SSH

Índex

Configuració de SSH

OBJECTIUS: Instal·lar i configurar l'accés remot entre dues màquines mitjançant el protocol SSH

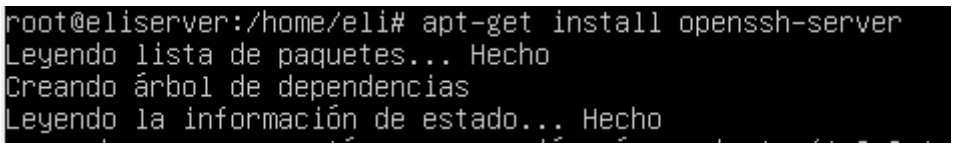
MATERIALS: Ubuntu Server i Ubuntu Desktop

Instal·lació de SSH

1. Instal·lació d'SSH en la màquina servidor. SSH utilitza una arquitectura client-servidor, en què el client es connecta a una màquina remota, el servidor. En la major part de distribucions GNU/Linux el client ja hi és però, si es vol accedir a una màquina de manera remota, en aquesta màquina hi haurà d'haver el servidor SSH instal·lat.

La implementació més popular d'SSH és la desenvolupada per la fundació OpenSSH, el servidor openSSH-server. Per procedir a instal·lar-la, executem la comanda següent:

#apt-get install openSSH-server



L'última part de la configuració bàsica es fa de manera automàtica quan intentem connectar un client per primera vegada. És la generació automàtica de la clau compartida que utilitzaran client i servidor per assegurar que les comunicacions són segures. Un cop s'ha comunicat la clau compartida entre totes dues estacions, el missatge s'encripta de forma convencional.

2. Configuració del client SSH. Arxius de configuració del client SSH

Per procedir a instal·lar-la, executem la comanda següent

#apt-get install openssh-client

```
root@eliclone:/home/usuarifinaleli# apt-get install openssh-client
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

El client d'OpenSSH es pot configurar de manera prou flexible com perquè l'administrador pugui definir una configuració general per a tot el sistema, perquè cada usuari pugui modificar els paràmetres adients per a les seves connexions o perquè pugui especificar opcions determinades per a cada connexió individual. La configuració del client d'OpenSSH és força flexible.

El client d'OpenSSH farà servir:

1. Les opcions indicades a la línia d'ordres
2. Els valors especificats en el fitxer de configuració de l'usuari: *$HOME/.ssh/ssh_config*
3. Els valors especificats en la configuració per a tot el sistema: */etc/ssh/ ssh_config*

Recordeu que *$HOME* és una variable d'entorn de Linux que conté el camí absolut del directori personal de l'usuari actiu.

Per a cada paràmetre, el client farà servir el primer valor trobat. És a dir, si s'especifica un paràmetre a la línia d'ordres no se'n consultarà el valor en els fitxers de configuració. Dins dels fitxers de configuració és possible definir seccions per a diferents equips (mitjançant la paraula reservada host). Les línies buides i les que comencen amb un # (comentari) seran ignorades.

Algunes de les opcions més bàsiques de la configuració d'un client

Opció	Funció
Host <patró>	Permet especificar opcions que només s'aplicaran a les connexions amb l'amfitrió indicat. L'amfitrió s'indica mitjançant patrons amb els caràcters * i ?. Especifica el port de destinació per a la connexió, que per defecte és el 22.
CheckHostIP <yes|no>	El seu valor predeterminat és *yes*. Si l'opció està activada, es comprovarà l'adreça de l'estació remota mitjançant el fitxer *known_hosts* per tal d'advertir un possible enverinament de DNS.

Opció	Funció
Cipher i Chipers	Permeten especificar respectivament l'algorisme d'encriptació per les a connexions SSH1 i la precedència d'algorismes que cal emprar en les connexions SSH2.
Compression <yes|no>	Si la connexió és molt lenta, la compressió pot millorar els resultats. Si la xarxa té prou amplada de banda normalment no es recomana.
Port <port>	Especifica el port de destinació per a la connexió, que de manera predeterminada és el 22.
RekeyLimit <limit>	Especifica el volum màxim d'informació que es pot transmetre abans d'haver de renegociar la clau de sessió. Es poden fer servir els sufixos K, M o G.
User <usuari>	Especifica l'usuari per establir la connexió a l'estació remota.
SendEnv <variables>	Permet enviar el valor de les variables d'entorn especificades a l'estació remota.

Configureu el fitxer ~~*$HOME/.ssh/ssh_config*~~ ***/etc/ssh/ssh_config*** del client perquè pugi fer una connexió amb el servidor.

L'arxiu ssh_config per l'usuari l'hauriem de crear manualment, ja que per defecte només trobem el que aplica a tot el sistema, que es troba a /etc/ssh/ssh_config:

```
usuarifinaleli@eliclone:~$ vim /etc/ssh/ssh_config
```

```
# This is the ssh client system-wide configuration file.  See
# ssh_config(5) for more information.  This file provides defaults for
# users, and the values can be changed in per-user configuration files
# or on the command line.

