:  0.0712890625   Users logged in:          1
 Usage of /:  35.7% of 9.75GB   IPv4 address for ens160: 192.168.1.20
 Memory usage: 16%            IPv4 address for ens160: 172.16.21.130
 Swap usage:  0%              IPv4 address for ens256: 192.168.2.20
 Processes:   211

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
  just raised the bar for easy, resilient and secure K8s cluster deployment.
  https://ubuntu.com/engage/secure-kubernetes-at-the-edge

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.
  https://ubuntu.com/pro

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 48 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Last login: Mon May 22 14:00:57 2023 from 192.168.2.10
ricardoruiz@m2-ricardoruiz:~$ ls copia_local/
archivo1 archivo2 archivo3
ricardoruiz@m2-ricardoruiz:~$ _
```

Figura 3: Copia del directorio mediante SCP

Tarea 2. Clonar contenido entre máquinas

Para directorios de mayor tamaño, es mejor utilizar rsync. La herramienta rsync es una opción útil para realizar copias y sincronización de archivos. Lo instalamos:

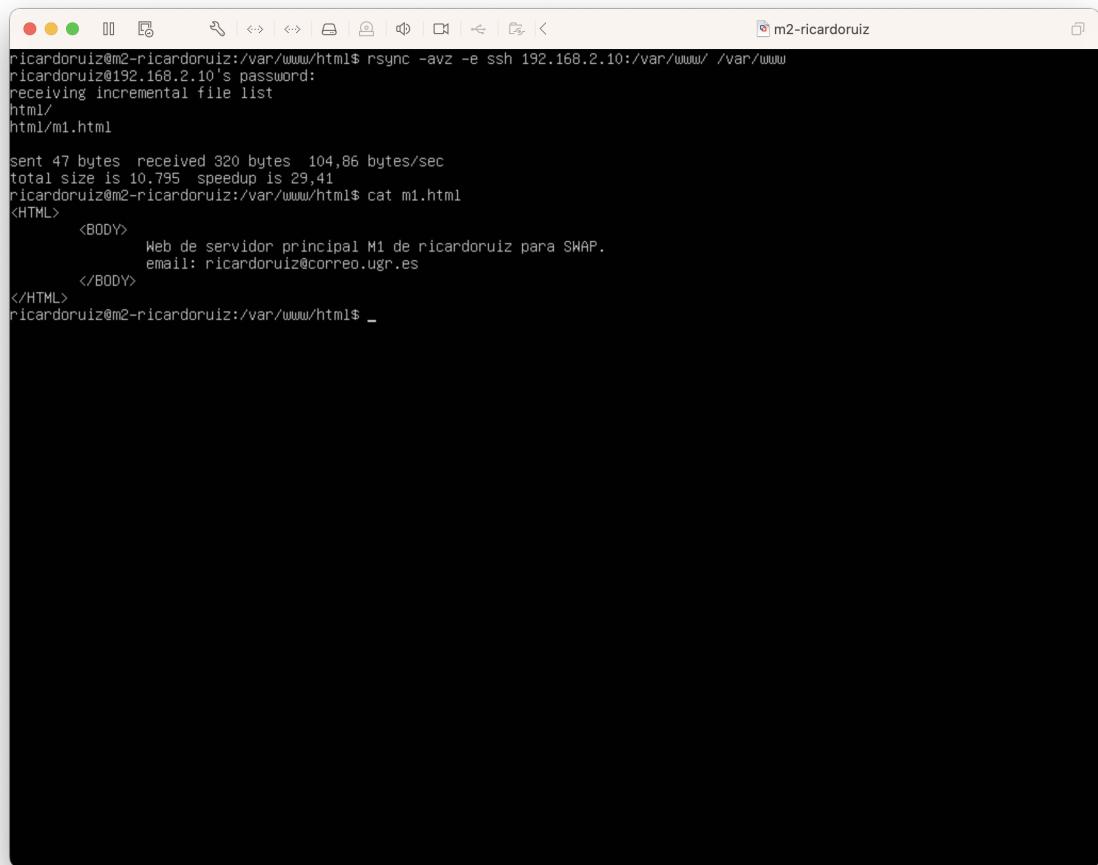
```
1 ricardoruiz@m2-ricardoruiz $ sudo apt-get install rsync
```

Queremos clonar la carpeta que contiene el servidor web principal desde la máquina secundario. Creamos un archivo `m1.html` en `/var/www/html` para distinguir el servidor principal.

Es necesario que el usuario sea propietario de la carpeta que queremos sincronizar.

```
1 ricardoruiz@m2-ricardoruiz $ sudo chown ricardoruiz:ricardoruiz -R /var/www
2 ricardoruiz@m2-ricardoruiz $ rsync -avz -e ssh 192.168.2.10:/var/www/ /var/www
```

Rsync nos pedirá la clave de usuario en M1 y comprobamos que se actualiza el contenido del directorio `/var/www/html` con el contenido del servidor principal (`m1.html`)



The screenshot shows a terminal window with a dark background and light-colored text. The window title is 'm2-ricardoruiz'. The terminal content is as follows:

```
ricardoruiz@m2-ricardoruiz:/var/www/html$ rsync -avz -e ssh 192.168.2.10:/var/www/ /var/www
ricardoruiz@192.168.2.10's password:
receiving incremental file list
html1/
html1/m1.html

