

A dark blue vertical bar runs along the left edge of the page. A blue arrow-shaped banner points to the right from this bar, containing the text 'DESPLIEGUE DE APLICACIONES WEB'. In the bottom-left corner, there are several thin, curved lines in dark blue and light grey, creating an abstract, organic shape.

DESPLIEGUE DE APLICACIONES WEB

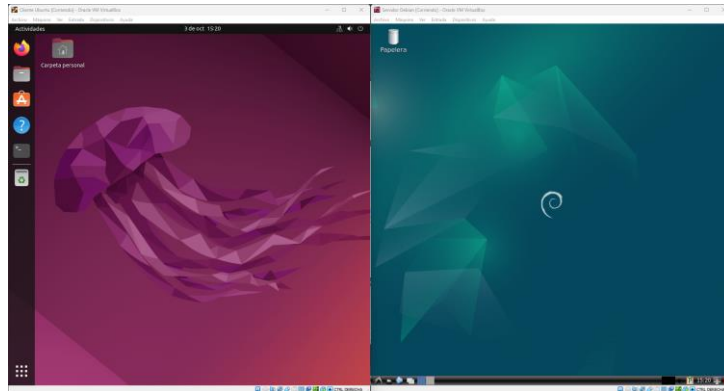
CONEXIÓN CLIENTE- SERVIDOR POR SSH

ÍNDICE

INTRODUCCIÓN	3
COMPROBACIÓN DEL ESTADO DEL SSH	4
COMPROBACIÓN DE LAS LLAVES PÚBLICA Y PRIVADA DEL SERVIDOR	5
CONEXIÓN DESDE EL CLIENTE AL SERVIDOR	5
CONEXIÓN POR SSH CON LLAVES PÚBLICAS EN CLIENTE	6

INTRODUCCIÓN

En esta práctica vamos a realizar una conexión SSH entre una máquina cliente (Ubuntu) y una máquina servidor (Debian).



Lo primero que se tendrá que realizar es una conexión por red Nat entre ambas máquinas.

A continuación, comprobaremos las ip de cada máquina (comando ip a):

```
albertom-cliente@albertoMartinezPerez:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:29:89:4e brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.4/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 550sec preferred_lft 550sec
    inet6 fe80::cd4a:79cf:f3b7:e9f8/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

```
albertom-servidor@servidor-debian:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,DYNAMIC,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c9:79:df brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.5/24 brd 10.0.2.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fec9:79df/64 scope link
        valid_lft forever preferred_lft forever
```

Y realizar un ping cliente-servidor y un ping servidor-cliente para comprobar que son visibles:

```
albertom-cliente@albertoMartinezPerez:~$ ping 10.0.2.5
PING 10.0.2.5 (10.0.2.5) 56(84) bytes of data.
64 bytes from 10.0.2.5: icmp_seq=1 ttl=64 time=1.67 ms
64 bytes from 10.0.2.5: icmp_seq=2 ttl=64 time=1.02 ms
64 bytes from 10.0.2.5: icmp_seq=3 ttl=64 time=2.39 ms
64 bytes from 10.0.2.5: icmp_seq=4 ttl=64 time=1.14 ms
64 bytes from 10.0.2.5: icmp_seq=5 ttl=64 time=1.60 ms

--- 10.0.2.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4011ms
rtt min/avg/max/mdev = 1.015/1.560/2.385/0.482 ms

albertom-servidor@servidor-debian:~$ ping 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.
64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=1.00 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=0.918 ms
64 bytes from 10.0.2.4: icmp_seq=3 ttl=64 time=1.44 ms
64 bytes from 10.0.2.4: icmp_seq=4 ttl=64 time=1.62 ms
64 bytes from 10.0.2.4: icmp_seq=5 ttl=64 time=0.859 ms

^C
--- 10.0.2.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 0.859/1.167/1.618/0.304 ms
```

COMPROBACIÓN DEL ESTADO DEL SSH

En la máquina servidora introducimos el siguiente comando:

```
systemctl status sshd
```

Con ello comprobamos el estado del ssh:

```
albertom-servidor@servidor-debian:~$ systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Fri 2023-10-06 18:17:42 CEST; 7min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 636 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 716 (sshd)
      Tasks: 1 (limit: 2284)
     Memory: 5.7M
        CPU: 66ms
    CGroup: /system.slice/ssh.service
            └─716 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Warning: some journal files were not opened due to insufficient permissions.
albertom-servidor@servidor-debian:~$
```

En el caso de no tener el servicio instalado podríamos hacer una instalación con el comando:

```
sudo apt-get install openssh-server
```

A continuación, vamos a comprobar que las conexiones activas del servidor, para ello utilizamos el comando

```
netstat -a
```

```
root@servidor-debian:/home/albertom-servidor# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.15:47756         151.101.134.132:http    TIME_WAIT
tcp6       0      0 localhost:domain        [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 localhost:ipp           [::]:*                  LISTEN
udp        0      0 0.0.0.0:631             0.0.0.0:*               *
udp        0      0 10.0.2.15:38586         liveboxfibra:domain     ESTABLISHED
udp        0      0 10.0.2.5:47912          liveboxfibra:domain     ESTABLISHED
udp        0      0 10.0.2.15:56340         liveboxfibra:domain     ESTABLISHED
udp        0      0 0.0.0.0:55339           0.0.0.0:*               *
udp        0      0 localhost:domain        0.0.0.0:*               *
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*               *
udp        0      0 0.0.0.0:mdns             0.0.0.0:*               *
udp6       0      0 [::]:59222              [::]:*                  *
udp6       0      0 localhost:domain        [::]:*                  *
udp6       0      0 [::]:mdns                [::]:*                  *
```

En el caso de no tener las herramientas instaladas, las instalaremos con el comando:

```
sudo apt-get install net-tools
```

COMPROBACIÓN DE LAS LLAVES PÚBLICA Y PRIVADA DEL SERVIDOR

Como root (su root) ejecutamos el siguiente comando:

```
ls -la /etc/ssh/
```

```
root@servidor-debian:/home/albertom-servidor# ls -la /etc/ssh
total 612
drwxr-xr-x  4 root root   4096 oct  6 15:20 .
drwxr-xr-x 118 root root   4096 oct  6 18:17 ..
-rw-r--r--  1 root root 573928 feb  8 2023 moduli
-rw-r--r--  1 root root  1650 feb  8 2023 ssh_config
drwxr-xr-x  2 root root   4096 feb  8 2023 ssh_config.d
-rw-r--r--  1 root root  3223 feb  8 2023 sshd_config
drwxr-xr-x  2 root root   4096 feb  8 2023 sshd_config.d
-rw-----  1 root root   513 oct  6 15:20 ssh_host_ecdsa_key
-rw-r--r--  1 root root   182 oct  6 15:20 ssh_host_ecdsa_key.pub
-rw-----  1 root root   411 oct  6 15:20 ssh_host_ed25519_key
-rw-r--r--  1 root root   102 oct  6 15:20 ssh_host_ed25519_key.pub
-rw-----  1 root root  2610 oct  6 15:20 ssh_host_rsa_key
-rw-r--r--  1 root root   574 oct  6 15:20 ssh_host_rsa_key.pub
```

Con ello veremos el contenido del directorio /etc/ssh/ y nos fijaremos en que tenemos 3 claves privadas y 3 claves públicas:

- Un par de claves con el algoritmo ecdsa.

```
ssh_host_ecdsa_key
ssh_host_ecdsa_key.pub
```

- Un par de claves con el algoritmo ed25519.

```
ssh_host_ed25519_key
ssh_host_ed25519_key.pub
```

- Un par de claves con el algoritmo rsa.

```
ssh_host_rsa_key
ssh_host_rsa_key.pub
```

CONEXIÓN DESDE EL CLIENTE AL SERVIDOR

Vamos a conectarnos desde el cliente a la cuenta del servidor. Para ello vamos a usar el comando:

```
ssh nombreUsuarioServidor@IPServidor
```

```
albertom-cliente@albertoMartinezPerez:~$ ssh albertom-servidor@10.0.2.5
The authenticity of host '10.0.2.5 (10.0.2.5)' can't be established.
ED25519 key fingerprint is SHA256:opGpBuUkklMIbeFfbVw7tMdvo42tix/AbXbsbHAQldc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Lo que ha sucedido es que el cliente se ha traído una de las llaves públicas del servidor (en concreto la de cifrado ed25519) y ahora nos solicitara la contraseña de la cuenta del servidor.