# Configuration data is parsed as follows:
#  1. command line options
#  2. user-specific file
#  3. system-wide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.

# Site-wide defaults for some commonly used options.  For a comprehensive
# list of available options, their meanings and defaults, please see the
# ssh_config(5) man page.

Host *
#   ForwardAgent no
#   ForwardX11 no
#   ForwardX11Trusted yes
#   PasswordAuthentication yes
#   HostbasedAuthentication no
#   GSSAPIAuthentication no
#   GSSAPIDelegateCredentials no
#   GSSAPIKeyExchange no
                                                              1,0-1     Comienzo
```

Sense necessitat de modificar res, podem fer connexió SSH al servidor:

```
usuarifinaleli@eliclone:~/.ssh$ ssh eli@172.100.100.1
The authenticity of host '172.100.100.1 (172.100.100.1)' can't be established.
ECDSA key fingerprint is SHA256:Bh49eR2dEQJxFooLBwfsbGkzcT2v+b7L2II2ZO1LSHU.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.100.100.1' (ECDSA) to the list of known hosts.
eli@172.100.100.1's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-100-generic x86_64)

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

  System information as of vie 04 mar 2022 15:27:34 UTC

  System load:  0.0               Processes:               130
  Usage of /:   28.6% of 8.79GB   Users logged in:         1
  Memory usage: 6%                IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%                IPv4 address for enp0s8: 172.100.100.1

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

17 updates can be applied immediately.
16 of these updates are standard security updates.
```

```
17 updates can be applied immediately.
16 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


Last login: Fri Mar  4 15:11:35 2022
eli@eliserver:~$
```

```
eli@eliserver:~$ hostname
eliserver
eli@eliserver:~$ exit
logout
Connection to 172.100.100.1 closed.
usuarifinaleli@eliclone:~/.ssh$ hostname
eliclone
usuarifinaleli@eliclone:~/.ssh$
```

Podem copiar el fitxer general al home de l'usuari per si hem de fer canvis:

```
usuarifinaleli@eliclone:~$ cp /etc/ssh/ssh_config $HOME/.ssh/ssh_config
```

3. Arxius de configuració del servidor SSH. La funció del servidor SSH és esperar les connexions dels clients (normalment al port TCP 22), dur a terme la seva autenticació i, si tot ha anat bé, obrir una sessió de treball, executar una ordre o bé redreçar ports.

Malgrat que en el funcionament del servidor d'OpenSSH hi intervenen diversos fitxers, l'arxiu principal de configuració és */etc/ssh/sshd_config*. Aquest fitxer conté diferents paraules clau amb el seu valor. Les línies que comencen amb # (comentaris) o les que estan en blanc són ignorades.

Si, el trobem a /etc/ssh del server:

```
root@eliserver:/home/eli# vim /etc/ssh/ssh_config
```

```
# This is the ssh client system-wide configuration file.  See
# ssh_config(5) for more information.  This file provides defaults for
# users, and the values can be changed in per-user configuration files
# or on the command line.

# Configuration data is parsed as follows:
#  1. command line options
#  2. user-specific file
#  3. system-wide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.

# Site-wide defaults for some commonly used options.  For a comprehensive
# list of available options, their meanings and defaults, please see the
# ssh_config(5) man page.

