



Pt1d - Instal·lació i configuració de SSH

Estudiant (COGNOMS, NOM):

RISSECH GASULLA, JORDI

Professors:

Mercedes Castellón

Objectius generals de la pràctica

Configuració d'un servidor de SSH, openssh-server

Desenvolupament de la pràctica

- Ubuntu server com a servidor.
- Windows o Ubuntu Desktop com a client.
- També pots utilitzar la màquina *host* com a client, si és Linux.

Explica quins arxius has fet servir per cadascun dels apartats i com els has configurat.
Si modifiques un fitxer de configuració, SEMPRE fer una còpia abans.

Part 1 – Servei SSH i administració remota

1. Al servidor comprova l'estat del servei SSH.

```
Se pueden aplicar 0 actualizaciones de forma inmediata.

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

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

jordi@jordiriszech:~$ systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-10-26 11:29:42 UTC; 14min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 1278 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 1279 (sshd)
    Tasks: 1 (limit: 4557)
   Memory: 1.7M
      CPU: 44ms
   CGroup: /system.slice/ssh.service
           └─1279 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

oct 26 11:29:42 jordiriszech systemd[1]: Starting OpenBSD Secure Shell server...
oct 26 11:29:42 jordiriszech sshd[1279]: Server listening on 0.0.0.0 port 22.
oct 26 11:29:42 jordiriszech sshd[1279]: Server listening on :: port 22.
oct 26 11:29:42 jordiriszech systemd[1]: Started OpenBSD Secure Shell server.
jordi@jordiriszech:~$ _
```



2. Al servidor comprova el port del servei SSH

```
jordi@jordirissec:~$ sudo ss -tlnp
[sudo] password for jordi:
State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
LISTEN 0      4096      127.0.0.1:53         0.0.0.0:*   users:(("systemd-resolve",pid=629,fd=14))
LISTEN 0       128        0.0.0.0:22         0.0.0.0:*   users:(("sshd",pid=1279,fd=3))
LISTEN 0       128        [::]:22            [::]:*     users:(("sshd",pid=1279,fd=4))
jordi@jordirissec:~$ _
```

Com es pot veure, el port 22, corresponent al servei SSH, està actiu i escoltant, ja que el servei està activat.

3. Mostra l'adreça IP del teu servidor

```
jordi@jordirissec:~$ 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 08:00:27:45:6d:1b brd ff:ff:ff:ff:ff:ff
    inet 192.168.128.163/24 metric 100 brd 192.168.128.255 scope global dynamic enp0s3
        valid_lft 1979sec preferred_lft 1979sec
    inet6 fe80::a00:27ff:fe45:6d1b/64 scope link
        valid_lft forever preferred_lft forever
```

L'adreça IP del servidor és **192.168.128.163/24**.

4. Conectar per SSH modo línia (amb el terminal de windows o ubuntu). Executa alguna ordre al servidor de forma remota.

Ens connectem al terminal amb **ssh <usuari>@<ip>**. En aquest cas, **ssh jordi@192.168.128.163**.

```
Microsoft Windows [Versión 10.0.17763.4974]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\jori9110>ssh jordi@192.168.128.163
The authenticity of host '192.168.128.163 (192.168.128.163)' can't be established.
ECDSA key fingerprint is SHA256:RSB2e036NtpWA4BYSATLk1YVtGWEOWIW98cgNAHWTF0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.128.163' (ECDSA) to the list of known hosts.
jordi@192.168.128.163's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-78-generic x86_64)

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

System information as of jue 26 oct 2023 11:54:12 UTC

System load:  0.04296875      Processes:            103
Usage of /:   35.9% of 18.53GB Users logged in:       1
Memory usage: 6%             IPv4 address for enp0s3: 192.168.128.163
Swap usage:   0%

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 0 actualizaciones de forma inmediata.

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

Last login: Thu Oct 26 11:33:18 2023
jordi@jordirissec:~$
```



Executem un parell de comandes per rebre informació sobre les connexions de xarxa de la màquina servidor.

ip a ens permet veure informació sobre els adaptadors de xarxa de la màquina.

ss -tlnp ens permet veure informació sobre els ports utilitzats. Aquesta comanda requereix **sudo** per executar-se que, com es pot veure, funciona també a través d'SSH.

```
jordi@jordinissech:~$ 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 08:00:27:45:6d:1b brd ff:ff:ff:ff:ff:ff
    inet 192.168.128.163/24 metric 100 brd 192.168.128.255 scope global dynamic enp0s3
        valid_lft 1638sec preferred_lft 1638sec
    inet6 fe80::a00:27ff:fe45:6d1b/64 scope link
        valid_lft forever preferred_lft forever
jordi@jordinissech:~$ sudo ss -tlnp
[sudo] password for jordi:
State      Recv-Q    Send-Q    Local Address:Port    Peer Address:Port    Process
LISTEN     0          4096      127.0.0.53%lo:53      0.0.0.0:*             users:((("systemd-resolve",pid=629,fd=14))
LISTEN     0          128       0.0.0.0:22           0.0.0.0:*             users:((("sshd",pid=1279,fd=3)))
LISTEN     0          128       [::]:22              [::]:*                users:((("sshd",pid=1279,fd=4)))
```

5. Modo línia, busca la manera d'executar comandes remotament sense entrar en mode interactiu. És a dir, has de poder entrar en el servidor, executar la comanda i sortir, escrivint una sola comanda des del client.

Per executar comandes en mode **no interactiu**, hem de fer servir la següent sintaxi:

```
ssh <user>@<ip> <command>
```

En aquest cas, utilitzem:

```
ssh jordi@192.168.128.163 ip a
```

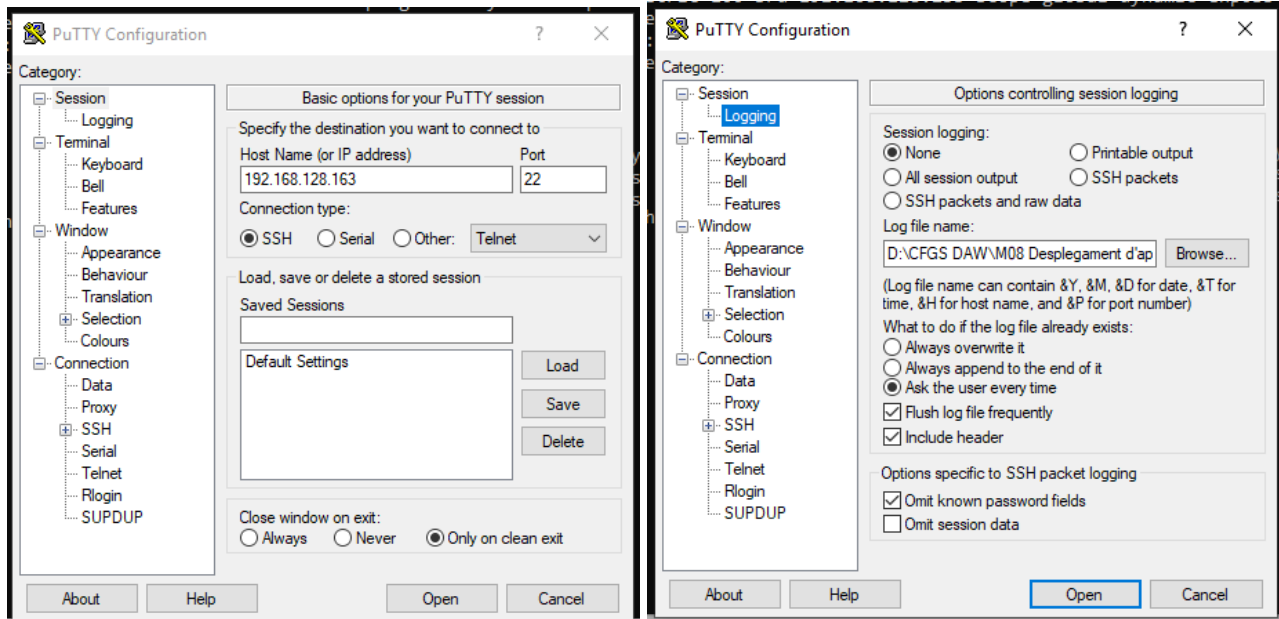
```
ssh jordi@192.168.128.163 ls -la
```

