

SERVIDORES WEB DE ALTAS PRESTACIONES (2016-2017)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Memoria Práctica 2

Guillermo Montes Martos

21 de marzo de 2017

Índice

0. Cuestión 0	3
0.1. Configurar los servidores para que se puedan comunicar entre ellos.	3
1. Cuestión 1	4
1.1. Probar el funcionamiento de de la copia de archivos por ssh	4
2. Cuestión 2	5
2.1. Realizar el clonado de una carpeta entre las dos máquinas virtuales. . . .	5
3. Cuestión 3	6
3.1. Configurar el servidor ssh para acceder sin que se solicite contraseña. . . .	6
4. Cuestión 4	8
4.1. Establecer una tarea en cron que se ejecute cada hora para mantener actualizado el contenido del directorio /var/www entre las dos máquinas. .	8

Índice de figuras

0.1. Activando el segundo adaptador red en VirtualBox.	3
0.2. Información del servicio systemd.	4
1.1. Uso del comando scp.	4
1.2. Creando tar en máquinas remotas.	5
2.1. Clonando carpetas en máquinas remotas.	6
3.1. Creando el par de claves pública/privada.	7
3.2. Copiando la clave pública en la máquina 2.	7
4.1. Haciendo uso de crontab.	8

0. Cuestión 0

0.1. Configurar los servidores para que se puedan comunicar entre ellos.

Para esta tarea se ha usado la configuración aprendida en la asignatura de Ingeniería de Servidores para conectar dos máquinas virtuales.

Esta se basa en, una vez instaladas las máquinas virtuales de manera corriente, activar desde la configuración de VirtualBox un segundo adaptador red en modo solo-anfitrión [1] y, posteriormente, cada vez que lancemos las máquinas, activar la interfaz de red que nos permitirá tener acceso entre ellas.

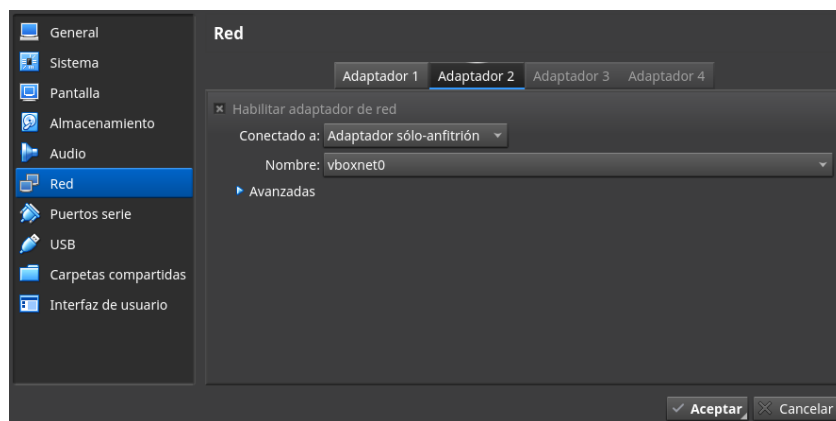


Figura 0.1: Activando el segundo adaptador red en VirtualBox.

Para activar la interfaz, se ejecutará en cada inicio un sencillo script (al cual hemos llamado `init_network`) para el cual serán necesarios permisos de administrador.