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

Host *
#   ForwardAgent no
#   ForwardX11 no
#   ForwardX11Trusted yes
#   PasswordAuthentication yes
#   HostbasedAuthentication no
#   GSSAPIAuthentication no
#   GSSAPIDelegateCredentials no
#   GSSAPIKeyExchange no
```

Paràmetres de configuració del servidor SSH

Opció	Funció
AllowGroups	Si s'especifica seguida d'una llista de grups (separats per espais), només els usuaris que tenen algun dels grups indicats com a grup principal o suplementari podran iniciar sessió. És possible emprar els patrons *?* i * en la definició dels grups. Només es poden indicar els grups mitjançant el seu nom, no en format numèric (GID).
AllowUsers	Té la mateixa funció que *AllowGroups*, però per als usuaris. En aquest cas, a més, és possible indicar des de quins amfitrions d'origen s'acceptarà la connexió. Per exemple: *AllowUsers usuari1@192* usuari2.*
Banner <fitxer>	Envia el contingut del fitxer indicat al client abans de dur a terme l'autenticació.
Compression <yes|delayed|no>	Especifica si es farà servir la compressió. El valor *delayed*, l'opció predeterminada, indica que només es farà servir la compressió un cop s'hagi autenticat l'usuari.
DenyGroups	Permet especificar una llista de grups, separats per espais, als quals no es permetrà iniciar sessió. Es poden especificar els grups mitjançant el seu nom, emprant els patrons *?* i * de manera opcional.
DenyUsers	Igual que *DenyGroups*, però per als usuaris. En aquest cas és possible indicar un equip (o subxarxa) per a cada usuari.

Opció	Funció
Port ListenAddress	Permeten especificar el port on el servidor escoltarà les connexions dels clients i les adreces on obrirà aquest port. De manera predeterminada s'utilitza el port 22 de qualsevol adreça local. És possible especificar múltiples vegades aquestes opcions, però convé que *Port* sempre aparegui abans que *ListenAddress*.
LoginGraceTime	Període de temps màxim per dur a terme l'autenticació. El valor 0 expressa que no hi ha límit.
MaxAuthTries	Nombre màxim d'intents d'autenticació que es poden fer.
MaxStartups	Nombre màxim de connexions simultànies que encara no han completat la seva autenticació.
PasswordsAuthenticati on <yes|no>	Indica si s'accepta l'autenticació mitjançant contrasenya (*password).*
PermitEmptyPasswords <no|yes>	Si s'utilitza l'autenticació mitjançant contrasenya, especifica si el servidor permet la connexió a comptes que tenen una contrasenya buida.
PermitRootLogin <yes|without-password|forced-commands-only|no>	Especifica si s'accepta la connexió de superusuari mitjançant SSH. El valor *without-password* indica que el superusuari no podrà fer servir l'autenticació basada en contrasenya i el valor *forced-commands-only*, que només es permetrà l'autenticació de clau pública per executar certes ordres de manera remota (normalment per fer còpies de seguretat).

Opció	Funció
Protocol	Especifica quins protocols es podran fer servir en les connexions dels clients (1, 2 o tots dos). És important recordar que el protocol SSH2 és força més segur que l'SSH1.
PubkeyAuthentication <yes|no>	Especifica si s'acceptarà l'autenticació de clau pública.
X11Forwarding <no|yes>	Especifica si s'acceptarà el reenviament X11 per tal que les aplicacions gràfiques executades en el servidor obrin la seva finestra en el servidor X del client.

1. Altres arxius de configuració

Hi ha altres arxius de configuració que permeten de forma genèrica el filtratge, control d'accés i mecanismes de protecció de diferents serveis (POP, Sendmail, Telnet, SSH, etc.) actuant de fet com un tallafocs bàsic.

Així, si volem habilitar o restringir l'accés a determinats equips i serveis podem editar els arxius de configuració */etc/hosts.deny* i */etc/hosts.allow* indicant en la directiva dintre de l'arxiu el servei que volem controlar, en aquest cas, el dimoni SSH.
Si, veiem els dos arxius.

```
# /etc/hosts.allow: list of hosts that are allowed to access the system.
#                   See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:    ALL: LOCAL @some_netgroup
#             ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
~
```

```
# /etc/hosts.deny: list of hosts that are _not_ allowed to access the system.
#                  See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:    ALL: some.host.name, .some.domain
#             ALL EXCEPT in.fingerd: other.host.name, .other.domain
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
# The PARANOID wildcard matches any host whose name does not match its
# address.
#
# You may wish to enable this to ensure any programs that don't
# validate looked up hostnames still leave understandable logs. In past
# versions of Debian this has been the default.
# ALL: PARANOID
```

D'aquesta manera el sistema, davant d'una petició d'accés al servei, fa la cerca següent, que conclou en el moment de la primera coincidència:

- Comprova l'arxiu */etc/hosts.allow*. Si hi troba coincidència valida l'accés.
- Comprova l'arxiu */etc/hosts/deny*. Si hi troba coincidència no valida l'accés.
- En cas de no trobar coincidència en cap dels arxius valida l'accés.

Exemples de directives d'aquests arxius es poden veure a continuació: #(permet/denega l'accés ssh a tothom)

sshd: ALL #(permet/denega l'accés SSH de la IP 192.168.56.10) sshd: 192.168.56.10 Recordeu que perquè qualsevol canvi tingui efecte s'ha de reiniciar el servei: *#service ssh restart*

Connexió remota amb SSH

OBJECTIUS: Fer una connexió remota entre un client i un servidor mitjançant SSH

MATERIALS: Ubuntu client i Ubuntu server.

La funció més comuna per a SSH és establir una sessió de treball remota fent ús de tècniques criptogràfiques per transmetre la informació. L'ús del client SSH és força senzill.

```
#ssh user@host
```

- user: és l'usuari que es connectarà a la màquina remota.
- host: representa la IP o el nom de domini del servidor SSH al qual ens volem connectar.

És tot el que hem d'escriure per iniciar una sessió remota com a usuari (user) en l'equip amfitrió (host). En executar l'ordre ens demanarà la contrasenya de l'equip remot i, si l'escrivim de manera correcta, podrem accedir a la sessió de treball remota per escriure ordres.

1) Fes una connexió amb ssh entre client i servidor

```
usuarifinaleli@eliclone:~$ ssh eli@172.100.100.1
eli@172.100.100.1's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-100-generic x86_64)

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

  System information as of vie 04 mar 2022 15:40:23 UTC

  System load:  0.4               Processes:               130
  Usage of /:   28.6% of 8.79GB   Users logged in:         1
  Memory usage: 6%                IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%                IPv4 address for enp0s8: 172.100.100.1

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

17 updates can be applied immediately.
16 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


Last login: Fri Mar  4 15:27:35 2022 from 172.100.100.2
```

2) Finalitza la connexió anterior mitjançant la comanda exit

```
eli@eliserver:~$ exit
logout
Connection to 172.100.100.1 closed.
```

3) Si no s'especifica el nom d'usuari a l'hora d'invocar l'ordre SSH, intentarà fer la connexió amb l'usuari amb el qual estem connectats al terminal de GNU/Linux. Prova-ho

Si, agafa l'usuari actual i no deixa connectar-nos.