Si introducimos los datos de forma correcta nos conectaremos.

```
albertom-cliente@albertoMartinezPerez:~$ ssh albertom-servidor@10.0.2.5
The authenticity of host '10.0.2.5 (10.0.2.5)' can't be established.
ED25519 key fingerprint is SHA256:opGpBUKkLMibeFfbVw7tMdvo42tix/AbXbsbHAQldc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.5' (ED25519) to the list of known hosts.
albertom-servidor@10.0.2.5's password:
Linux servidor-debian 6.1.0-12-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.52-1 (2023-09-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

Esto ha hecho que se almacene un fichero que contiene la clave pública del servidor SSH, este fichero se llama `known_host` y es visible en la carpeta `.ssh` (la cual se crea cuando se lanza una orden `ssh`).

```
albertom-cliente@albertoMartinezPerez:~$ ls -la .ssh
total 16
drwx----- 2 albertom-cliente albertom-cliente 4096 oct  6 18:27 .
drwxr-x--- 15 albertom-cliente albertom-cliente 4096 oct  6 18:27 ..
-rw----- 1 albertom-cliente albertom-cliente  978 oct  6 18:27 known_hosts
-rw-r--r-- 1 albertom-cliente albertom-cliente  142 oct  6 18:27 known_hosts.old
```

Si queremos ver el aspecto que tiene una llave pública podemos hacerlo con el siguiente comando:

`cat .ssh/known_hosts`

```
albertom-cliente@albertoMartinezPerez:~$ cat .ssh/known_hosts
|1|Uy6u5Ffbby6P0mFh2MwcvH92Eq8=|T0dUjFrh0FF0uXx48PL0+hkKXoQ= ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAA
IPOXecgIuuuqvqdM1W3FM0AKiA84a5jrTx0MwkkFln/e
|1|S9uunfbfjNbICURJbaapvmWbmCo=|42FGCW1VTiU6M8hmbUHEsf6A0U= ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQ=
gQDCwY3zsets0sybd30sLRNPumvuQtIJBu39nfnmuc7HeE29p8SbB3AzcfJt3dKYFF72T6o7r56BJSHMUjEgCPPIkQEWJAZ
SPKPe+www6A9EQaTdDVIDc9f8Fec+/o9fh+B6uCKsZloPPrc38XfaYlFhPQvCV1v0Sb70qREMGIFJFIgQzTdInSBFyYcrLVJW
WL27A8vkByNUhAvBZu4l/Rm3JTOHP0FEXW3q42BV7/nKc7bsB7nSRWMqVvIX3DUGX6bNhdw/4LRiczxwY+n5VR3j94T6I3efR
R35RT3kZwvqp2k9Cx5zXCrnf0evVVDhQ1vriICUfgIMUV6cIRCGJrkYnEmGTeeU/tFtpx+ZZZiHMLmOSLJR1KRgjdnp6Cw7
qr7ZmV7ZzmM9HL2PurHE0SUsiBqoUUUAQz4d6G/ae5Cpa9qRILX7axMrLS1DS65BvfIkeYD0cjcghkxMKRnnVZVUJpuGxAr
8jCq3kVAsIcV90J0gdyRx1PzEWPmk=
|1|fm8BHSFZpy0rOFJB1S65BONN1Uo=|Nxa0HCNp9sFQ5yixD1qt1wUyC48= ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNo
YTItbmlzdHAYNTYAAAAIbmlzdHAYNTYAAABBLayMNLYJkLpNfwltbIEcMuWqHRU5oKBKfStSwqK54TqyVAGAsydU2UsHNRNV
6cdD54CmAv7aMxIn5hTS9Fbr1Q=
```

En el lado del servidor también podemos verlo con otro `cat`, en este caso hacia una de las llaves que hemos visto más arriba, por ejemplo, a la `ed25519_key.pub`.