Com es pot veure al final d'aquesta captura, no hem entrat en mode interactiu: seguim tenint la màquina client com a espai de treball.

```
C:\Users\jori9110>ssh jordi@192.168.128.163 ip a
jordi@192.168.128.163's password:
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 08:00:27:45:6d:1b brd ff:ff:ff:ff:ff:ff
    inet 192.168.128.163/24 metric 100 brd 192.168.128.255 scope global dynamic enp0s3
        valid_lft 85320sec preferred_lft 85320sec
    inet6 fe80::a00:27ff:fe45:6d1b/64 scope link
        valid_lft forever preferred_lft forever
```

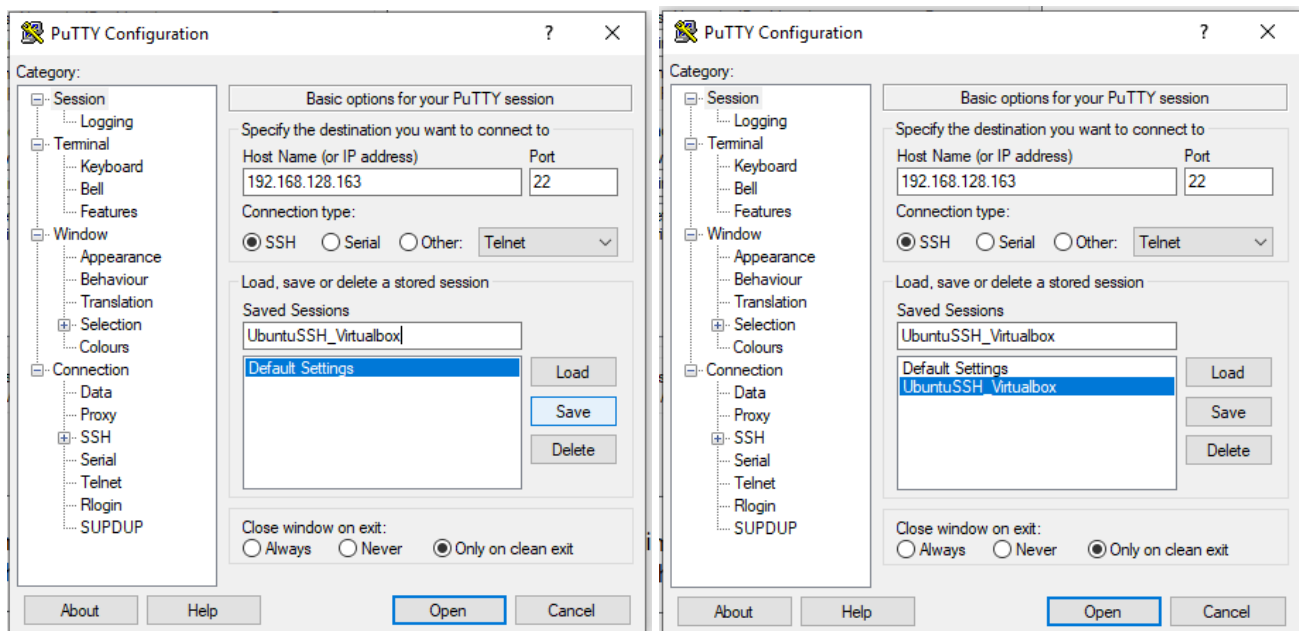
```
C:\Users\jori9110>ssh jordi@192.168.128.163 ls -la
jordi@192.168.128.163's password:
total 32
drwxr-x--- 4 jordi jordi 4096 oct 26 11:48 .
drwxr-xr-x 3 root  root  4096 oct 26 11:29 ..
-rw-r--r-- 1 jordi jordi  220 ene  6 2022 .bash_logout
-rw-r--r-- 1 jordi jordi 3771 ene  6 2022 .bashrc
drwx----- 2 jordi jordi 4096 oct 26 11:33 .cache
-rw----- 1 jordi jordi   20 oct 26 11:44 .lessht
-rw-r--r-- 1 jordi jordi  807 ene  6 2022 .profile
drwx----- 2 jordi jordi 4096 oct 26 11:29 .ssh
-rw-r--r-- 1 jordi jordi   0 oct 26 11:48 .sudo_as_admin_successful
C:\Users\jori9110>
```

6. Conecta amb el putty, guardant les dades de la connexió. Executa alguna ordre al servidor de forma remota.



Configurem la connexió introduint l'adreça IP. Com a pas addicional, assignem un arxiu de per fer el *logging* de la sessió.

Desem la informació d'accés i fem click a **Open**.



El terminal de **putty** ens demana les credencials.





Un cop introduïdes les credencials, tenim accés a la màquina remota i podem executar comandes.

```
Last login: Thu Oct 26 11:54:13 2023 from 192.168.128.148
jordi@jordirissec:~$ ls -la
total 32
drwxr-x--- 4 jordi jordi 4096 oct 26 11:48 .
drwxr-xr-x 3 root  root  4096 oct 26 11:29 ..
-rw-r--r-- 1 jordi jordi  220 ene  6  2022 .bash_logout
-rw-r--r-- 1 jordi jordi 3771 ene  6  2022 .bashrc
drwx----- 2 jordi jordi 4096 oct 26 11:33 .cache
-rw----- 1 jordi jordi   20 oct 26 11:44 .lessht
-rw-r--r-- 1 jordi jordi  807 ene  6  2022 .profile
drwx----- 2 jordi jordi 4096 oct 26 11:29 .ssh
-rw-r--r-- 1 jordi jordi    0 oct 26 11:48 .sudo_as_admin_successful
```

Part 2 - SSH keys

Busca un article que expliqui com fer SSH entre 2 màquines sense necessitat de introduir contrasenya. El client (imitant un PC de casa contra un servidor remot) es connectarà al servidor sense contrasenya. Pista: es recomana la documentació oficial de Debian o d'Ubuntu, son molt clares i és un tema típic. INFORME: llista tots els arxius i comandes implicats i explica per a què serveix cadascun. En particular:

1. Connecta't en modo línia sense pasword, fent servir la clau SSH


Per tal de poder-nos connectar sense contrasenyes, fent servir claus SSH, primer hem de crear una parella de claus pública/privada. Per fer-ho, utilitzem la comanda **ssh-keygen**.