```
usuarifinaleli@eliclone:~$ ssh 172.100.100.1
usuarifinaleli@172.100.100.1's password:
Permission denied, please try again.
```

4) El client d'OpenSSH disposa de diferents opcions que es detallen en el seu manual. Algunes de les més freqüents són les descrites a la taula

ó	
-1	Força l'ús de la versió 1 del protocol SSH. Només es recomana emprar SSH1 per connectar-se a servidors antics que no són compatibles amb SSH2.
-2	Força l'ús de la versió 2 del protocol: SSH2.
-4	Força l'ús de l'adreçament IPv4.
-6	Força l'ús de l'adreçament Ipv6.
-C	Activa la compressió *gzip* en la connexió. Es recomana activar la compressió si s'està emprant SSH amb un enllaç lent, com un mòdem. Si l'enllaç és de banda ampla es recomana treballar sense compressió.
-p port	Port al qual es connectarà en l'equip remot. De manera predeterminada el servidor SSH s'executa al port TCP 22, però si es tracta d'un servidor accessible des d'Internet és recomanable escollir un altre port per evitar els intents de connexió.
-q	No imprimeix els missatges d'advertència, només els errors. Amb una altra *-q* no imprimeix ni els errors.
-X	Activa la retransmissió X11 per tal que els programes gràfics llançats en l'estació remota obrin la seva interfície gràfica en el servidor X local.
-x	Desactiva la retransmissió X11.

Connecteu-vos al vostre servidor i visualitzeu el final del fitxer de registre */var/log/messages* al servidor **172.100.100.10**

Al servidor no trobem els logs de /var/log/messages, però podem provar amb qualsevol altre fitxer de log i el podem llegir des de la connexió ssh del client :

```
eli@eliserver:/var/log$ ls
alternatives.log        dmesg.0      lastlog
alternatives.log.1      dmesg.1.gz   private
apache2                 dmesg.2.gz   syslog
apt                     dmesg.3.gz   syslog.1
auth.log                dmesg.4.gz   syslog.2.gz
auth.log.1              dpkg.log     ubuntu-advantage.log
bootstrap.log           dpkg.log.1   ubuntu-advantage.log.1
btmp                    faillog      ubuntu-advantage-timer.log
btmp.1                  installer    ubuntu-advantage-timer.log.1
cloud-init.log          journal      unattended-upgrades
cloud-init-output.log   kern.log     wtmp
dist-upgrade            kern.log.1
dmesg                   landscape
```

Per visualitzar les ultimes línies del fitxer, empreu la comanda tail

tail

És una instrucció de GNU/Linux que permet veure les 10 últimes línies. El paràmetre n permet especificar el nombre de línies que es vol mostrar. En aquest cas, se'n mostren tres.

```
eli@eliserver:~/.ssh$ tail /var/log/syslog
Mar  4 15:26:28 eliserver systemd[1]: Finished Cleanup of Temporary Directories
.
Mar  4 15:27:34 eliserver systemd[1]: Started Session 4 of user eli.
Mar  4 15:30:57 eliserver systemd[1]: session-4.scope: Succeeded.
Mar  4 15:39:00 eliserver systemd[1]: Starting Clean php session files...
Mar  4 15:39:00 eliserver systemd[1]: phpsessionclean.service: Succeeded.
Mar  4 15:39:00 eliserver systemd[1]: Finished Clean php session files.
Mar  4 15:39:03 eliserver CRON[2621]: (root) CMD (   [ -x /usr/lib/php/sessioncl
ean ] && if [ ! -d /run/systemd/system ]; then /usr/lib/php/sessionclean; fi)
Mar  4 15:40:23 eliserver systemd[1]: Started Session 6 of user eli.
Mar  4 15:40:50 eliserver systemd[1]: session-6.scope: Succeeded.
Mar  4 15:42:12 eliserver systemd[1]: Started Session 7 of user eli.
```

Al syslog (equivalent de messages) veiem logs sobre les nostres sessions ssh.

5) Configura el servidor per autentificar el usuari de manera automàtica. https://www.ssh.com/ssh/keygen/

Des del client s'ha de generar una clau rsa:

Acceptar que la guardi a $HOME/.ssh/id_rsa i enter dos cops sense posar passphrase.