```
root@servidor-debian:/home/albertom-servidor# cat /etc/ssh/ssh_host_ed25519_key.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIPOXecgIuuuqvqdM1W3FM0AKiA84a5jrTx0MwkkFln/e root@servidor-debian
```

CONEXIÓN POR SSH CON LLAVES PÚBLICAS EN CLIENTE

Ahora vamos a crear un usuario en el cliente que tenga el mismo nombre que el del servidor:

```
albertom-cliente@albertoMartinezPerez:~$ sudo adduser albertom-servidor -home /home/albertom-servidor
[sudo] contraseña para albertom-cliente:
Añadiendo el usuario 'albertom-servidor' ...
Añadiendo el nuevo grupo 'albertom-servidor' (1001) ...
Añadiendo el nuevo usuario 'albertom-servidor' (1001) con grupo 'albertom-servidor' ...
Creando el directorio personal '/home/albertom-servidor' ...
Copiando los ficheros desde '/etc/skel' ...
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña tiene menos de 8 caracteres
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para albertom-servidor
Introduzca el nuevo valor, o presione INTRO para el predeterminado
Nombre completo []: albertom-servidor
Número de habitación []:
Teléfono del trabajo []:
Teléfono de casa []:
Otro []:
¿Es correcta la información? [S/n] s
```

Nos conectamos con su cuenta:

```
albertom-cliente@albertoMartinezPerez:~$ su albertom-servidor
Contraseña:
```

```
albertom-servidor@albertoMartinezPerez:~$ ls -la
total 20
drwxr-x--- 2 albertom-servidor albertom-servidor 4096 oct  6 18:33 .
drwxr-xr-x 4 root              root            4096 oct  6 18:33 ..
-rw-r--r-- 1 albertom-servidor albertom-servidor 220 oct  6 18:33 .bash_logout
-rw-r--r-- 1 albertom-servidor albertom-servidor 3771 oct  6 18:33 .bashrc
-rw-r--r-- 1 albertom-servidor albertom-servidor  807 oct  6 18:33 .profile
```

Y crearemos las llaves rsa con el comando:

ssh-keygen -t rsa -b 4096

```
albertom-servidor@albertoMartinezPerez:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/albertom-servidor/.ssh/id_rsa): /home/albertom-servidor/.ssh/albertom-servidor
Created directory '/home/albertom-servidor/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/albertom-servidor/.ssh/albertom-servidor
Your public key has been saved in /home/albertom-servidor/.ssh/albertom-servidor.pub
The key fingerprint is:
SHA256:AaynTTta2nG6U3ShSFcYe1QPfsGU9ktt9qsr/zE+jmg albertom-servidor@albertoMartinezPerez
The key's randomart image is:
+----[RSA 4096]-----+
| .. .+0.000. |
| 0.00.. 0+. |
| 0 00.....0..|
| . + 00. . .|=|
| = 0S. .00|
| . * 0 ..|
| = * 0. |
| 0 + E. 000|
| .0 ..0==+.|
+----[SHA256]-----+
```

Esto nos ha creado una carpeta .ssh y dos ficheros, uno con la clave pública y otro con la clave privada:

```
albertom-servidor@albertoMartinezPerez:~$ ls -la .ssh
total 16
drwx----- 2 albertom-servidor albertom-servidor 4096 oct  6 18:37 .
drwxr-x--- 3 albertom-servidor albertom-servidor 4096 oct  6 18:37 ..
-rw----- 1 albertom-servidor albertom-servidor 3414 oct  6 18:37 albertom-servidor
-rw-r--r-- 1 albertom-servidor albertom-servidor  764 oct  6 18:37 albertom-servidor.pub
```

Para poder conectarnos sin contraseña a la MV servidor, se debe copiar allí la clave pública del usuario "alumno-servidor". Para ello debemos hacer uso del proceso ssh-agent.

Para comprobar si el proceso está lanzado en la máquina usamos el comando:

```
ps -ef | grep ssh-agent
```

En el caso de que no lo esté, podemos usar el comando

```
eval "$(ssh-agent)"
```

```
albertom-servidor@albertoMartinezPerez:~$ eval "$(ssh-agent)"
Agent pid 3806
albertom-servidor@albertoMartinezPerez:~$ ps -ef | grep ssh-agent
alberto+ 2363 1602 0 18:27 ? 00:00:00 /usr/bin/ssh-agent -D -a /run/user/1000/keyring/.ssh
alberto+ 3806 1541 0 18:38 ? 00:00:00 ssh-agent
alberto+ 3808 3720 0 18:38 pts/0 00:00:00 grep --color=auto ssh-agent
```

Ahora debemos comprobar las claves públicas registradas por ssh-agent en la máquina con el comando:

```
ssh-add -l
```

```
albertom-servidor@albertoMartinezPerez:~$ ssh-add -l
The agent has no identities.
```

En el caso de que aparezca vacío, como se ve en la imagen, debemos añadir la clave pública, para ello usamos el comando:

```
ssh-add /home/alumno-servidor/.ssh/alumno-servidor
```

```
albertom-servidor@albertoMartinezPerez:~$ ssh-add /home/albertom-servidor/.ssh/albertom-servidor
Identity added: /home/albertom-servidor/.ssh/albertom-servidor (albertom-servidor@albertoMartinezPerez)
albertom-servidor@albertoMartinezPerez:~$ ssh-add -l
4096 SHA256:AaynTTta2nG6U3ShSFCYe1QPfsGU9ktt9qsr/zE+jmg albertom-servidor@albertoMartinezPerez (RSA)
```

Para borrar claves registradas podemos usar el comando `ssh-add -D`.

Ahora vamos a hacer una copia de la clave pública en la máquina servidor, para ello se utiliza el comando `ssh-copy-id usuario@IP_servidor`, en este caso:

```
ssh-copy-id albertom-servidor@10.0.2.5
```

```
albertom-servidor@albertoMartinezPerez:~$ ssh-copy-id albertom-servidor@10.0.2.5
The authenticity of host '10.0.2.5 (10.0.2.5)' can't be established.
ED25519 key fingerprint is SHA256:opGpBuUKlMIbEfbVw7tMdvo42tix/AbXbsbHAQldc.
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 i
nstalled
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install th
e new keys
albertom-servidor@10.0.2.5's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'albertom-servidor@10.0.2.5'"
and check to make sure that only the key(s) you wanted were added.
```

Si ahora miramos en la máquina virtual del servidor, veremos que se ha creado un directorio donde se guarda la clave recién copiada:

```
albertom-servidor@servidor-debian:~$ ls -la .ssh
total 12
drwx----- 2 albertom-servidor albertom-servidor 4096 oct 6 18:41 .
drwx----- 14 albertom-servidor albertom-servidor 4096 oct 6 18:41 ..
-rw----- 1 albertom-servidor albertom-servidor 764 oct 6 18:41 authorized_keys
```



```
albertom-servidor@servidor-debian:~$ cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQC7eNz7cvMIFbKff5Epj0hbkiWPuoIgv+Ev0d6g0jGd64it3NVKCq21Xwmzd6xJsBfWpHN7Hgy0oeST
G1DB9Dz5npJqLLmwi9EfJEier9iPpw9AD7WvLUn6qvDBp0gms2v5C2eyi4DSsGrGK18bizGwJCCbTXdWIBmGaRCogZzsJmSXydduiYfpdDhXjc86swJKp
QNMFeSeInoCEjnfG/mH0aE2HDy6H1k2IVX/Bzhb7ME6j+3270Y2U2X/DUKd83uYbA9Lt0W5UrU86pmf+v/WHM9Kx58uejvtBzM3sTZVqgVYhgYItFNv1
qtVbkciTLGTbGwDjVjidPkrybYTKPZI+F5p8nVQffM22pEAV9ToaKh7uNGxeantv7M0sXh15lncQuu0SZdvtHuZts/DXIP95jJwT8H72+XQXdz0IfzKMkJ
dCF4lWI2hyEzRGXYv9zKkqSbjd2MtnBNVjsj6xUH1NHvIu+RaTDqcDHyYr0yNI8u6RchEuFpHXc+LYTfUdNgv9V5m3bAnt4ox++V1CAQMyqWxSswx5D
GRzJEI999CNsS7rf6P8THkizAKKuFsutUU6uDiY9NPdKrdp3bR0ihiz1A4a20fiI0Se9KvvFXnG1W0dURAUdg3CAvdxNA0y9QcdYMU+AMaDuYFEGMMbRN
fynd/LchbmZ1uyMKQYew== albertom-servidor@albertoMartinezPerez
```

Ahora podremos conectarnos desde el cliente al servidor sin necesidad de introducir contraseña:

```
albertom-servidor@albertoMartinezPerez:~$ ssh albertom-servidor@10.0.2.5
Linux servidor-debian 6.1.0-12-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.52-1 (2023-09-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Oct  6 18:52:10 2023 from 10.0.2.4
```