

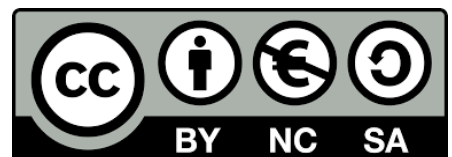


4. Web Server Administration

Miquel Àngel París i Peñaranda

Web Application Deployment

2nd C-VET Web Application Development



Index

SSH exercises3

SSH exercises

Practice 1: SSH

At the end of this practice, you must submit a script that explains and demonstrates the process and steps followed to complete each of the following tasks:

1. Start the SSH server on your machine and verify that you can access it with any user from any device.
2. Configure the SSH service so that:
 - Logging in as root is not allowed.
 - Then, allow logging in as root.
 - Create two new users: user2 and user3, and a group named ssh_users. Add these users to the group and ensure that only the members of this group can access the SSH service.
3. The SSH service, by default, listens on port 22. Modify it to start on port 10022 (or another port), and figure out the command required to access the server on this port. (*Hint: Use man ssh*)
4. By default, when authenticating successfully on the SSH server, it shows the date and time of the last connection. Find the option responsible for this behavior and modify it.
5. Configure the SSH server to enable X11 redirection, allowing the execution of graphical applications remotely. Test and verify its functionality.
6. Test connecting from a Windows SSH client (e.g., PuTTY) to a GNU/Linux SSH server and verify the connection.

Practice 1:

Exercise 1: Start the SSH server on your machine and verify that you can access it with any user from any device.

Server's machine: \$ systemctl start ssh && systemctl status ssh

```
user@server:~$ systemctl start ssh && systemctl status ssh
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'ssh.service'.
Authenticating as: user
Password:
==== AUTHENTICATION COMPLETE ====
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-01-13 14:43:28 UTC; 1min 34s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 629 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 658 (sshd)
    Tasks: 1 (limit: 4546)
   Memory: 3.2M
      CPU: 39ms
   CGroup: /system.slice/ssh.service
           └─658 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 13 14:43:28 server systemd[1]: Starting OpenBSD Secure Shell server...
Jan 13 14:43:28 server sshd[658]: Server listening on 0.0.0.0 port 22.
Jan 13 14:43:28 server sshd[658]: Server listening on :: port 22.
Jan 13 14:43:28 server systemd[1]: Started OpenBSD Secure Shell server.
```

Server's machine: \$ ssh user@192.168.18.251

```
user@server:~$ ssh user@192.168.18.251
The authenticity of host '192.168.18.251 (192.168.18.251)' can't be established.
ED25519 key fingerprint is SHA256:f45jifu0batm5N1R1xJI1Au6b09iuOpEdAfszzJq1/I.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.18.251' (ED25519) to the list of known hosts.
user@192.168.18.251's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-130-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Jan 13 14:44:01 2025
user@server:~$ _
```

Client's machine: \$ssh user@192.168.18.251

Unit 4. Web Server Administration

```
usuario@user:~$ ssh user@192.168.18.251
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-130-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Jan 13 14:47:09 2025 from 192.168.18.251
user@server:~$
```

Exercise 2: Configure the SSH service so that:

Exercise 2.1: Logging in as root is not allowed.

Server's machine(folder:'/etc/ssh/'): \$ sudo nano sshd_config

In this file, we must descomment the line 'PermitRootLogin' and change the value for 'no'

```
# Authentication:
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Server's machine: \$ sudo nano systemctl restart ssh && sudo systemctl status ssh

Unit 4. Web Server Administration

```
user@server:~$ systemctl start ssh && systemctl status ssh
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'ssh.service'.
Authenticating as: user
Password:
==== AUTHENTICATION COMPLETE ====
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-01-13 14:43:28 UTC; 1min 34s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 629 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 658 (sshd)
     Tasks: 1 (limit: 4546)
    Memory: 3.2M
       CPU: 39ms
   CGroup: /system.slice/ssh.service
           └─658 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 13 14:43:28 server systemd[1]: Starting OpenBSD Secure Shell server...
Jan 13 14:43:28 server sshd[658]: Server listening on 0.0.0.0 port 22.
Jan 13 14:43:28 server sshd[658]: Server listening on :: port 22.
Jan 13 14:43:28 server systemd[1]: Started OpenBSD Secure Shell server.
```

Client's machine:(After we connected to the server):\$su root

```
Last login: Mon Jan 13 14:50:17 2025 from 192.168.18.100
user@server:~$ su root
Password:
su: Authentication failure
```

Exercise 2.1: Then, allow logging in as root

Server's machine(folder: '/etc/ssh/'): \$ sudo nano sshd_config

In this file, we must descomment the line 'PermitRootLogin' and change the value for 'yes'