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

Per defecte, les claus generades són de 3072 bits. L'opció "**-b 4096**" fa que aquestes claus siguin de 4096 bits.

```
C:\Users\jori9110>ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\jori9110\.ssh\id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\jori9110\.ssh\id_rsa.
Your public key has been saved in C:\Users\jori9110\.ssh\id_rsa.pub.
The key fingerprint is:
SHA256:Mb9SMr1++d094V3tb337t2KFSkQnf0JkGyVKIuShVSU proven\jori9110@A28-12
The key's randomart image is:
+---[RSA 4096]---+
|      .=.E.O.=.. |
|    + O +000+   |
|  . .O ..=.    |
|      = . O .   |
|    S +   +    |
|      = O . ..  |
|    O O....O+   |
|  . .O..OO.X    |
|    .. .O..*&   |
+---[SHA256]-----+
```

Copiem l'arxiu **id_rsa.pub** generat a la carpeta **.ssh** per tal de tenir-lo en una carpeta més accessible. Aquest pas és opcional, només per comoditat, ja que podem copiar directament l'arxiu des de la carpeta **.ssh**.

equipo > Disco local (C:) > Usuarios > Jordi Rissech > authorized_keys				▼	🔄
Nombre	Fecha de modificación	Tipo	Tamaño		
 id_rsa	08/11/2023 20:11	Documento de Mi...	1 KB		

Com es pot observar, he canviat d'ordinador per acabar la pràctica.

```
C:\Users\Jordi Rissech>scp authorized_keys/id_rsa.pub jordi@192.168.1.41:~/ssh
jordi@192.168.1.41's password:
id_rsa.pub
100% 756 125.9KB/s 00:00
```

Al servidor, comprovem si l'arxiu s'ha copiat correctament llegint-lo mitjançant l'ordre **cat**.

```
jordi@server:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDe1mualMtVmqE25rdJ7aHiVYxVf0mL0K2t0Jw1MRekSWFa6qnxIYDu5LKZjH9t
MTXhatM1b80U7R5A1Swb530ZstJEFehTvvNX7VoYbRQLdokBg8mxyCrBnjcBy3dJRE1Qn/MDA+S1pyYjRnwe3Ny/pH1RCA80QCDW
S7H1uixg5Ry9+27WRqg8R6uG52P0YVWtIhYXcQ85enE120oHMDPct3DsEjRIIw47AsCpaEpKB3YiVoJ8mwR5aqdAdNIYf5UN07eR
AspRi+Ed9aRND0o6jtCCherG3/k80r4jVvEn4FcBVQI63LtysgewZnrQjraUqXRfKeK2HCXUUBxxqzKPf7FQtW0JA1hN3G9Grxd
NzhXJgW3JIET+rPt7tLy+zvgGIU2qwxPP/zWtLJVD6ye0XfgNcOYsD1w1VLD7Eh+qmUzomhJDazk4FJ41t/sxsd6ud/LiMr0F52u
yVTuvRKfh0oaYa3i1f0qr9VuXgTubHHT4rWpjqJwZtChy21S1rsYKah22Q/v3+qSMMdp2PR3i6DdnGSxtkW1FbgRcE3Bm9ikFRvD
o/jSFjT0wadWnXXdtL2fkjgDHgyvdh4K8LRNg/YgtYzEc1QdZBR8Z+ynBHN2R8zJrBH1YrYcQkrqiMrRgUeTQG64NTxp90UsCqw2
YjgsDbQmjDRTN49nuCSn0Q== jordi rissech@DESKTOP-QJI30V7
```

S'ha copiat correctament.

Ens desplacem a la carpeta **.ssh** i llistem els arxius per comprovar que **authorized_keys** existeixi.

```
jordi@server:~$ cd .ssh
jordi@server:~/.ssh$ ls -la
total 12
drwx----- 2 jordi jordi 4096 nov  8 19:21 .
drwxr-x--- 4 jordi jordi 4096 nov  8 19:11 ..
-rw----- 1 jordi jordi    0 nov  8 19:06 authorized_keys
-rw-rw-r-- 1 jordi jordi 756 nov  8 19:21 id_rsa.pub
```

Escrivim, en mode **append**, el contingut de l'arxiu **id_rsa.pub** a l'arxiu **authorized_keys**, per tal d'autoritzar les connexions de la nostra màquina client.

```
jordi@server:~/.ssh$ cat id_rsa.pub >> authorized_keys
```

Comprovem amb **cat** que s'hagi afegit el contingut correctament a l'arxiu **authorized_keys**.

```
jordi@server:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDe1mualMtVmqE25rdJ7aHiVYxVf0mL0K2t0Jw1MRekSWFa6qnxIYDu5LKZjH9t
MTXhatM1b80U7R5A1Swb530ZstJEFehTvvNX7VoYbRQLdokBg8mxyCrBnjcBy3dJRE1Qn/MDA+S1pyYjRnwe3Ny/pH1RCA80QCDW
S7H1uixg5Ry9+27WRqg8R6uG52P0YVWtIhYXcQ85enE120oHMDPct3DsEjRIIw47AsCpaEpKB3YiVoJ8mwR5aqdAdNIYf5UN07eR
AspRi+Ed9aRND0o6jtCCherG3/k80r4jVvEn4FcBVQI63LtysgewZnrQjraUqXRfKeK2HCXUUBxxqzKPf7FQtW0JA1hN3G9Grxd
NzhXJgW3JIET+rPt7tLy+zvgGIU2qwxPP/zWtLJVD6ye0XfgNcOYsD1w1VLD7Eh+qmUzomhJDazk4FJ41t/sxsd6ud/LiMr0F52u
yVTuvRKfh0oaYa3i1f0qr9VuXgTubHHT4rWpjqJwZtChy21S1rsYKah22Q/v3+qSMMdp2PR3i6DdnGSxtkW1FbgRcE3Bm9ikFRvD
o/jSFjT0wadWnXXdtL2fkjgDHgyvdh4K8LRNg/YgtYzEc1QdZBR8Z+ynBHN2R8zJrBH1YrYcQkrqiMrRgUeTQG64NTxp90UsCqw2
YjgsDbQmjDRTN49nuCSn0Q== jordi rissech@DESKTOP-QJI30V7
```




Finalment, ens podem connectar al servidor amb ssh sense necessitat d'introduir la contrasenya.

```
C:\Users\Jordi Rissech>ssh jordi@192.168.1.41
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-88-generic x86_64)

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

System information as of mié 08 nov 2023 19:35:23 UTC

System load:  0.0               Processes:            102
Usage of /:   47.4% of 9.75GB   Users logged in:     1
Memory usage: 11%              IPv4 address for enp0s3: 192.168.1.41
Swap usage:   0%

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 33 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: Wed Nov  8 19:09:42 2023 from 192.168.1.47
jordi@server:~$
```

2. Explica quin arxiu te les claus públiques i quin les privades en servidor i client.

Servidor:

- **Clau pública:** S'emmagatzema normalment a l'arxiu "`~/ssh/authorized_keys`", a la carpeta home de cada usuari que s'utilitzi algun cop per fer una connexió ssh.
- **Clau privada:** La genera automàticament el programa SSH del servidor, i la gestiona el propi programa.

Client:

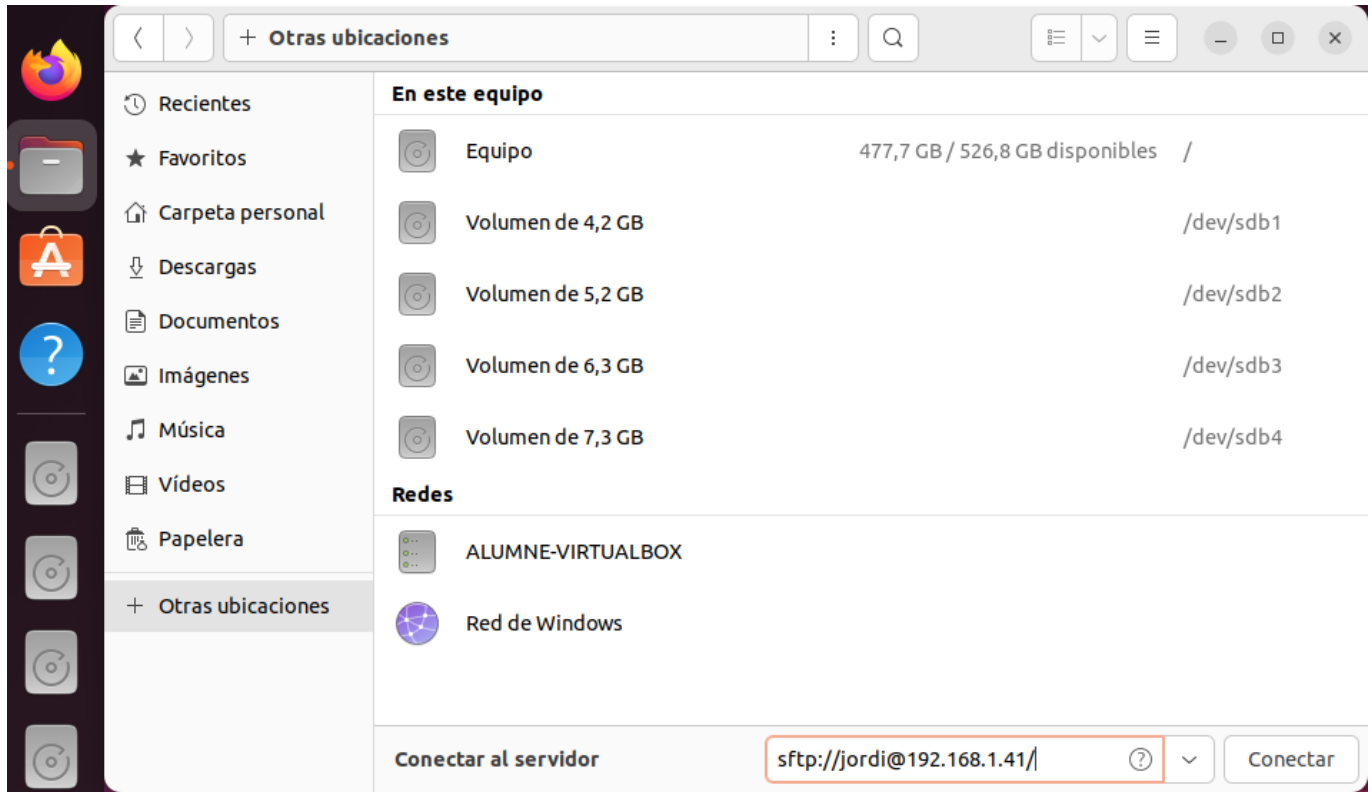
- **Clau pública:** S'emmagatzema normalment a l'arxiu "`C:/Users/my-user/.ssh/id_rsa.pub`". A vegades l'arxiu s'anomena "`C:/Users/my-user/.ssh/id_dsa.pub`".
- **Clau privada:** S'emmagatzema a l'arxiu `id_rsa`, que també s'emmagatzema a la carpeta "`C:/Users/my-user/.ssh/`" (per tant, l'arxiu és "`C:/Users/my-user/.ssh/id_rsa`"). Com amb la clau pública, a vegades, l'arxiu s'anomena `id_dsa`.

Part 3 - SFTP

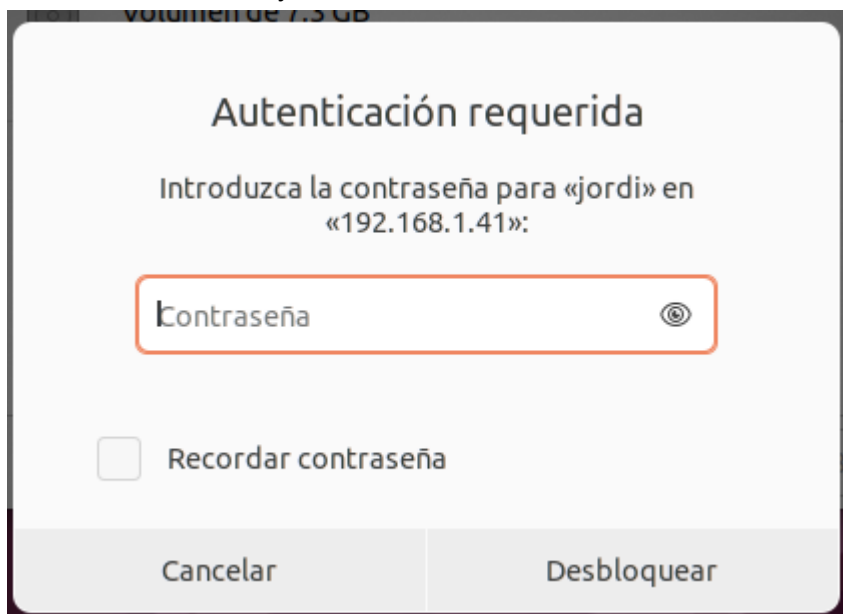
El SSH també serveix per fer transferència segura de fitxers. Realitza:

1. A Ubuntu, navega amb el 'nautilus' (l'explorador d'arxius) pels fitxers del servidor mitjançant el protocol SFTP.

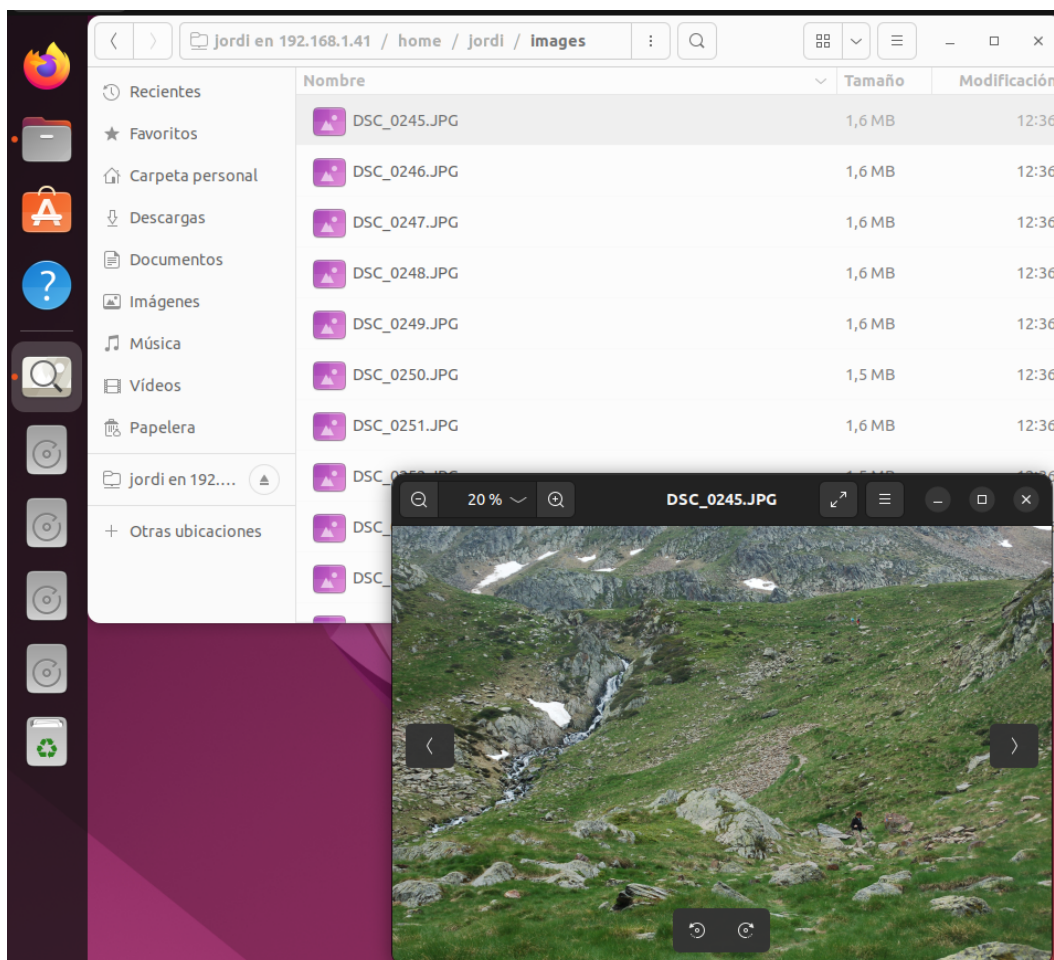
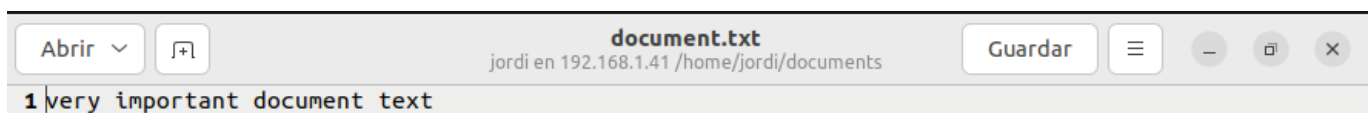
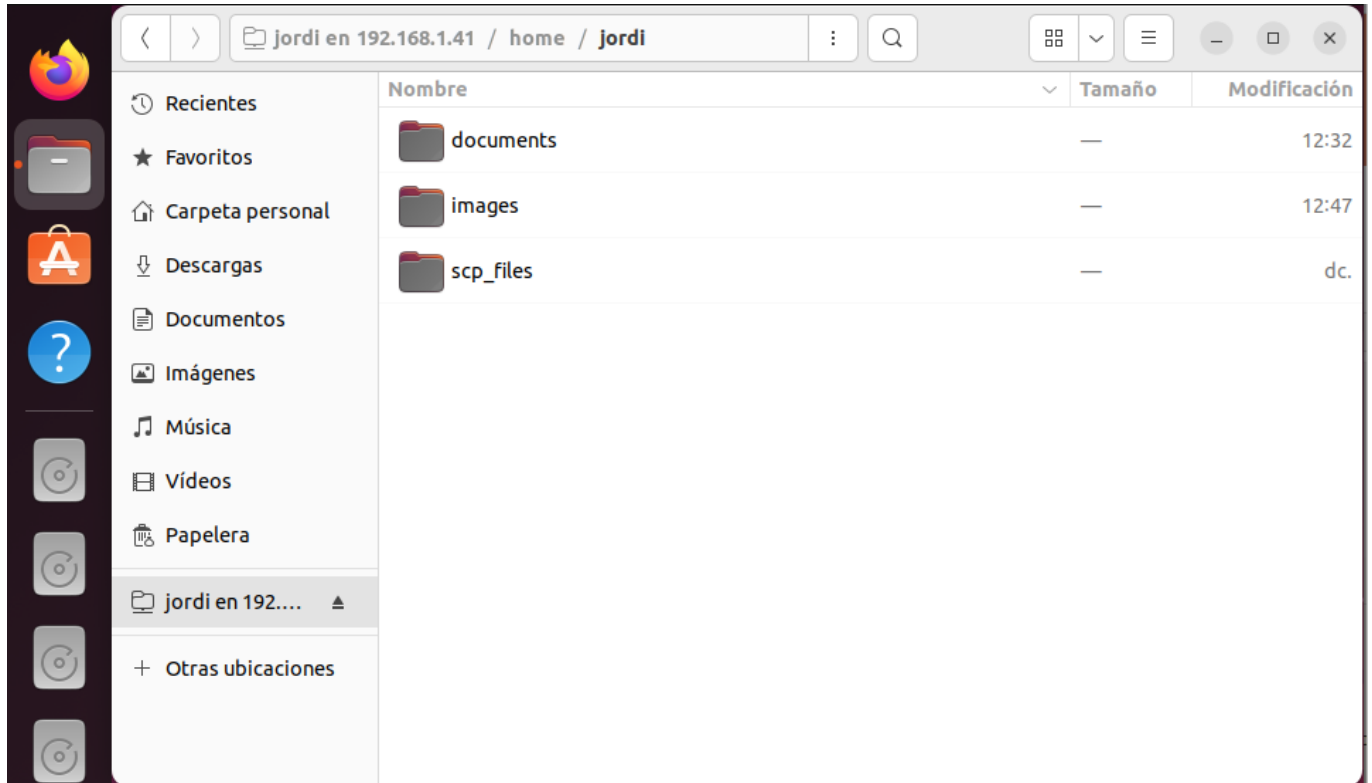
Per connectar-nos mitjançant nautilus a un servidor, configurem una unitat de xarxa nova. Anem a "Otras ubicaciones" i introduïm l'adreça del nostre servidor a l'opció "Conectar al servidor".



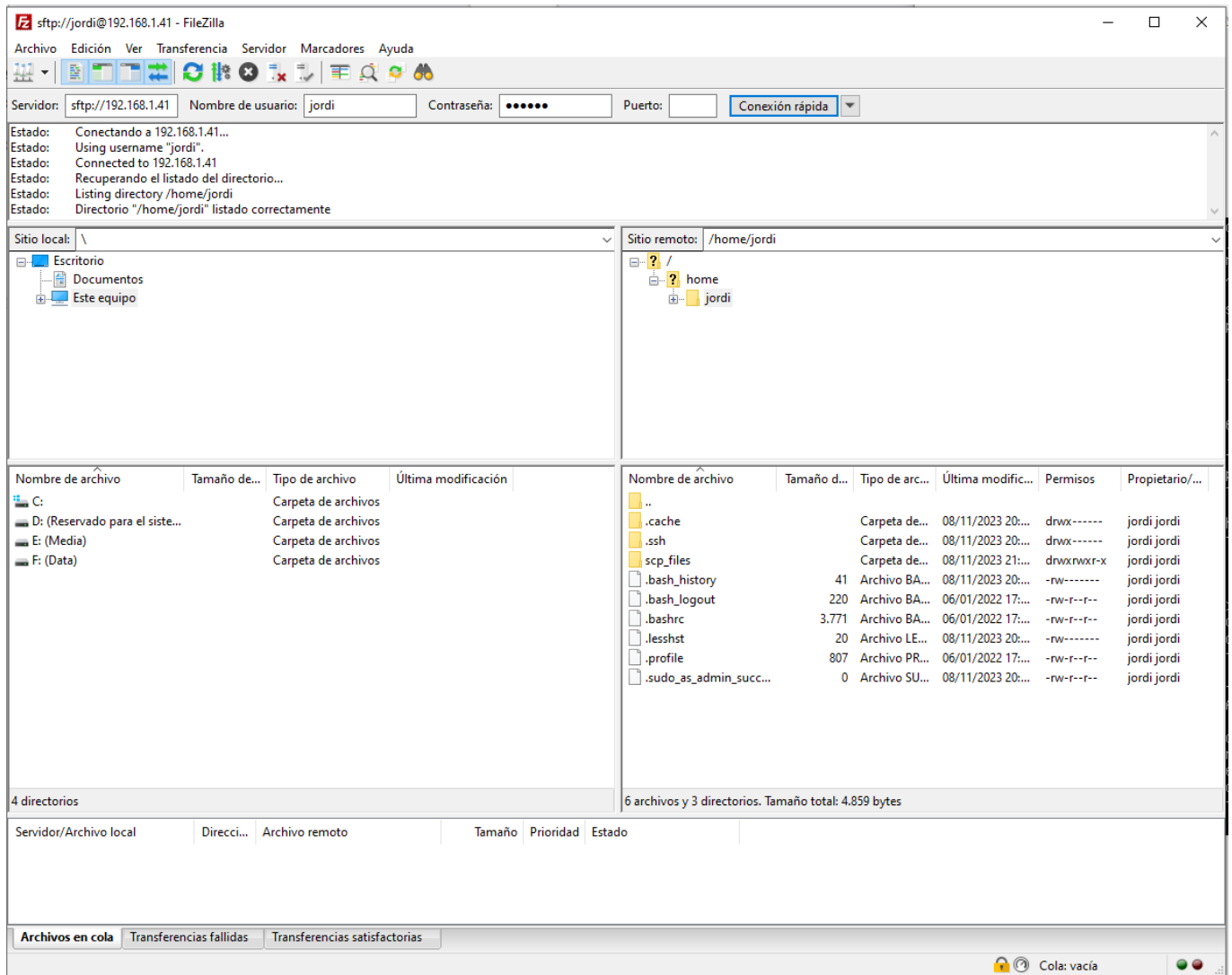
Com que no hem afegit aquesta màquina Ubuntu a les claus autoritzades del nostre servidor, ens demana la contrasenya.