Listing 1: `init_network`

```
ifup enp0s8
ifconfig up enp0s8
dhclient
```

Para automatizar el proceso, se creó un servicio `systemd` con el mismo nombre que ejecuta el script cada vez que se inicia la máquina virtual bajo el usuario `root`.

```

gmm@UbuntuSWAP1:~$ systemctl status init_network.service
■ init_network.service - Inicializar la red que comunica con otras VM
   Loaded: loaded (/etc/systemd/system/init_network.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since mar 2017-03-21 12:41:25 CET; 1h 10min ago
     Process: 829 ExecStart=/bin/sh /home/gmm/.init_network (code=exited, status=0/SUCCESS)
    Main PID: 829 (code=exited, status=0/SUCCESS)

mar 21 12:41:23 UbuntuSWAP1 systemd[1]: Started Inicializar la red que comunica con otras VM.
mar 21 12:41:23 UbuntuSWAP1 sh[829]: Unknown interface enp0s8
mar 21 12:41:23 UbuntuSWAP1 sh[829]: enp0s8: Nombre de «host» no encontrado
mar 21 12:41:23 UbuntuSWAP1 sh[829]: ifconfig: `--help' le da información de como usar la orden.
mar 21 12:41:24 UbuntuSWAP1 dhclient[855]: DHCPREQUEST of 192.168.56.101 on enp0s8 to 255.255.255.255
mar 21 12:41:24 UbuntuSWAP1 dhclient[855]: DHCPREQUEST of 10.0.2.15 on enp0s3 to 255.255.255.255 por
mar 21 12:41:24 UbuntuSWAP1 dhclient[855]: DHCPACK of 192.168.56.101 from 192.168.56.100
mar 21 12:41:25 UbuntuSWAP1 dhclient[855]: bound to 192.168.56.101 -- renewal in 535 seconds.
mar 21 12:41:25 UbuntuSWAP1 dhclient[955]: DHCPACK of 10.0.2.15 from 10.0.2.2
lines 1-15/15 (END)

```

Figura 0.2: Información del servicio systemd.

1. Cuestión 1

1.1. Probar el funcionamiento de de la copia de archivos por ssh

Para probar su funcionamiento, se han realizado dos pruebas distintas.

Una primera se ha realizado mediante el comando scp, el cual permite copiar archivos mediante ssh [2]. Para ello, se crearon los archivos *prueba1* en la máquina 1 y *prueba2* en la máquina 2, con contenidos distintos. A continuación, copiamos el archivo *prueba1* de la máquina 1 en la máquina 2 haciendo uso de scp y comprobamos que el envío ha sido correcto.

The image shows two terminal windows side-by-side. The left window is titled 'UbuntuSWAP1 [Corriendo] - Oracle VM VirtualBox' and shows a user 'gmm' at 'UbuntuSWAP1' creating a file 'prueba1' and then using 'scp' to copy it to 'gmm@192.168.56.102'. The right window is titled 'UbuntuSWAP2 [Corriendo] - Oracle VM VirtualBox' and shows the same user 'gmm' at 'UbuntuSWAP2' receiving the file 'prueba1' via scp. Both windows show network statistics at the bottom, indicating a successful transfer of 11.8 KB.

Figura 1.1: Uso del comando scp.