```
# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Also, for activate the user *root*, we need to set new password: \$sudo passwd root

```
user@server:~$ sudo passwd root
New password: _
```

Unit 4. Web Server Administration

Client's machine:(After we connected to the server):\$su root

```
user@server:~$ su root
Password:
root@server:/home/user#
```

Exercise 2.2:Create two new users: user2 and user3, and a group named ssh_users. Add these users to the group and ensure that only the members of this group can access the SSH service

Server's machine: \$sudo groupadd ssh_users

```
user@server:~$ sudo groupadd ssh_users
user@server:~$
```

Server's machine: \$useradd -m(with this parameter, when the server create the user, also create the personal folder for the user called /home/<name_of_user>) user2 && sudo passwd user2

```
user@server:~$ sudo useradd -m user2
user@server:~$ sudo passwd user2
New password:
Retype new password:
passwd: password updated successfully
user@server:~$ _
```

```
user@server:~$ sudo useradd -m user3
user@server:~$ sudo passwd user3
New password:
Retype new password:
passwd: password updated successfully
user@server:~$ _
```

Server's machine: \$usermod -aG ssh_users user2 && usermod -aG ssh_users user3

```
user@server:~$ sudo usermod -aG ssh_users user2 && sudo usermod -aG ssh_users user3
user@server:~$ _
```

Server's machine(folder:'/etc/ssh/'): \$ sudo nano sshd_config

```
#PubkeyAuthentication yes
AllowGroups ssh_users_
# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
```

Unit 4. Web Server Administration

Server's machine: `$ sudo nano systemctl restart ssh && sudo systemctl status ssh`

```
user@server:~$ systemctl start ssh && systemctl status ssh
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'ssh.service'.
Authenticating as: user
Password:
==== AUTHENTICATION COMPLETE ====
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-01-13 14:43:28 UTC; 1min 34s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 629 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 658 (sshd)
      Tasks: 1 (limit: 4546)
     Memory: 3.2M
        CPU: 39ms
    CGroup: /system.slice/ssh.service
            └─658 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 13 14:43:28 server systemd[1]: Starting OpenBSD Secure Shell server...
Jan 13 14:43:28 server sshd[658]: Server listening on 0.0.0.0 port 22.
Jan 13 14:43:28 server sshd[658]: Server listening on :: port 22.
Jan 13 14:43:28 server systemd[1]: Started OpenBSD Secure Shell server.
```

Client's machine: `$ssh user2@192.168.18.251`

```
usuario@user:~$ ssh user2@192.168.18.251
user2@192.168.18.251's password:
Permission denied, please try again.
user2@192.168.18.251's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-130-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

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.

$
```


Unit 4. Web Server Administration

Exercise 3: The SSH service, by default, listens on port 22. Modify it to start on port 10022 (or another port), and figure out the command required to access the server on this port.

Server's machine(folder: '/etc/ssh/'): \$ sudo nano sshd_config

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

Port 10022
#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
```

Server's machine: \$ sudo nano systemctl restart ssh && sudo systemctl status ssh

```
user@server:~$ systemctl start ssh && systemctl status ssh
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'ssh.service'.
Authenticating as: user
Password:
==== AUTHENTICATION COMPLETE ====
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-01-13 14:43:28 UTC; 1min 34s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 629 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 658 (sshd)
      Tasks: 1 (limit: 4546)
     Memory: 3.2M
        CPU: 39ms
    CGroup: /system.slice/ssh.service
            └─658 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 13 14:43:28 server systemd[1]: Starting OpenBSD Secure Shell server...
Jan 13 14:43:28 server sshd[658]: Server listening on 0.0.0.0 port 22.
Jan 13 14:43:28 server sshd[658]: Server listening on :: port 22.
Jan 13 14:43:28 server systemd[1]: Started OpenBSD Secure Shell server.
```

To comprove that the SSH service is listening for port 10022 we will execute in the server's machine: \$grep -i port /etc/ssh/sshd_config

```
user@server:~$ grep -i port /etc/ssh/sshd_config
Port 10022
#GatewayPorts no
```

Client's machine: \$ssh -p 10022 user2@192.168.18.251

```
jsuarto@user:~$ ssh -p 10022 user2@192.168.18.251
user2@192.168.18.251's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-130-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Jan 13 15:33:00 2025 from 192.168.18.100
$
```

Unit 4. Web Server Administration

Exercise 4: By default, when authenticating successfully on the SSH server, it shows the date and time of the last connection. Find the option responsible for this behavior and modify it.

Server's machine(folder: '/etc/ssh/'): \$ sudo nano sshd_config