Un cop introduïda la contrasenya, podem accedir als documents del servidor.



2. A windows feu servir els Filezilla per navegar pels fitxers del servidor
Primer, ens connectem al servidor introduint les dades del servidor per fer la connexió.



Un cop feta la connexió, podem navegar lliurement pels arxius de les carpetes del servidor de l'usuari seleccionat.



Nombre de archivo	Tamaño d...	Tipo de arc...	Última modific...	Permisos	Propietario/...
..					
DSC_0245.JPG	1.602.521	Archivo JPG	11/11/2023 12:...	-rw-rw-r--	jordi jordi
DSC_0246.JPG	1.600.778	Archivo JPG	11/11/2023 12:...	-rw-rw-r--	jordi jordi
DSC_0247.JPG	1.620.929	Archivo JPG	11/11/2023 12:...	-rw-rw-r--	jordi jordi
DSC_0248.JPG	1.604.985	Archivo JPG	11/11/2023 12:...	-rw-rw-r--	jordi jordi
DSC_0249.JPG	1.589.601	Archivo JPG	11/11/2023 12:...	-rw-rw-r--	jordi jordi
DSC_0250.JPG	1.528.257	Archivo JPG	11/11/2023 12:...	-rw-rw-r--	jordi jordi
DSC_0251.JPG	1.628.347	Archivo JPG	11/11/2023 12:...	-rw-rw-r--	jordi jordi
DSC_0252.JPG	1.464.607	Archivo JPG	11/11/2023 12:...	-rw-rw-r--	jordi jordi
DSC_0253.JPG	1.597.271	Archivo JPG	11/11/2023 12:...	-rw-rw-r--	jordi jordi
DSC_0254.JPG	1.614.821	Archivo JPG	11/11/2023 12:...	-rw-rw-r--	jordi jordi
DSC_0255.JPG	1.632.787	Archivo JPG	11/11/2023 12:...	-rw-rw-r--	jordi jordi
DSC_0256.JPG	1.714.267	Archivo JPG	11/11/2023 12:...	-rw-rw-r--	jordi jordi
DSC_0257.JPG	1.612.291	Archivo JPG	11/11/2023 12:...	-rw-rw-r--	jordi jordi

87 archivos. Tamaño total: 137.955.475 bytes

3. Mode línia, connecteu-vos amb sftp, i mostreu fitxers remots.
Fem la connexió.

```
C:\Users\Jordi Rissech>ssh jordi@192.168.1.41
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-88-generic x86_64)
```

Podem veure els arxius del servidor.

```
jordi@server:~$ ls -l
total 12
drwxrwxr-x 2 jordi jordi 4096 nov  4 11:32 documents
drwxrwxr-x 2 jordi jordi 4096 nov  4 11:36 images
drwxrwxr-x 3 jordi jordi 4096 nov  8 20:02 scp_files
jordi@server:~$
```

Podem visualitzar-los i manipular-los.

```
jordi@server:~$ cd documents
jordi@server:~/documents$ ls -l
total 4
-rw-rw-r-- 1 jordi jordi 29 nov  4 11:32 document.txt
jordi@server:~/documents$ cat document.txt
very important document text
```

```
jordi@server:~/documents$ cd ..
jordi@server:~$ cd images
```

```
jordi@server:~/images$ ls -l
total 134896
-rw-rw-r-- 1 jordi jordi 1602521 nov  4 11:36 DSC_0245.JPG
-rw-rw-r-- 1 jordi jordi 1600778 nov  4 11:36 DSC_0246.JPG
-rw-rw-r-- 1 jordi jordi 1620929 nov  4 11:36 DSC_0247.JPG
-rw-rw-r-- 1 jordi jordi 1604985 nov  4 11:36 DSC_0248.JPG
-rw-rw-r-- 1 jordi jordi 1589601 nov  4 11:36 DSC_0249.JPG
```



```
jordi@server:~/imagei$ rm DSC_0331.JPG
jordi@server:~/imagei$ ls -l
total 133344
-rw-rw-r-- 1 jordi jordi 1600778 nov  4 11:36 DSC_0246.JPG
-rw-rw-r-- 1 jordi jordi 1620929 nov  4 11:36 DSC_0247.JPG
-rw-rw-r-- 1 jordi jordi 1604985 nov  4 11:36 DSC_0248.JPG
-rw-rw-r-- 1 jordi jordi 1589601 nov  4 11:36 DSC_0249.JPG
```



Part 4 – scp (còpia segura)

El SSH també serveix per fer còpia segura de fitxers. Realitza:

1. Utilitza la comanda "scp" per transferir arxius del client al servidor

Ens desplacem a la carpeta **Documents/arxius_de_prova**, on farem les transferències d'arxius..

```
C:\Users\Jordi Rissech>cd Documents  
C:\Users\Jordi Rissech\Documents>cd arxius_de_prova
```

Enviem l'arxiu **arxiu_de_prova.txt**.

```
C:\Users\Jordi Rissech\Documents\arxius_de_prova>scp arxiu_de_prova.txt jordi@192.168.1.41:~/scp_files/arxiu_de_prova.txt  
arxiu_de_prova.txt 100% 7 7.1KB/s 00:00
```

2. Utilitza la comanda "scp" per transferir arxius del servidor al client.

Comprovem el text de l'arxiu **~/scp_files/arxiu_de_prova_2.txt**, present al servidor.

```
jordi@server:~$ cd scp_files  
jordi@server:~/scp_files$ cat arxiu_de_prova_2.txt  
hola què tal  
jordi@server:~/scp_files$
```

El copiem al client amb **scp**.

```
C:\Users\Jordi Rissech\Documents\arxius_de_prova>scp jordi@192.168.1.41:~/scp_files/arxiu_de_prova_2.txt arxiu_de_prova_2.txt  
arxiu_de_prova_2.txt 100% 14 14.0KB/s 00:00
```

Finalment, comprovem el contingut de l'arxiu per confirmar que s'hagi copiat correctament.

```
C:\Users\Jordi Rissech\Documents\arxius_de_prova>type arxiu_de_prova_2.txt  
hola què tal
```

3. Cerca la manera de pujar de cop una carpeta amb subcarpetes i arxius

La carpeta, ara en conté una altra anomenada **dir_de_prova** que, al seu torn, conté els arxius de prova.

```
C:\Users\Jordi Rissech\Documents\arxius_de_prova>dir  
El volumen de la unidad C no tiene etiqueta.  
El número de serie del volumen es: B2D2-305A  
  
Directorio de C:\Users\Jordi Rissech\Documents\arxius_de_prova  
08/11/2023 20:58 <DIR> .  
08/11/2023 20:58 <DIR> ..  
08/11/2023 20:46 7 arxiu_de_prova.txt  
08/11/2023 20:56 14 arxiu_de_prova_2.txt  
08/11/2023 20:58 <DIR> dir_de_prova  
2 archivos 21 bytes  
3 dirs 106.441.596.928 bytes libres
```

Ens desplacem a Documents i copiem **de forma recursiva** amb **scp -rp** la carpeta **arxius_de_prova**. El paràmetre **-p** preserva algunes metadades extra dels arxius originals i les copia també.

```
C:\Users\Jordi Rissech\Documents\arxius_de_prova>cd ..  
C:\Users\Jordi Rissech\Documents>scp -rp arxius_de_prova jordi@192.168.1.41:~/scp_files/  
arxiu_de_prova.txt 100% 7 3.5KB/s 00:00  
arxiu_de_prova_2.txt 100% 14 14.4KB/s 00:00  
arxiu_de_prova.txt 100% 7 0.0KB/s 00:00  
arxiu_de_prova_2.txt 100% 14 13.9KB/s 00:00
```

Ara, ens connectem al servidor amb **ssh** i, mitjançant la comanda **tree**, comprovem que s'ha fet correctament la còpia recursiva.



```
jordi@server:~/scp_files$ tree
.
├── arxiu_de_prova_2.txt
├── arxiu_de_prova.txt
├── arxius de prova
│   ├── arxiu_de_prova_2.txt
│   ├── arxiu_de_prova.txt
│   └── dir de prova
│       ├── arxiu_de_prova_2.txt
│       └── arxiu_de_prova.txt
2 directories, 6 files
```


Part 5 - Configuració del servidor

5.1 Canvi de port

Els servidors reben molts atacs i el port 22 és el primer de la llista. Canvia el port del servidor al nº 1022.