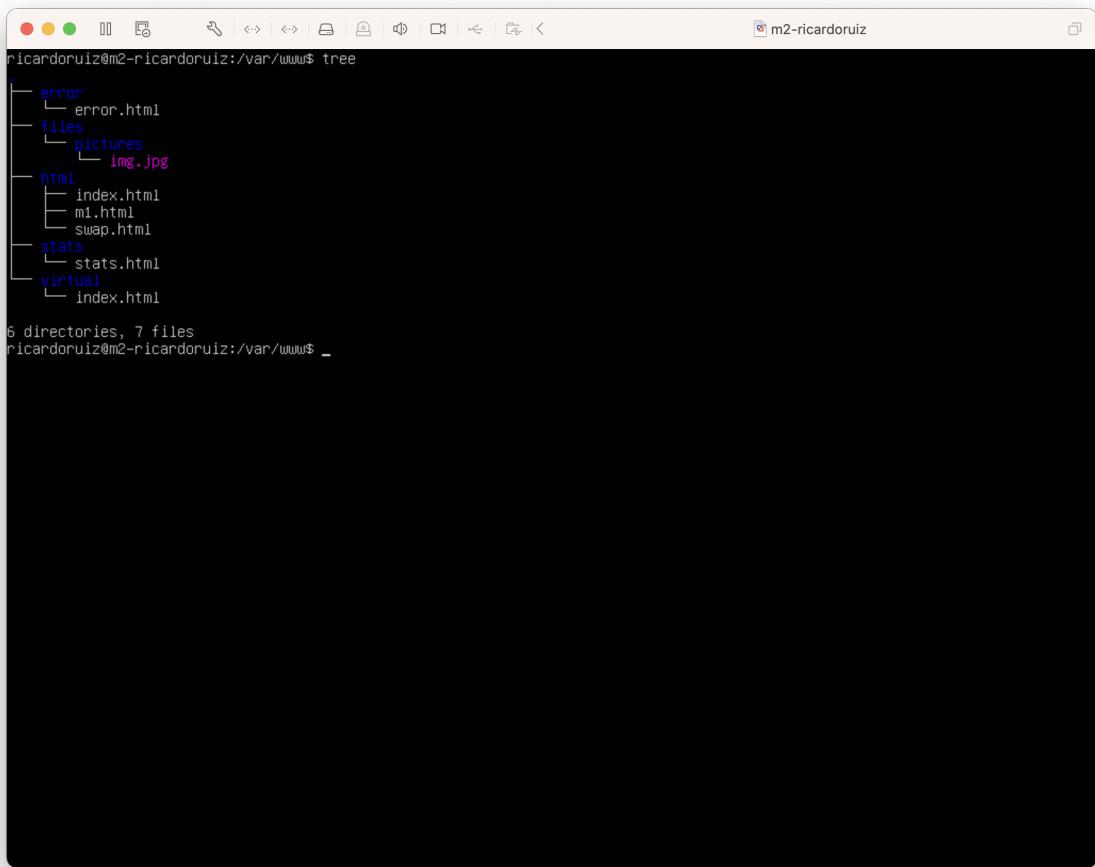
sent 47 bytes  received 320 bytes  104,86 bytes/sec
total size is 10,795  speedup is 29,41
ricardoruiz@m2-ricardoruiz:/var/www/html$ cat m1.html
<HTML>
    <BODY>
        Web de servidor principal M1 de ricardoruiz para SWAP.
        email: ricardoruiz@correo.ugr.es
    </BODY>
</HTML>
ricardoruiz@m2-ricardoruiz:/var/www/html$
```

Figura 4: Rsync

Configuraciones avanzadas de Rsync

Podemos especificar qué directorios ignorar durante el proceso de copia. Por ejemplo, si queremos realizar rcopia de `/var/www` pero excluir los directorios `/var/www/error`, `/var/www/stats` y `/var/www/files/pictures` usamos `--exclude`.

En primer lugar, creamos la estructura de directorios en el servidor secundario, con los archivos `error.html`, `stats.html` y añadimos `img.jpg` a `/var/www/files/pictures`.



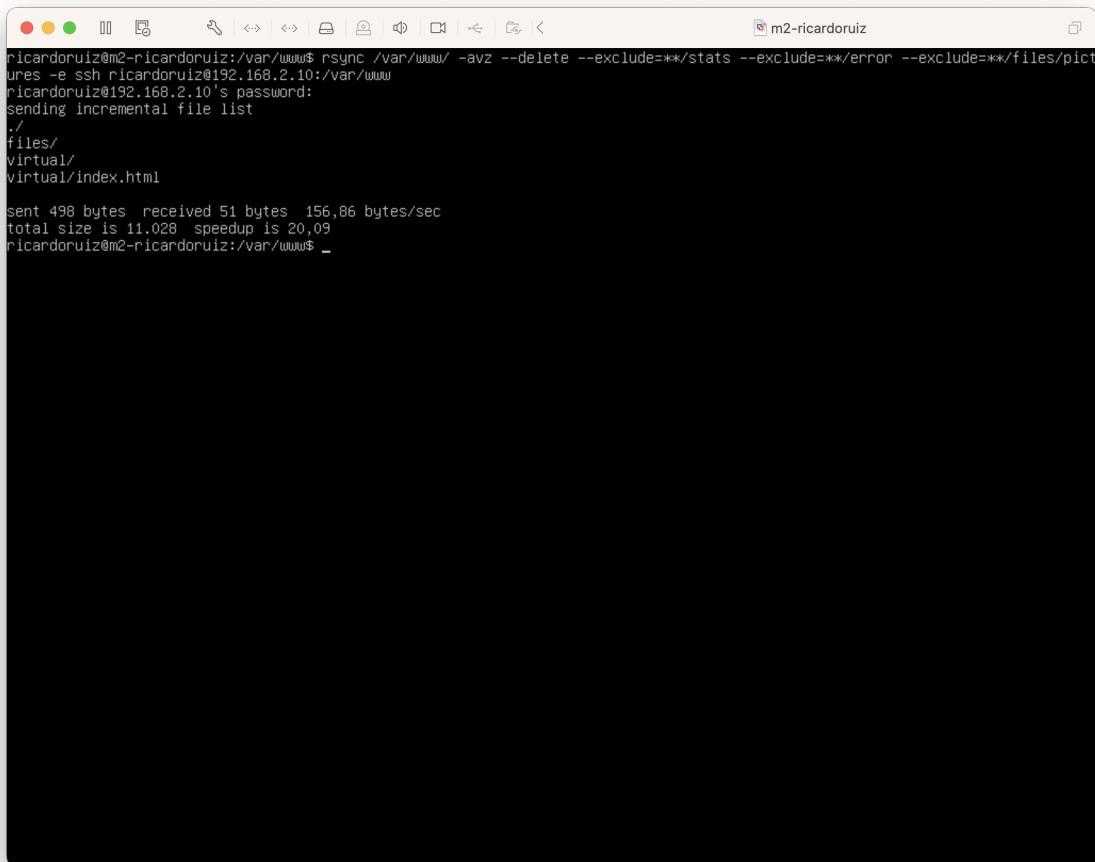
```
ricardoruiz@m2-ricardoruiz:/var/www$ tree
.
+-- error
|   '-- error.html
+-- files
|   '-- pictures
|       '-- img.jpg
+-- html
|   '-- index.html
|   '-- m1.html
|   '-- swap.html
+-- stats
|   '-- stats.html
+-- virtual
|   '-- index.html