```
usuarifinaleli@eliclone:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/usuarifinaleli/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/usuarifinaleli/.ssh/id_rsa.
Your public key has been saved in /home/usuarifinaleli/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:N9CJ7rLBkOgBtVOK1tcmiIZ28Z+gyjIzbYUWcwIpdog usuarifinaleli@eliclone
The key's randomart image is:
+---[RSA 2048]----+
|o.o..            |
|E*o=+ .  o .     |
|*=O+.= oo o      |
|+..Oo.=...       |
|  =.+  oS o      |
|.+.o o . . .     |
|=o+   + .        |
|.=     +         |
|      .          |
+----[SHA256]-----+
```

Al directori .ssh veiem que tenim dos fitxers. La clau pública (id_rsa.pub) i la clau privada (id_rsa)

```
usuarifinaleli@eliclone:~$ ll ~/.ssh/
total 24
drwx------  2 usuarifinaleli usuarifinaleli 4096 mar  4 16:55 ./
drwxr-xr-x 16 usuarifinaleli usuarifinaleli 4096 mar  4 16:29 ../
-rw-------  1 usuarifinaleli usuarifinaleli 1679 mar  4 16:55 id_rsa
-rw-r--r--  1 usuarifinaleli usuarifinaleli  405 mar  4 16:55 id_rsa.pub
-rw-r--r--  1 usuarifinaleli usuarifinaleli  222 mar  4 16:27 known_hosts
-rw-r--r--  1 usuarifinaleli usuarifinaleli 1580 mar  4 16:32 ssh_config
```

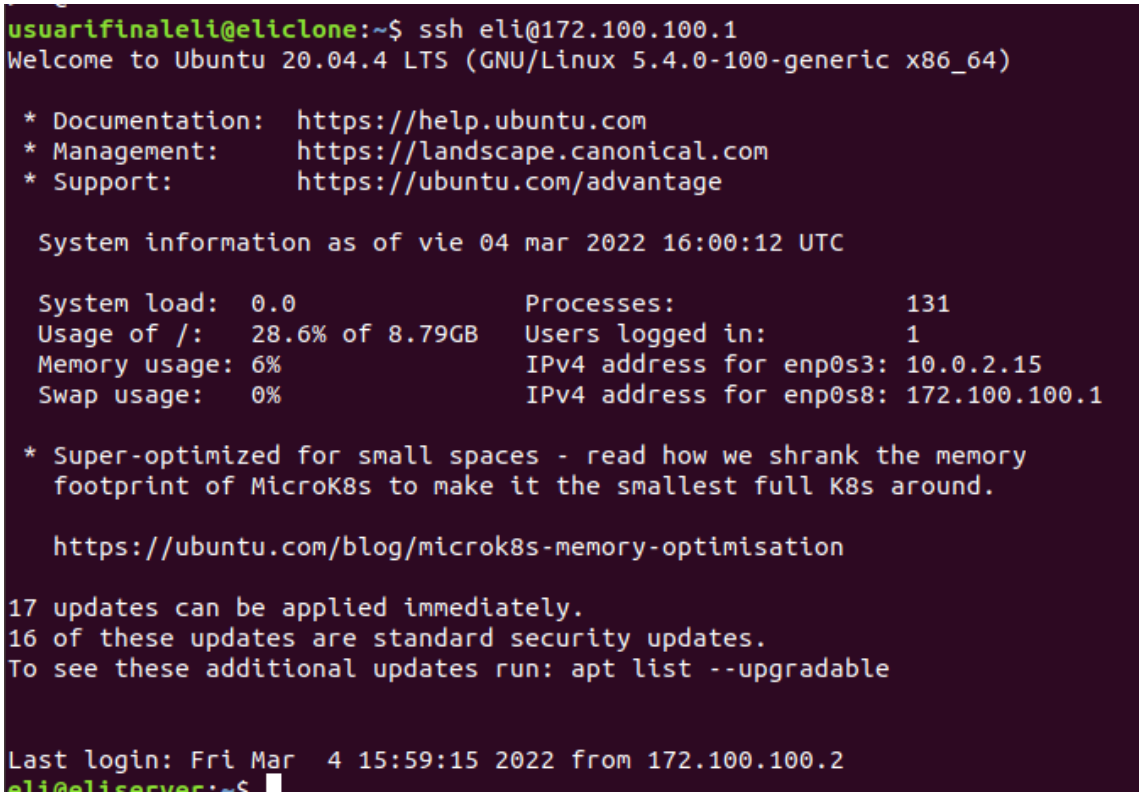
Per tal que autentiqui automàticament, hem de copiar la clau pública al servidor amb la següent comanda:

```
usuarifinaleli@eliclone:~$ ssh-copy-id -i ~/.ssh/id_rsa.pub eli@172.100.100.1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/usuarifina
leli/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
 out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are promp
ted now it is to install the new keys
eli@172.100.100.1's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'eli@172.100.100.1'"
and check to make sure that only the key(s) you wanted were added.
```

Efectivament, ara no demanarà més password al connectar-nos per ssh desde el client:



Sistemes de fitxers amb ssh: SSHFSe

OBJECTIUS: Algunes vegades necessitem treballar durant força temps amb un sistema remot, copiant i editant fitxer. En aquest exercici muntarem un sistema de fitxers mitjançant SSH – SSHFS

MATERIALS: Ubuntu Server i ubuntu Desktop. Podeu usar ubuntu server amb o sense interfície gràfica.

1. Es possible muntar un sistema de fitxers remot en el servidor mitjançant NFS i SAMBA, però també es pot fer amb SSH. Per muntar sistemes de fitxers per SSH farem ús de **SSHFS.**

Fuse és un mòdul del kenel que permet muntar diferents sistemes de fitxers amb un usuari normal sense privilegis. Ubuntu porta fuse instal·lat per defecte, de manera que no cal instal·lar-lo. El que sí has de fer és verificar que formes part del grup fuse. Pots agregar-te a aquest grup escrivint el terminal.

```
root@eliserver:/home/eli# groupadd profesors
root@eliserver:/home/eli# useradd jordi
root@eliserver:/home/eli# usermod -a -G profesors jordi
```

Crea un usuari anomenat jordi que pertany al grup professors. Després executa la següent comanda per afegir-lo al grup de fuse.

#gpasswd -a [usuari] fuse

```
root@eliserver:/home/eli# groupadd fuse
^[[A^[[B^[[Aroot@eliserver:/home/eli# gpasswd -a jordi fuse
Adding user jordi to group fuse
```

2. SSHFS és un programa creat per l'autor de fuse que permet muntar un directori remot usant SSH. Accedirem localment com si estigués en la nostra pròpia màquina. Cal instal·lar el programa sshfs al client. L'usuari que pot muntar el sistema de fitxers ha de pertànyer al grup fuse.

#apt-get install sshfs

```
root@eliclone:/home/usuarifinaleli# apt-get install sshfs
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
 son necesarios.
  fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0
  gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0
  grilo-plugins-0.3-base gstreamer1.0-gtk3 libboost-date-time1.65.1
```

3. Crear una carpeta al servidor anomenada remot. Aquest directori també l'heu de crear al client.

```
root@eliclone:~# mkdir remot
root@eliclone:~#
```

```
root@eliserver:/home/eli# cd /root/
root@eliserver:~# mkdir remot
root@eliserver:~#
```

4. A continuació munta la unitat remota amb la següent comanda

#sshfs usuari@ip_servidor:recurs_remot_servidor unitat_muntatje_client

Perque em funcionés he hagut de posar la opció nonempty (-o nonempty) perquè ja havia creat un fitxer al remot.

```
root@eliclone:/mnt/remot# sshfs -o nonempty eli@172.100.100.1:/home/eli/remot /
mnt/remot
eli@172.100.100.1's password:
root@eliclone:/mnt/remot# ls
EliLaMillor.txt  hola.txt
```

5. Ara comprova que l'usuari creat pot accedir a la unitat que hem muntat amb sshfs. Crea un fitxer al client i després comprova a la part del server que s'ha creat correctament. A aquest recurs remot soles poden accedir els usuaris que pertanyen al grup professors.

```
eli@eliserver:~/remot$ ls
EliLaMillor.txt  hola.txt
eli@eliserver:~/remot$
```

Faig mes proves del server a client un cop muntat:

```
root@eliserver:/home/eli/remot# touch eli
root@eliserver:/home/eli/remot# touch prova2
root@eliserver:/home/eli/remot#
```

Veiem que apareixen automàticament:

```
root@eliclone:/mnt/remot# ll
total 8
drwxr-xr-x 1 root root 4096 mar 10 18:45 ./
drwxr-xr-x 4 root root 4096 mar 10 18:34 ../
-rw-r--r-- 1 root root    0 mar 10 18:45 eli
-rw-r--r-- 1 root root    0 mar 10 18:39 EliLaMillor.txt
-rw-r--r-- 1 root root    0 mar 10 18:36 hola.txt
-rw-r--r-- 1 root root    0 mar 10 18:45 prova2
root@eliclone:/mnt/remot#
```

6. Si volem desmuntar la unitat d'intercanvi de fitxers, executem la comanda

#fusemount -u unitat_de_muntatge

```
root@eliclone:/mnt# fusermount -u remot
root@eliclone:/mnt#
```

7. Si treballarem diàriament amb aquest directori remot, potser és bona idea afegir-lo al fitxer /etc/fstab. D'aquesta manera es muntarà automàticament en iniciar el nostre ordinador o manualment (si triem l'opció noauto) sense necessitat d'especificar la localització remota cada vegada. Aquest és un exemple de configuració:

sshfs#usuari_remot@ip_server:/directori_remot /home/usuari/directori_remot fuse defaults,auto 0 0

```
                                    root@eliclone: /mnt
 Archivo  Editar  Ver  Buscar  Terminal  Ayuda
# /etc/fstab: static file system information.
#
```