Per canviar el port del nostre servidor, hem d'editar l'arxiu `/etc/ssh/sshd_config`.

```
jordi@server:~$ sudo nano /etc/ssh/sshd_config
```

```
GNU nano 6.2 /etc/ssh/sshd_config *

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 1022
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#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

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo
```



- Quina comanda has de fer servir ara per connectar-te?

Per poder-nos connectar amb algun port que no sigui el predeterminat, haurem d'utilitzar la condició **-p**. Fem servir la comanda **ssh jordi@192.168.1.41 -p 1022**

```
C:\Users\Jordi Rissech>ssh jordi@192.168.1.41 -p 1022
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-88-generic x86_64)

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

System information as of mié  8 nov 2023 22:29:57 UTC

System load:  0.0               Processes:            114
Usage of /:   53.5% of 9.75GB   Users logged in:     1
Memory usage: 18%              IPv4 address for enp0s3: 192.168.1.41
Swap usage:   0%

 * 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

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 33 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: Mié Nov  8 21:40:53 2023 from 192.168.1.47
jordi@server:~$
```

- Com faràs ara un SCP?

Pel què fa a les transferències scp, haurem d'especificar també el port mitjançant una condició, que serà **-P** (-p té una altra funció).

scp -P 1022 arxiu_de_prova.txt jordi@192.168.1.41:~/scp_files/port1022/arxiu_de_prova.txt

```
C:\Users\Jordi Rissech\Documents\arxiu_de_prova>scp -P 1022 arxiu_de_prova.txt jordi@192.168.1.41:~/scp_files/port1022/arxiu_de_prova.txt
arxiu_de_prova.txt
100% 7 3.5KB/s 00:00
```

5.2 - Restriccions per usuari

Restringeix el servidor per tal que l'usuari "funky" NO es pugui connectar però sí l'usuari "james" (caldrà crear-lo si no existeix).

Dóna permisos de superusuari a "james" posant-lo al grup *sudo* (en versions anteriors d'Ubuntu és el grup *admin*).

Per especificar quins usuaris poden connectar-se via **ssh**, hem d'editar de nou l'arxiu **/etc/ssh/sshd_config**.

Posarem la condició **AllowUsers**, seguida dels usuaris que estaran autoritzats a utilitzar connexions ssh. Si introduïm aquesta opció, només els usuaris que hi especifiquem podran connectar-se al servidor via ssh. Per tant, si **no** volem que l'usuari **funky** es pugui connectar, simplement no l'hem d'incloure a la llista d'usuaris de l'opció.

```
GNU nano 6.2 /etc/ssh/sshd_config *
# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

LoginGraceTime 30
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
AllowUsers jordi james

#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

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo
```

Reiniciem el servidor.

```
jordi@server:~$ sudo systemctl restart sshd
[sudo] password for jordi:
```

Per poder utilitzar l'usuari **james**, primer l'hem d'haver creat. Per fer-ho, utilitzem la comanda **sudo adduser james**.

Un cop creat l'usuari, l'afegim al grup de sudoers amb la comanda **sudo usermod -aG sudo james**.

```
jordi@server:~$ sudo usermod -aG sudo james
```



Finalment, ens podem connectar com a **james**.

```
C:\Users\Jordi Rissech>ssh james@192.168.1.41 -p 1022
james@192.168.1.41's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-88-generic x86_64)

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

System information as of mié  8 nov 2023 22:42:31 UTC

System load:  0.0               Processes:            121
Usage of /:   53.6% of 9.75GB   Users logged in:     1
Memory usage: 18%              IPv4 address for enp0s3: 192.168.1.41
Swap usage:   0%

 * 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

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 33 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»

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

james@server:~$
```

.El propi terminal ens informa que podem utilitzar **sudo** per executar comandes com a administrador.

5.3 - Restriccions per IP

Restringeix el servidor per tal que les màquines de la xarxa interna NO es puguin connectar.

En [aquest link](#) indica diverses maneres d'assegurar les connexions SSH. Utilitzeu DenyHosts per restringir per les IPs.

Si no tenies xarxa interna, crea la interfície i configura el client correctament i comprova que fa pings però no connecta per SSH i sí ho deixa fer des de la màquina amfitriona.

SSH protocolo

SSH permite a los usuarios iniciar sesión en servidores remotamente. A diferencia de protocolos como FTP o Telnet, SSH encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas sin cifrar.

SSH ha sido diseñado para reemplazar a otros protocolos **telnet** o **rsh** más antiguos y menos seguros para conectarse a máquinas remotamente.

A través del protocolo ssh podemos realizar copias de ficheros seguras con el comando **scp** y transferencias de archivos seguras con el **sftp**, o sincronizar carpetas locales y remotas con **rsync**

Características del SSH

SSH (Secure Shell) o interprete de ordenes seguro es un protocolo que nos facilita las comunicaciones entre 2 siste,as usando la arquitectura cliente / servidor.

El protocolo SSH proporciona los siguientes tipos de protección:

- Después de la conexión inicial, el cliente puede verificar que está conectado al mismo servidor que estaba anteriormente.
- El cliente transmite su información de autenticación utilizando encriptación de 128 bits
- Todos los datos enviados y recibidos se transmiten utilizando encriptación de 128 bits.

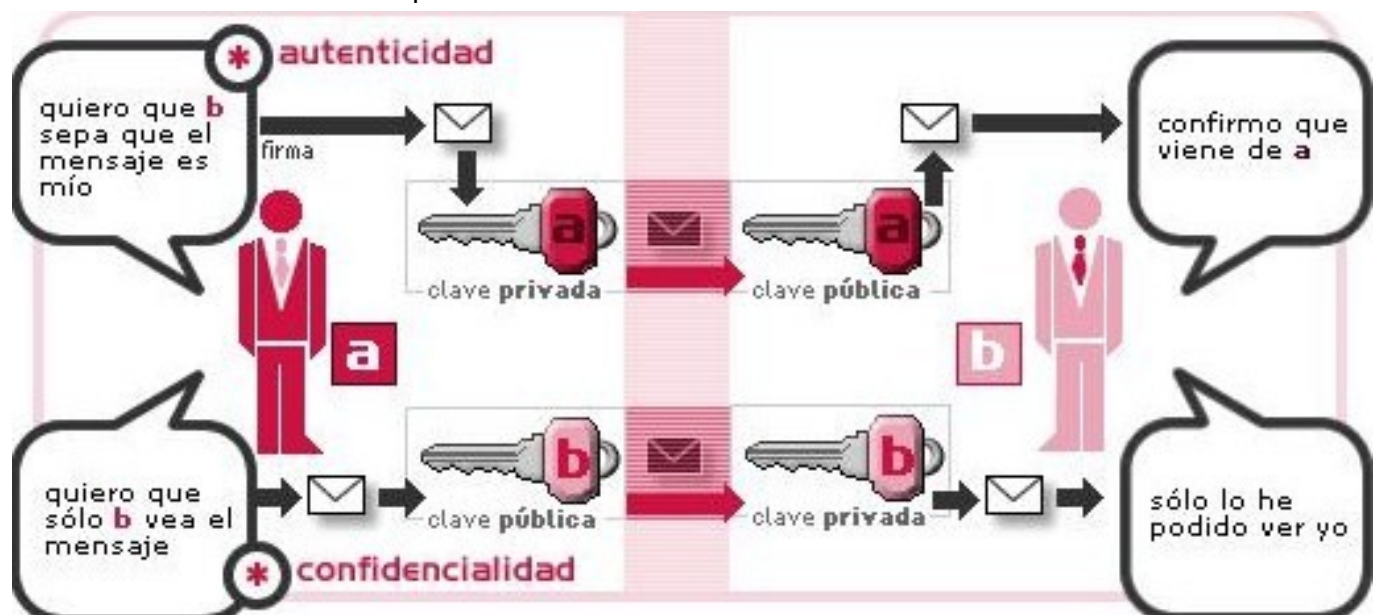
¿Por qué usar SSH?