6 directories, 7 files
ricardoruiz@m2-ricardoruiz:/var/www$
```

Figura 5: Estructura de directorios

```
1 ricardoruiz@m2-ricardoruiz $ rsync /var/www/ -avz --delete --exclude
=**/stats --exclude=**/error --exclude=**/files/pictures -e ssh
192.168.2.10:/var/www
```

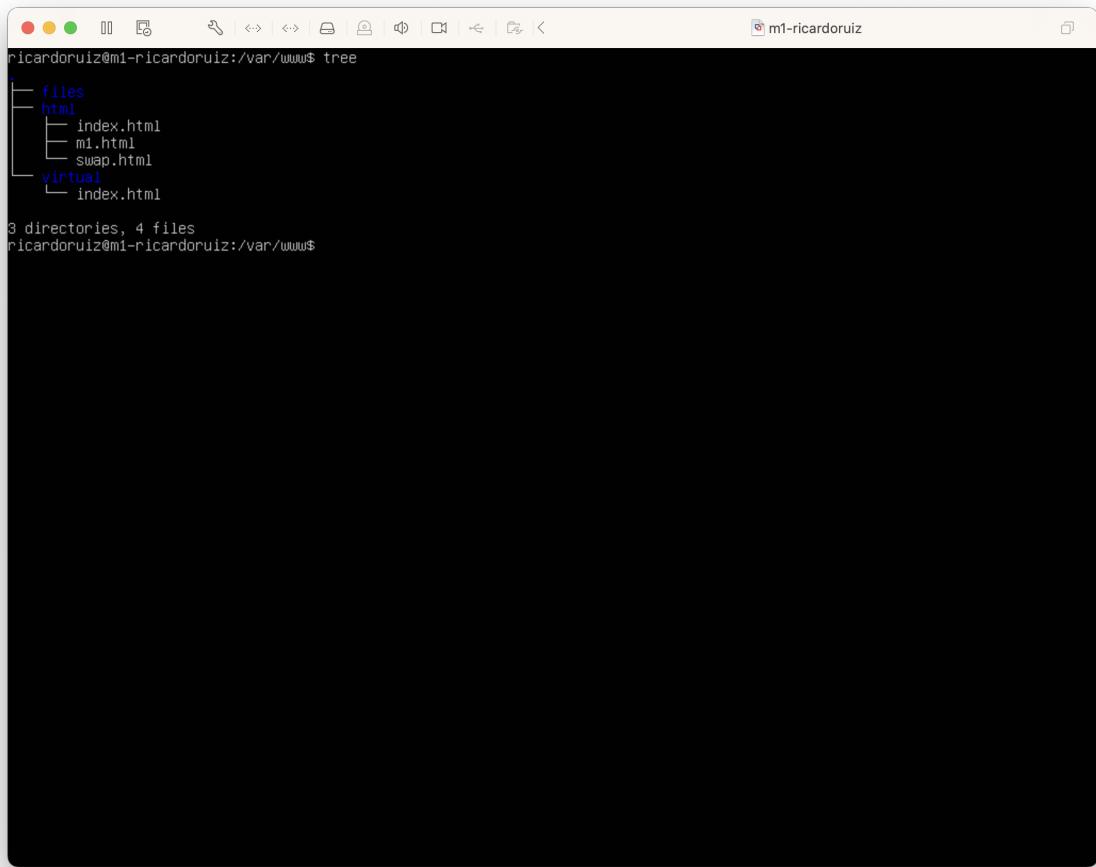
Así evitamos que los errores y estadísticas de la máquina 2 se sobreescriban con los de la principal.