Una segunda forma ha sido la manera descrita en el guión de la práctica sobre como

```
Archivo Máquina Ver Entrada Dispositivos Ayuda
gmn@h8:~$ Link encap:Ethernet direcciónHW 08:00:27:00:aa:49
Direc.: Dirección:192.168.56.101 Difus.:192.168.56.255 Másc:255.255.255.0
Dirección inet6: ::1:128 Alcance:anfitrión
Activación de función: RTU=65536 Métrica:1
Paquetes RX:168 errores:0 perdidos:0 overruns:0 frame:0
Paquetes TX:65 errores:0 perdidos:0 overruns:0 carrier:0
colisiones:0 long.cola:TX:1000
Link RX:24185 (24.1 KB) TX bytes:20643 (20.6 KB)

gmn@h8:~$ Link encap:Bucle local
Direc.: Dirección:127.0.0.1 Másc:255.0.0.0
Dirección inet6: ::1:128 Alcance:anfitrión
Activación de función: RTU=65536 Métrica:1
Paquetes RX:164 errores:0 perdidos:0 overruns:0 frame:0
Paquetes TX:164 errores:0 perdidos:0 overruns:0 carrier:0
colisiones:0 long.cola:TX:1
Link RX:12104 (12.1 KB) TX bytes:12104 (12.1 KB)

gmn@h8:~$ gnm@h8:~$ $ ll
total 40
drwxr-xr-x 4 gnm gnm 4096 nov 21 13:32 /
drwxr-xr-x 3 root root 4096 nov 7 19:01 /bin
drwxr-xr-x 1 root root 32768 nov 21 12:41 /bash_history
-rw-r--r-- 1 gnm gnm 220 nov 7 19:01 /bash_logout
-rw-r--r-- 1 gnm gnm 3771 nov 7 19:01 /bashrc
-rw-r--r-- 2 gnm gnm 4096 nov 7 19:02 /cable
-rw-r--r-- 1 root root 40 nov 21 12:40 /init_network
-rw-r--r-- 1 gnm gnm 655 nov 7 19:01 /profile
drwxr-xr-x 2 gnm gnm 4096 nov 21 12:51 /ssh
-rw-r--r-- 1 gnm gnm 0 nov 7 19:02 /sudo_as_admin_successful
-rw-r--r-- 1 gnm gnm 186 nov 21 13:32 /sudo_log
-rw-r--r-- 1 root root 4461 nov 21 12:40 /vminfo

gmn@h8:~$ gnm@h8:~$ $ tar xzf tar.gz
home-gnm-preub2
gmn@h8:~$ gnm@h8:~$ $

Archivo Máquina Ver Entrada Dispositivos Ayuda
gmn@h8:~$ gnm@h8:~$ $ ll
total 48
drwxr-xr-x 4 gnm gnm 4096 nov 21 13:32 /
drwxr-xr-x 3 root root 4096 nov 7 19:01 /bin
-rw-r--r-- 1 gnm gnm 432 nov 21 12:42 /bash_history
-rw-r--r-- 1 gnm gnm 220 nov 19 15:44 /bash_logout
-rw-r--r-- 1 gnm gnm 3771 nov 19 15:44 /bashrc
drwxr-xr-x 2 gnm gnm 4096 nov 19 15:47 /cable
-rw-r--r-- 1 root root 40 nov 21 12:40 /init_network
-rw-r--r-- 1 gnm gnm 655 nov 19 15:44 /profile
-rw-r--r-- 1 gnm gnm 29 nov 21 13:32 /preub1
-rw-r--r-- 1 gnm gnm 29 nov 21 13:25 /preub2
drwxr-xr-x 2 gnm gnm 4096 nov 21 13:32 /ssh
-rw-r--r-- 1 gnm gnm 0 nov 19 15:40 /sudo_as_admin_successful
-rw-r--r-- 1 root root 1498 nov 21 12:40 /vminfo

gmn@h8:~$ gnm@h8:~$ $
```

2. Cuestión 2

Para realizar dicho clonado, se usará la herramienta especificada en el guión de prácticas, rsync [4]. Para darnos cuenta de su realización, se ha creado un fichero html sencillo llamado *hola.html* en el directorio `/var/www/html/` de la máquina 1, el cual será clonado en la máquina 2.

5

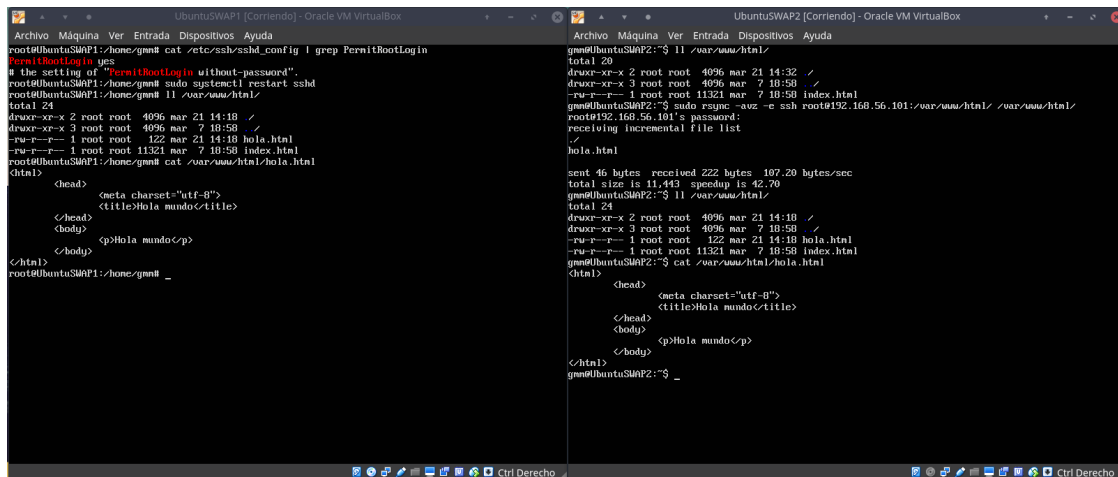


Figura 2.1: Clonando carpetas en máquinas remotas.