En una red los datos que se transmiten son inseguros ya que cualquiera que esté conectado a la red puede capturar el tráfico que se genera entre 2 máquinas, por eso es importante encriptar la información de autenticación y los datos que se transmiten si se quiere mantener la privacidad sobre ellos.

Encriptación llaves privadas y llaves públicas

Criptografía Asimétrica: Este mecanismo utiliza dos claves: Una de las llaves será pública, podrá ser conocida por todos, y otra, que será privada, deberá estar custodiada por su propietario.

- Autenticidad : Firma digital para verificar que quien lo envia es quie dice que es.
- Confidencialidad : Encriptación de los datos.





Servidor SSH

Iniciar el servicio del servidor ssh

```
root:/root# /etc/init.d/ssh start
```

Starting OpenBSD Secure Shell server: sshd.

Para la configuración de parámetros del servidor tenemos que editar el siguiente fichero:

```
/etc/ssh/sshd_config
```

Para editarlo desde entorno gráfico

```
root:/root # gedit /etc/ssh/sshd_config
```

SSH usa por defecto el puerto 22. Esto significa que si no lo cambiamos estamos entregando a un caco que sabe la dirección de dónde vivimos (nuestra IP) también la llave del portal.

Cambiaremos el puerto para evitarlo. Esto no quita que el caco pueda intentar averiguar “el portal” si sabe cómo hacerlo pero al menos le ponemos impedimentos. También hay scripts que atacan directamente el puerto 22, por lo que el cambio de puerto es algo obligatorio. Poned el que queráis y abridlo también en el router para que podáis acceder a vuestro ordenador desde otro. Usaremos por ejemplo el 4321, podéis poner el que queráis. Así pues en el fichero de configuración:

```
port 4321
```

Un poco más abajo buscad la opción “Protocol” debe estar a valor 2, si no es así (valor 1 ó 2, ponedla. Hay dos versiones de protocolo SSH. La primera está ya en desuso y tiene varias vulnerabilidades. Así debéis dejarlo en vuestra configuración:

```
Protocol 2
```

Buscad la sección “Authentication”. Sus dos primeras opciones son también importantes. La primera es el número de segundos que tendrá el usuario remoto para hacer login en tu máquina. Poned ese valor a pocos segundos, no tardamos mucho en hacer login si sabemos la cuenta y la password. De esta forma evitamos ciertos scripts que se aprovechan de ese tiempo. El valor típico en términos de seguridad es 30, aunque podéis poner incluso menos si estáis más conformes.

```
LoginGraceTime 30
```

Justo debajo tenéis otras de las opciones más importantes, PermitRootLogin. Si antes usé la metáfora del caco y el portal, esta opción viene a ser que le digáis también en qué planta del bloque de pisos vivís y qué puerta, faltándole sólo la llave. Con esto lo que insinúo es que si sabe por qué puerto entrar, tan sólo le queda averiguar dos datos: el nombre de una cuenta y su contraseña.

Si tenemos esta opción habilitada (yes) el caco ya tiene la mitad del trabajo hecho, pues el usuario “root” existe en todas las máquinas GNU/Linux, tan sólo le queda averiguar la contraseña. Por eso es más que recomendable deshabilitar esta opción. No os preocupéis los que tenéis en mente usar SSH para hacer un uso administrativo, podéis hacerlo con vuestra cuenta y sudo sin problema alguno. Así pues...

```
PermitRootLogin no
```

También podéis señalar con el dedo las cuentas que tienen permitido el uso SSH (AllowUsers).

Pongamos un ejemplo, que es como mejor se entienden las cosas: Supongamos que tienes un amigo con el que quieres compartir algo vía SSH y además tiene un hermano que es un enreda y en el que no



confías por si te la puede liar. Llamaremos a las cuentas “amigo” y “pesado” respectivamente. Para restringir el uso de SSH a tu amigo y a tu propia cuenta (llamémosla “pepino”) podemos indicárselo mediante configuración. Incluso podemos indicar también que tu amigo sólo se pueda conectar a tu ordenador desde el suyo, sabiendo su IP (supongamos que es 83.45.258.21). Pondríamos en la configuración:

```
AllowUsers pepino amigo@83.45.258.21
```

De esta forma tú podrías usar tu cuenta (pepino) para conectar a tu equipo desde cualquier lugar, tu amigo podría hacerlo sólo desde su ordenador (si tiene esa IP) y tu hermano no podría conectar a tu máquina vía SSH, si no tiene tu cuenta.

Otra opción interesante es el número de intentos que tiene el usuario remoto para hacer login (MaxAuthTries). Como comenté antes, quien intente conectar debe acordarse de su login y password, por lo que es tontería darle un número grande de intentos. En principio con dos son más que suficientes. Si al segundo intento no lo ha conseguido se cortará la conexión SSH. Siempre se puede volver a conectar y reintentarlo, pero así nos quitamos de encima ciertos scripts que intentan encontrar el login por fuerza bruta a base de ensayo y error.

```
MaxAuthTries 2
```

Por último hay otra opción que define el número máximo de usuarios conectados simultáneamente a tu máquina. Esto ha de adaptarse a tus propias necesidades. Si estamos hablando de un ordenador personal donde sólo vas a conectar tú, pues lo lógico sería que como mucho hubiera una. Si estamos hablando de un ordenador que hará las veces de servidor compartiendo una carpeta a varias máquinas, deberás decidir cuántos son.

```
MaxStartups 2
```

Para que tengan efecto los cambios realizados en el servidor ssh hay que reiniciar el servicio

```
root:/root # /etc/init.d/ssh restart
```

Restarting OpenBSD Secure Shell server: sshd.

Cliente SSH

Conexión al servidor.

```
$ ssh usuario_remoto@host_remoto
```

Enviar u obtener archivos y carpetas con scp

Con el comando scp podemos realizar transferencias remotas seguras de archivos a través del protocolo SSH.

En una consola o terminal tecleamos:

```
$ scp -r usuario@maquina:/home/carpeta .
```

La opción “-r” significa recursivo, es decir, copia la carpeta y todo su contenido, incluidas las subcarpetas y el contenido de éstas.

Si no lo ponemos la orden para copiar todos los archivos de una carpeta sería:

```
$ scp usuario@maquina:/home/carpeta/* .
```

Si lo que queremos es enviar una carpeta con su contenido, utilizaremos la orden:

```
$ scp /home/carpeta/* usuario@maquina:/carpeta/
```