```
ricardoruiz@m2-ricardoruiz:~$ rsync /var/www/ -avz --delete --exclude=**/stats --exclude=**/error --exclude=**/files/pictures -e ssh ricardoruiz@192.168.2.10:/var/www
ricardoruiz@192.168.2.10's password:
sending incremental file list
/
files/
virtual/
virtual/index.html

sent 498 bytes received 51 bytes 156,86 bytes/sec
total size is 11.028 speedup is 20,09
ricardoruiz@m2-ricardoruiz:~$ _
```

Figura 6: Opción exclude



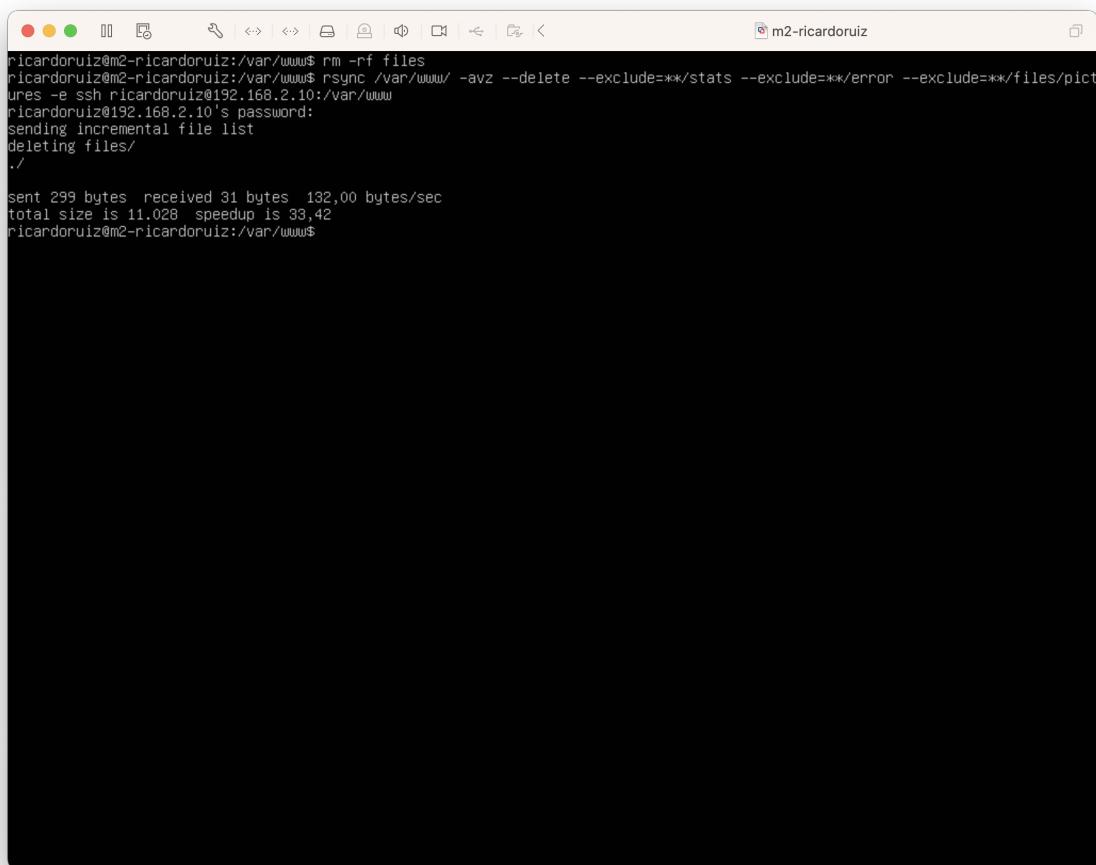
```
ricardoruez@m1-ricardoruez:/var/www$ tree
.
├── files
├── html
│   ├── index.html
│   ├── m1.html
│   └── swap.html
└── virtual
    └── index.html