3. Cuestión 3

3.1. Configurar el servidor ssh para acceder sin que se solicite contraseña.

Para la realización de esta tarea, suponemos que nos encontramos en la máquina 1 y que deseamos acceder mediante ssh sin contraseña al usuario root de la máquina 2.

El primer paso consistiría en crear el par de claves pública/privada en la máquina 1 introduciendo el comando `ssh-keygen -t rsa -b 4096` [5]. Mediante el parámetro `-t` indicamos el tipo de clave y con `-b` la longitud en bytes de esta. Es importante dejar el campo *passphrase* vacío, ya que es lo que nos permitirá acceder sin contraseña.

```

gmm@UbuntuSWAP1:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/gmm/.ssh/id_rsa):
/home/gmm/.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/gmm/.ssh/id_rsa.
Your public key has been saved in /home/gmm/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:n9iouYyJRZuk0NZxHfDRH9d5h8/0pi9zaz4bo20YtIs gmm@UbuntuSWAP1
The key's randomart image is:
+---[RSA 4096]-----+
|      ..o. .o|
|      o  ooo*|
|      . . o  =*|
|      . . o  . *|
|      . o $   . o |
|      o + * . o. |
|      . = + . ++ |
|      o + E o+=+|
|      . o =o..+0+|
+---[SHA256]-----+

```

Figura 3.1: Creando el par de claves pública/privada.

El segundo paso sería copiar la clave pública de la máquina 1 en la máquina 2. Para ello, hacemos uso del comando `ssh-copy-id -i ~/.ssh/id_rsa.pub root@<máquina2>` [6]. El parámetro `-i` sirve para indicar donde se encuentra el archivo con la clave pública y la cadena `<máquina2>` tendremos que sustituirla por la dirección IP de la máquina 2. Hecho esto, ya podemos acceder mediante ssh sin contraseña a la segunda máquina.

```

gmm@UbuntuSWAP1:~$ ssh-copy-id -i ~/.ssh/id_rsa.pub root@192.168.56.102
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/gmm/.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 alr
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
root@192.168.56.102's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@192.168.56.102'"
and check to make sure that only the key(s) you wanted were added.

gmm@UbuntuSWAP1:~$ ssh root@192.168.56.102
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-66-generic x86_64)

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

Pueden actualizarse 139 paquetes.
?? actualizaciones son de seguridad.

root@UbuntuSWAP2:~# _

```

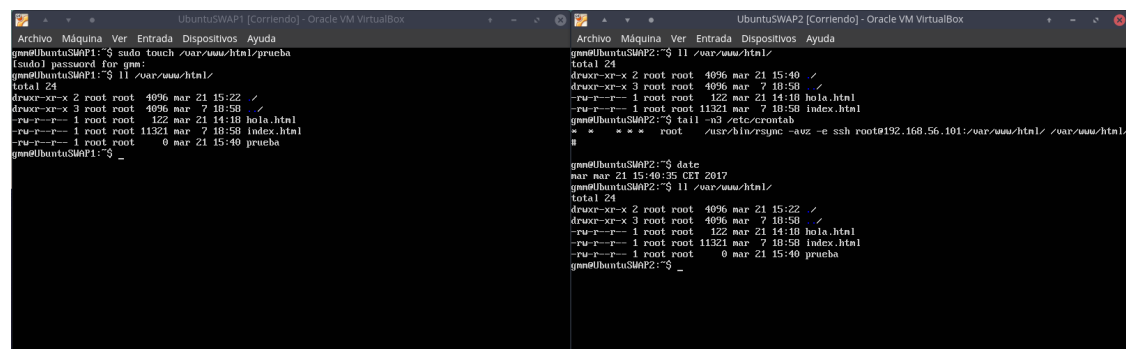
Figura 3.2: Copiando la clave pública en la máquina 2.