```
sshfs#eli@172.100.100.1:/home/eli/remot /mnt/remot fuse defaults auto 0 0
```

8. Si anem a utilitzar fuse i sshfs regularment, hauries d'editar el fitxer /etc/modules i afegir el mòdul fuse. D'una altra manera hauràs de carregar el mòdul manualment cada vegada que ho vulguis fer servir:

   #sh -c "echo fuse >> /etc/modules"

```
root@eliclone:/mnt# sh -c "echo fuse >> /etc/modules"
root@eliclone:/mnt#
```

(aquesta part te la deixo ja que la vaig començar)

Administració remota

Programari d'accés remot TightVNC

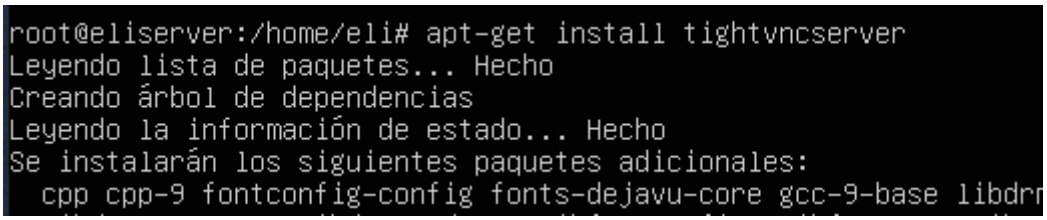
El programari d'accés remot TightVNC és una de les implementacions de VNC més avançades. Es distribueix amb llicència GPL, la qual cosa permet un accés lliure al seu codi font, a la seva distribució i a la seva instal·lació.

OBJECTIUS: Aconseguir l'accés remot entre en client i un servidor amb l'eina TightVNC

MATERIALS: Màquines Ubuntu Server i Ubuntu Desktop

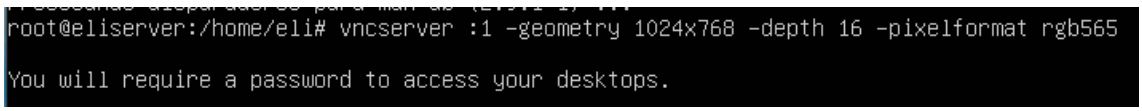
1. **Instal·lació del servidor TightVNC.** El paquet del servidor TightVNC s'anomena tightvncserver i per instal·lar-lo executeu la següent comanda:

   #apt-get install tightvncserver



2. Un cop instal·lat ja podem arrencar el servidor VNC, vncserver, juntament amb una sèrie d'opcions.

   #vncserver :1 -geometry 1024x768 -depth 16 -pixelformat rgb565



Les diferents opcions de la comanda anterior son:

- **:1**: indica la pantalla en la qual s'establirà la sessió per fer les connexions remotes.
- **-geometry**: estableix les dimensions de la finestra amb què es farà la connexió.

- **-depth**: estableix la profunditat dels colors en bits per píxel. Ha de ser un valor entre 8 i 32.
- **-pixelformat**: estableix el format de la representació dels píxels.

3. A continuació, el programari ens demanarà dues contrasenyes, la segona de les quals és opcional. La primera és la necessària per permetre l'accés total a l'equip servidor des de l'equip remot. La segona és per permetre només un accés de visualització de l'escriptori.

```
You will require a password to access your desktops.

Password:
Verify:
Would you like to enter a view-only password (y/n)? n
xauth:  file /root/.Xauthority does not exist

New 'X' desktop is eliserver:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/eliserver:1.log
```

Contrasenya

Atenció! Hi ha un límit de 8 caràcters per a la contrasenya de TightVNC.

4. Per aturar el servidor TightVNC es pot fer servir l'ordre *vncserver* i s'indica la pantalla en què s'ha invocat.

#vncserver -kill :1

On el resultat és:

Killing Xtightvnc process ID 4223

5. **Instal·lació del client TightVNC.** El paquet que permet instal·lar el client TightVNC es diu *xtightvncviewer*

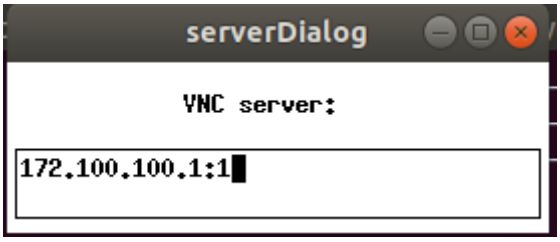
```
usuarifinaleli@eliclone:~$ sudo apt-get install xtightvncviewer
[sudo] contraseña para usuarifinaleli:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
 son necesarios.
  fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0
  gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0
```

6. A continuació, com que s'està executant el servidor de TightVNC, es pot fer la connexió des del client remot. S'ha d'indicar l'adreça IP del servidor i la pantalla on està funcionant el servidor. Posa la comanda i una captura de pantalla on es mostren les pantalles del client i del server.

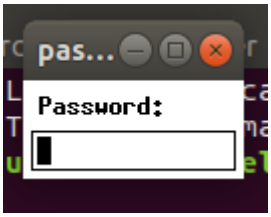
   Executar vncviewer al client:

```
usuarifinaleli@eliclone:~$ vncviewer
```

   S'obrirà un diàleg on s'ha d'afegir la IP i la pantalla on està funcionant.

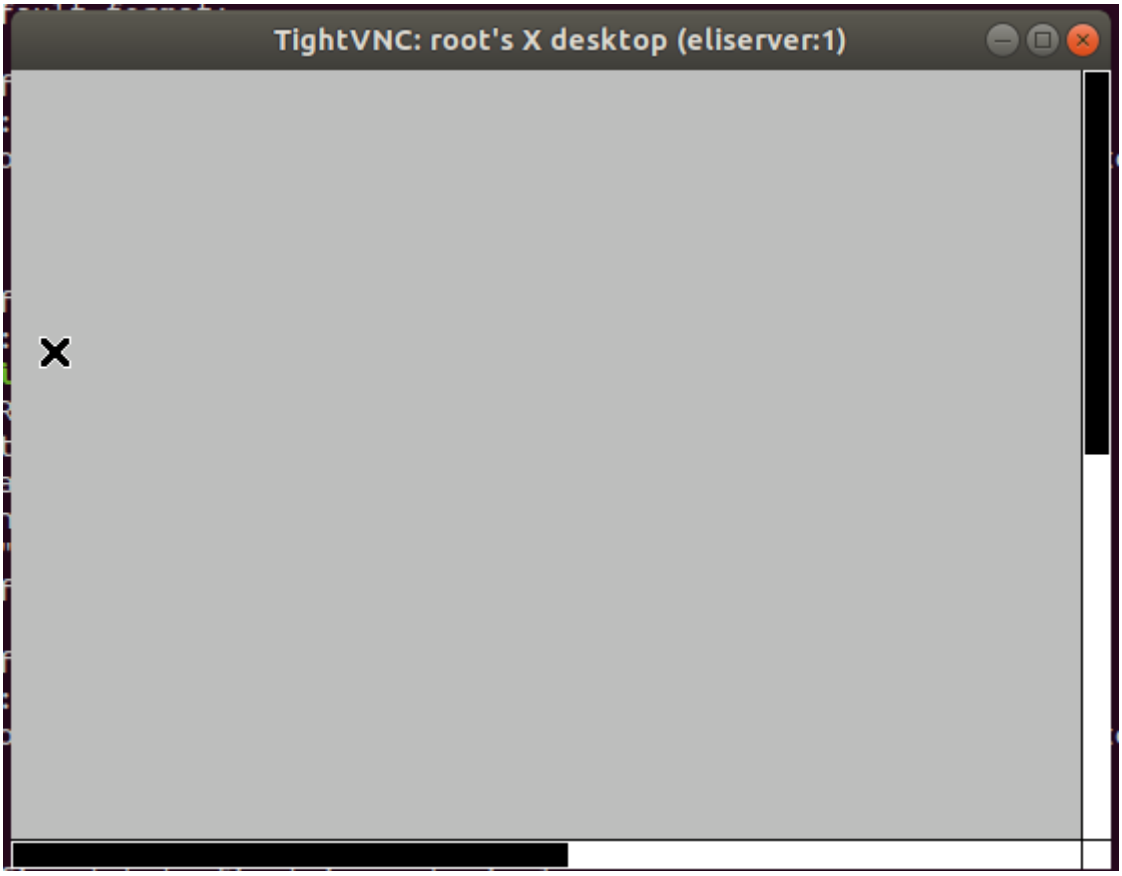


   A continuació demanarà la password:




```
usuarifinaleli@eliclone:~$ vncviewer
Connected to RFB server, using protocol version 3.8
Enabling TightVNC protocol extensions
Performing standard VNC authentication
Authentication successful
Desktop name "root's X desktop (eliserver:1)"
VNC server default format:
  16 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 31 green 63 blue 31, shift red 11 green 5 blue 0
Warning: Cannot convert string "-*-helvetica-bold-r-*-*-16-*-*-*-*-*-*-*" to ty
pe FontStruct
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
```

   Per algun motiu, la autenticació es fa correctament pel que veiem a la terminal. però la finestra de tightVNC es queda en gris.



7. **Accés remot per web i TightVNC**. L'accés remot es pot fer també mitjançant un navegador web que sigui compatible amb les miniaplicacions (applets) de Java, com Firefox.

Per poder accedir d'aquesta manera, el primer que s'ha de fer és instal·lar al servidor el component que permetrà aquest tipus d'accés.

 # apt-get install tightvnc-java

```
root@eliclone:/home/usuarifinaleli# apt-get install tightvnc-java
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
 son necesarios.
  fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0
  gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0
  grilo-plugins-0.3-base gstreamer1.0-gtk3 libboost-date-time1.65.1
  libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1
  libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcmis-0.5-5v5
  libcolamd2 libdazzle-1.0-0 libe-book-0.1-1 libedataserverui-1.2-2 libeot0
  libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14
```