3 directories, 4 files
ricardoruez@m1-ricardoruez:/var/www$
```

Figura 7: Opción exclude

La opción `--delete` indica que los archivos que se hayan eliminado en la máquina de origen también se eliminarán en la máquina de destino para asegurar una clonación idéntica.

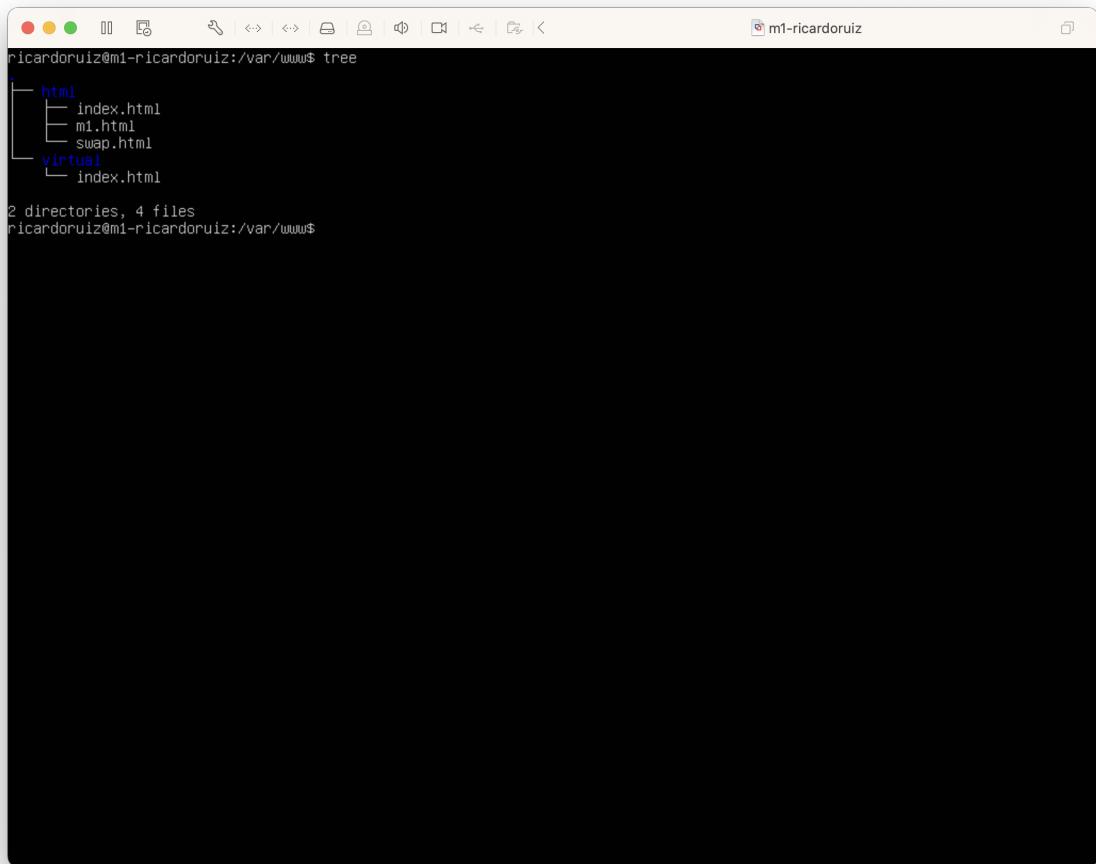
Comprobémoslo eliminando el directorio `files` y volviendo a ejecutar `rsync`:



```
ricardoruiz@m2-ricardoruiz:~/var/www$ rm -rf files
ricardoruiz@m2-ricardoruiz:~/var/www$ rsync -avz --delete --exclude=**/stats --exclude=**/error --exclude=**/files/pictures -e ssh ricardoruiz@192.168.2.10:~/var/www
ricardoruiz@192.168.2.10's password:
sending incremental file list
deleting files/
./

sent 299 bytes received 31 bytes 132,00 bytes/sec
total size is 11.028 speedup is 33,42
ricardoruiz@m2-ricardoruiz:~/var/www$
```

Figura 8: Opción delete



```
ricardoruez@m1-ricardoruez:/var/www$ tree
.
├── html
│   ├── index.html
│   ├── m1.html
│   └── swap.html
└── virtual
    └── index.html

2 directories, 4 files
ricardoruez@m1-ricardoruez:/var/www$
```

Figura 9: Opción delete

Tarea 3. Configurar SSH para acceder sin contraseña

Para lograr una actualización automática sin intervención del administrador, es necesario utilizar autenticación con claves pública-privada.

De forma parecida como se hicimos en la configuración avanzada de la primer práctica, ejecutamos esta vez en M2:

```
1 ricardoruez@m2-ricardoruez $ ssh-keygen -b 4096 -t rsa
```

Esto generará, por defecto, el fichero `~/.ssh/id_rsa` para la clave privada y el fichero `~/.ssh/id_rsa.pub` para la clave pública.

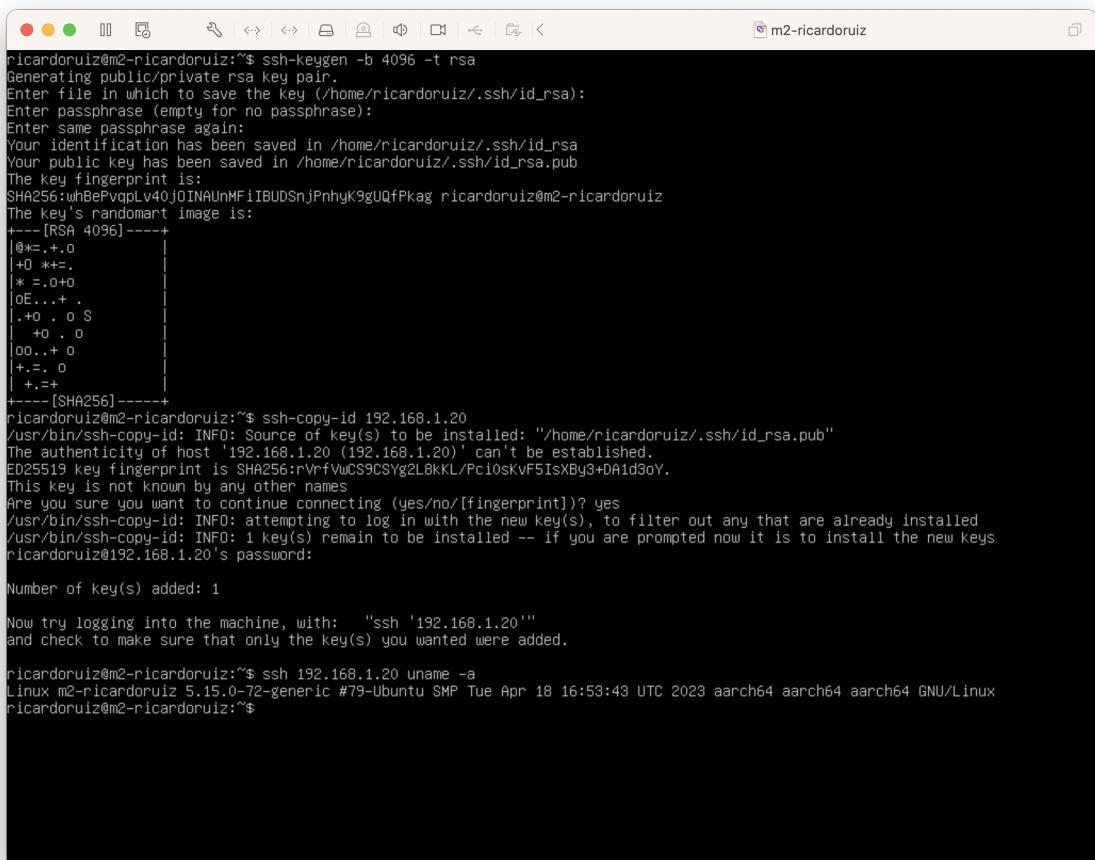
Este formato es válido para el protocolo 2 de SSH. Debemos copiar la clave pública al equipo remoto (máquina principal) en `~/.ssh/authorized_key` con permisos 60-s0.

Podemos realizarlo de forma sencilla utilizando el comando `ssh-copy-id`.