4. Cuestión 4

4.1. Establecer una tarea en cron que se ejecute cada hora para mantener actualizado el contenido del directorio `/var/www` entre las dos máquinas.

Aunque no se pide en el enunciado, primero se realizará una prueba que ejecutará el clonado cada minuto, de manera que podamos comprobar fácilmente si esté ha tenido lugar. Se realiza esto debido a los numerosos problemas encontrados con el uso del demonio cron a lo largo de la carrera.

Para la prueba, crearemos un fichero *prueba* en el directorio `/var/www/html/` de la máquina 1, el cual queremos clonar en la máquina 2. Para ello, editamos el fichero `/etc/crontab` [7] de la máquina 2 y añadimos el mismo comando *rsync* usado en la cuestión 2. Como ya se comentó, primero se establecerá cada minuto, al contrario de lo que se pide.



```
UbuntuSWAP1 [Corriendo] - Oracle VM VirtualBox
gms@UbuntuSWAP1:~$ sudo touch /var/www/html/prueba
[sudo] password for gms:
gms@UbuntuSWAP1:~$ ll /var/www/html/
total 24
drwxr-xr-x 2 root root 4096 mar 21 15:22 /
drwxr-xr-x 3 root root 4096 mar  7 18:58 /
-rw-r--r-- 1 root root 122 mar 21 14:18 hola.html
-rw-r--r-- 1 root root 11321 mar  7 18:58 index.html
-rw-r--r-- 1 root root   0 mar 21 15:40 prueba
gms@UbuntuSWAP1:~$ _

UbuntuSWAP2 [Corriendo] - Oracle VM VirtualBox
gms@UbuntuSWAP2:~$ ll /var/www/html/
total 24
drwxr-xr-x 2 root root 4096 mar 21 15:40 /
drwxr-xr-x 3 root root 4096 mar  7 18:58 /
-rw-r--r-- 1 root root 122 mar 21 14:18 hola.html
-rw-r--r-- 1 root root 11321 mar  7 18:58 index.html
gms@UbuntuSWAP2:~$ tail -n3 /etc/crontab
* * * * * root    /usr/bin/rsync -avz -e ssh root@192.168.56.101:/var/www/html/ /var/www/html/
#
gms@UbuntuSWAP2:~$ date
mar mar 21 15:40:35 CET 2017
gms@UbuntuSWAP2:~$ ll /var/www/html/
total 24
drwxr-xr-x 2 root root 4096 mar 21 15:22 /
drwxr-xr-x 3 root root 4096 mar  7 18:58 /
-rw-r--r-- 1 root root 122 mar 21 14:18 hola.html
-rw-r--r-- 1 root root 11321 mar  7 18:58 index.html
-rw-r--r-- 1 root root   0 mar 21 15:40 prueba
gms@UbuntuSWAP2:~$ _
```

Figura 4.1: Haciendo uso de crontab.

Como podemos comprobar, la tarea se ha ejecutado satisfactoriamente. De esta manera, ya podemos establecer el periodo correcto, el cual se pide que sea cada hora. Para ello, tan solo tendremos que modificar el primer asterisco por un 0 (se ejecutará en el minuto 0 de cada hora), de manera que la sentencia quede así:

Listing 2: crontab

```
0 * * * * /usr/bin/rsync -avz -e ssh root@192.168.56.101:/var/www/html/ /var/www/html/
```

Como nota aclaratoria, se han usado paths absolutos, incluso en el comando, porque el uso de paths relativos suele dar problemas al tratarse de diferentes entornos.

Referencias

- [1] VirtualBox. *Chapter 6. Virtual networking*. Consultado el 21 de marzo de 2017.
- [2] *scp - Linux man page*. Consultado el 21 de marzo de 2017.
- [3] *tar - Linux man page*. Consultado el 21 de marzo de 2017.
- [4] *rsync - Linux man page*. Consultado el 21 de marzo de 2017.
- [5] *ssh-keygen - Linux man page*. Consultado el 21 de marzo de 2017.
- [6] *ssh-copy-id - Linux man page*. Consultado el 21 de marzo de 2017.
- [7] *crontab - Linux man page*. Consultado el 21 de marzo de 2017.