```
1 ricardorui@m2-ricardorui $ ssh-copy-id 192.168.2.10
```

Finalmente, podemos destacar la manera de ejecutar comandos en el equipo remoto, esta vez sin solicitud de contraseña.

```
1 ricardorui@m2-ricardorui $ ssh 192.168.1.20 uname -a
```



The screenshot shows a terminal window with the following session:

```
ricardorui@m2-ricardorui:~$ ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ricardorui/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ricardorui/.ssh/id_rsa
Your public key has been saved in /home/ricardorui/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:whBefVqpLv40j0INAUnMfiIBUDSnjPnhyK9gUQfPkag ricardorui@m2-ricardorui
The key's randomart image is:
+---[RSA 4096]---+
|@*+.o
|+0 **+.
|* =.o+o
|oE...+ .
|..o . o S
| +o . o
|oo..+ o
|+.z. o
| +.=+
+---[SHA256]---+
ricardorui@m2-ricardorui:~$ ssh-copy-id 192.168.1.20
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ricardorui/.ssh/id_rsa.pub"
The authenticity of host '192.168.1.20 (192.168.1.20)' can't be established.
ED25519 key fingerprint is SHA256:rVrfVuCS9CSYg2L8KKL/Pci0sKvF5IsXBy3+DA1d3oY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/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 the new keys
ricardorui@192.168.1.20's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh '192.168.1.20'"
and check to make sure that only the key(s) you wanted were added.

ricardorui@m2-ricardorui:~$ ssh 192.168.1.20 uname -a
Linux m2-ricardorui 5.15.0-72-generic #79-Ubuntu SMP Tue Apr 18 16:53:43 UTC 2023 aarch64 aarch64 aarch64 GNU/Linux
ricardorui@m2-ricardorui:~$
```

Podemos ahora automatizar la sincronización de directorios con rsync utilizando cron.

Tarea 4. Establecer una tarea con Cron

Cron es un administrador de procesos en segundo plano que ejecuta tareas programadas en momentos específicos. Está configurado mediante el archivo `/etc/crontab`, que contiene las tareas a ejecutar junto con su frecuencia y el usuario que las ejecuta.

Las tareas se definen utilizando siete campos en cada línea del archivo crontab, que representan el minuto, hora, día del mes, mes, día de la semana, usuario y comando a ejecutar.

Un asterisco representa la tarea se ejecutará en cada valor válido en esos campos.

Como ejemplo, la siguiente tarea apagará el ordenador cada día a las 00:30h:

```
1 30 0 * * * root /sbin/shutdown -h now
```

Utilizar Cron para automatizar Rsync

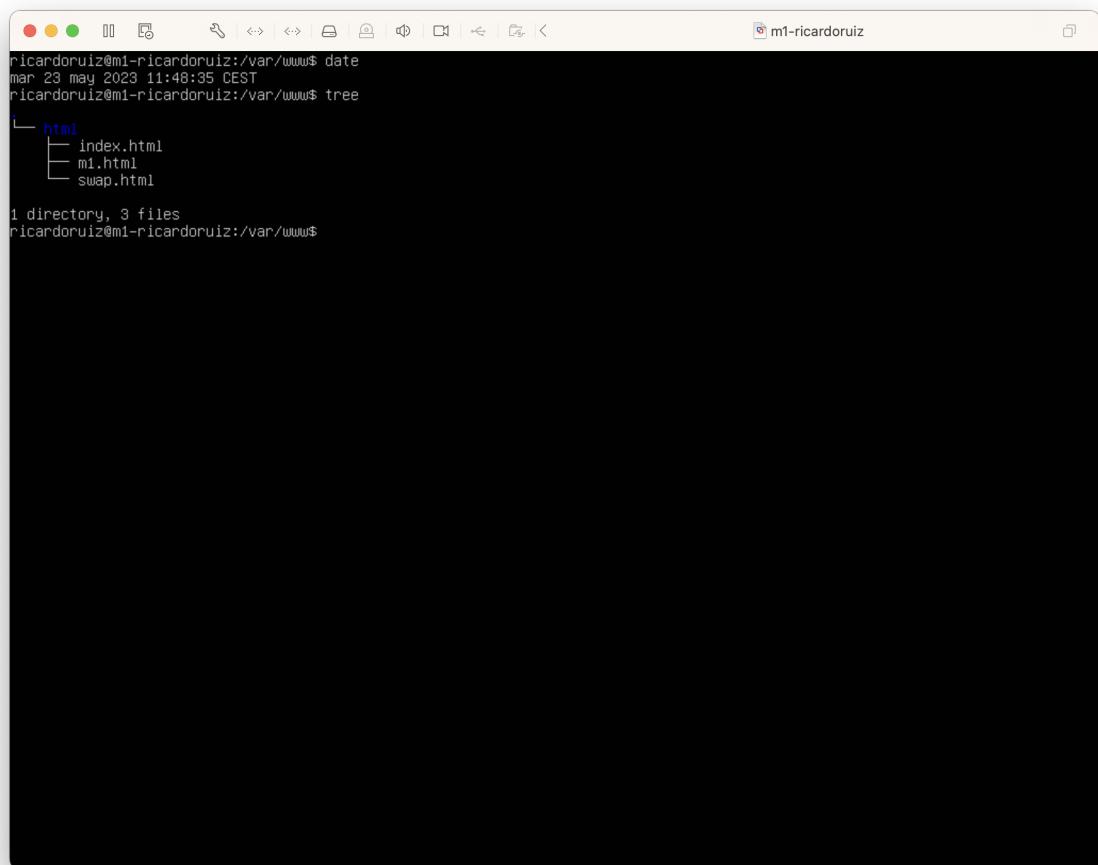
Creamos la siguiente tarea de cron `crontab -e`

```
1 0 * * * * rsync /var/www/ -avz --exclude=**/stats --exclude=**/error --  
exclude=**/files/pictures -e ssh 192.168.2.10:/var/www
```

La línea `0 * * * *` indica que la tarea se ejecutará en el minuto 0 de cada hora todos los días.

Demostración

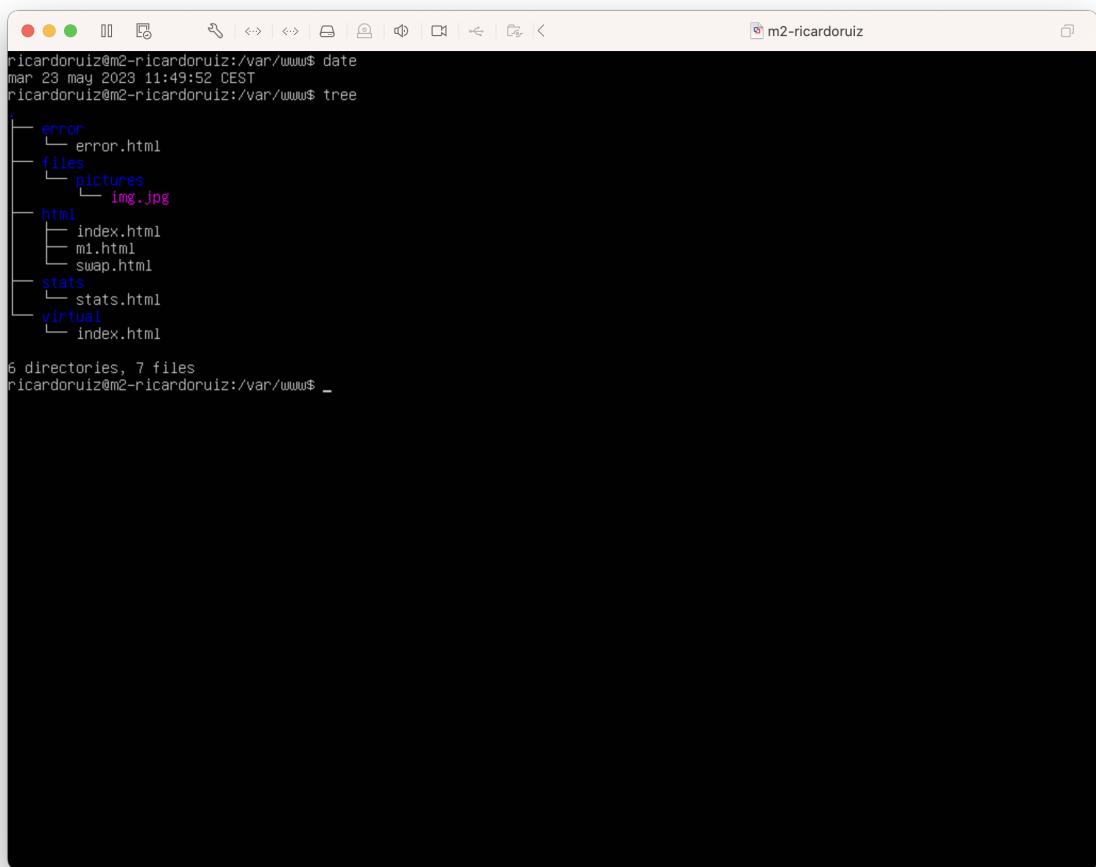
Los estados de las carpetas `/var/www` de las máquinas M2 y M1 respectivamente aproximadamente a las 11:48 / 11:49 son:



```
ricardoruiz@m1-ricardoruiz:/var/www$ date
mar 23 may 2023 11:48:35 CEST
ricardoruiz@m1-ricardoruiz:/var/www$ tree
└── html
    ├── index.html
    ├── m1.html
    └── swap.html

1 directory, 3 files
ricardoruiz@m1-ricardoruiz:/var/www$
```

Figura 10: Estado /var/www en M1



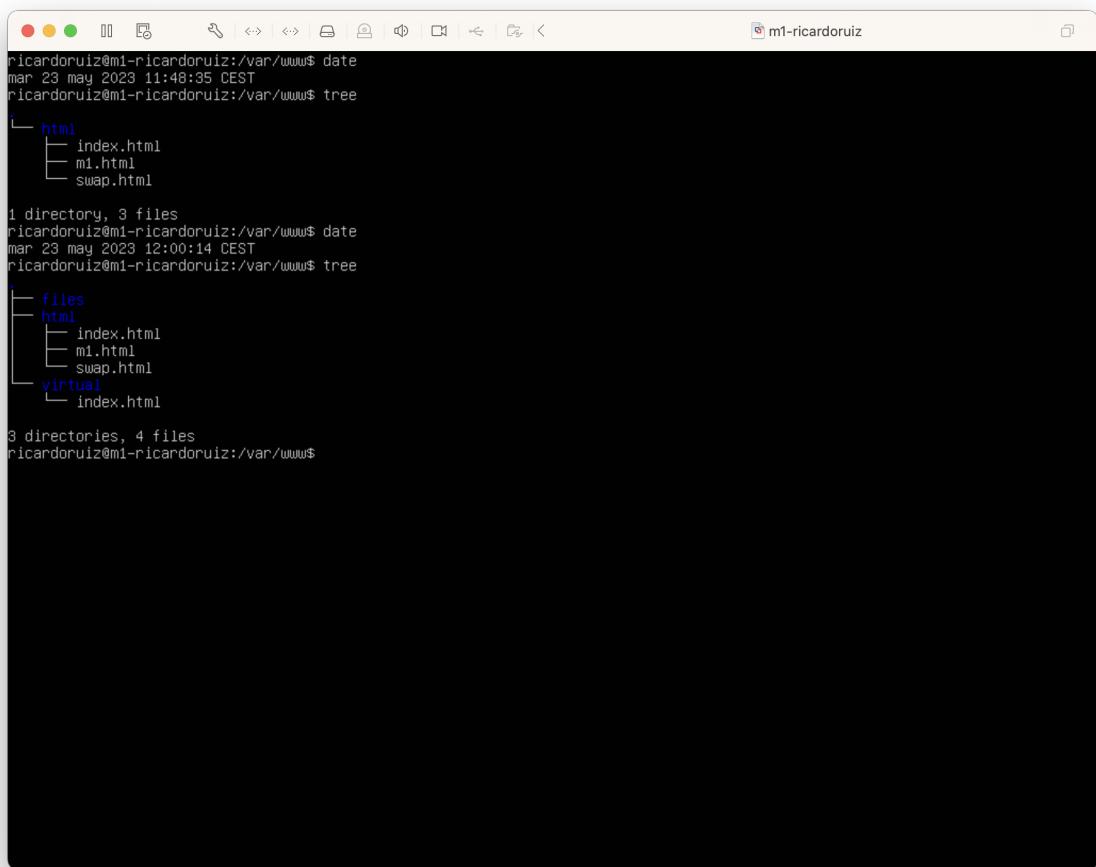
```
ricardoruiz@m2-ricardoruiz:/var/www$ date
mar 23 may 2023 11:49:52 CEST
ricardoruiz@m2-ricardoruiz:/var/www$ tree
.
+-- error
|   '-- error.html
+-- files
|   '-- pictures
|       '-- img.jpg
+-- html
|   '-- index.html
|   '-- m1.html
|   '-- swap.html
+-- stats
|   '-- stats.html
+-- virtual
|   '-- index.html
6 directories, 7 files
ricardoruiz@m2-ricardoruiz:/var/www$ _
```

Figura 11: Estado /var/www en M2

Configuramos la tarea de Crontab para que se ejecute cada hora:

]

Esperamos a las 12:00 y comprobamos que se ha sincronizado en M1:



```
ricardoruiz@mi-ricardoruiz:~$ date
mar 23 may 2023 11:48:35 CEST
ricardoruiz@mi-ricardoruiz:~$ tree
.
└── html
    ├── index.html
    ├── m1.html
    └── swap.html

1 directory, 3 files
ricardoruiz@mi-ricardoruiz:~$ date
mar 23 may 2023 12:00:14 CEST
ricardoruiz@mi-ricardoruiz:~$ tree
.
└── files
    └── html
        ├── index.html
        ├── m1.html
        └── swap.html
    └── virtual
        └── index.html

3 directories, 4 files
ricardoruiz@mi-ricardoruiz:~$
```

Figura 12: Sincronización automática

Referencias

Aquí tienes la lista de referencias en el formato solicitado:

- **OpenSSH.** Recuperado de <https://www.openssh.com/>
- **SCP (Secure Copy).** Recuperado de <https://man.openbsd.org/scp>
- **Rsync.** Recuperado de <https://rsync.samba.org/>
- **SSH Keygen.** Recuperado de <https://man.openbsd.org/ssh-keygen>
- **Cron.** Recuperado de <https://man7.org/linux/man-pages/man8/cron.8.html>