```
PrintLastLog no
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none
```

We must descomment the line 'PrintLastLog' and assign value for 'no'

To check this: Client's machine: \$ssh -p 10022 **user2@192.168.18.251**

```
usuario@user:~$ ssh -p 10022 user2@192.168.18.251
user2@192.168.18.251's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-130-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

$ █
```

Before, we appeared the datetime of the last connection, but now, how we changed the configuration, now it's not visible

Unit 4. Web Server Administration

Exercise 5: Configure the SSH server to enable X11 redirection, allowing the execution of graphical applications remotely. Test and verify its functionality.

Server's machine(folder:'/etc/ssh/'): \$ sudo nano sshd_config

```
# Example of overriding settings on a per-user basis
#Match User anoncvs
    X11Forwarding yes
    AllowTcpForwarding yes_
#
# ForceCommand cvs server
```

Server's machine: \$ sudo nano systemctl restart ssh && sudo systemctl status ssh

```
user@server:~$ systemctl start ssh && systemctl status ssh
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'ssh.service'.
Authenticating as: user
Password:
==== AUTHENTICATION COMPLETE ====
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-01-13 14:43:28 UTC; 1min 34s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 629 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 658 (sshd)
      Tasks: 1 (limit: 4546)
     Memory: 3.2M
        CPU: 39ms
   CGroup: /system.slice/ssh.service
           └─658 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 13 14:43:28 server systemd[1]: Starting OpenBSD Secure Shell server...
Jan 13 14:43:28 server sshd[658]: Server listening on 0.0.0.0 port 22.
Jan 13 14:43:28 server sshd[658]: Server listening on :: port 22.
Jan 13 14:43:28 server systemd[1]: Started OpenBSD Secure Shell server.
```

To check this: Client's machine: \$ssh -x -p 10022 user2@192.168.18.251

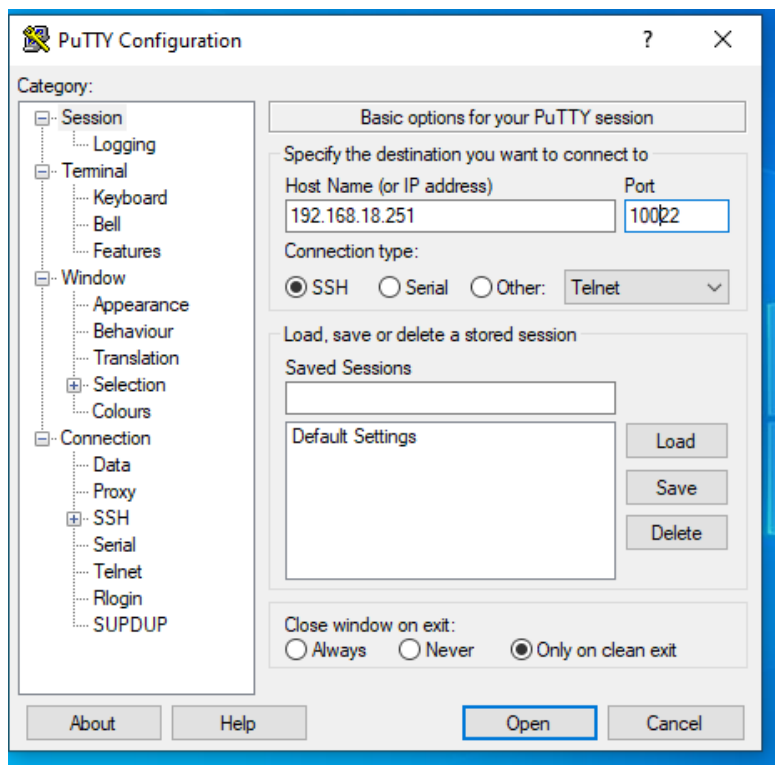
```
$ firefox &
$ -sh: 3: firefox: not found
^C
[1] + Done(127)                firefox
```

In this case, the server has not firefox app because it don't have any graphical enviroment, but the process will execute

Unit 4. Web Server Administration

Exercise 6: Test connecting from a Windows SSH client (e.g., PuTTY) to a GNU/Linux SSH server and verify the connection.

Client's W10 machine: Oppen the puTTY program on Windows and select SSH with the port and the IP of the server:



Click over Open and a window of CMD will be open and introduce the information from the user of the group ssh_users:

```
192.168.18.251 - PuTTY
login as: user2
user2@192.168.18.251's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-130-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

$
```

Practice 2: SSH

At the end of this practice, you must submit a script that explains and demonstrates the process and steps followed to complete each of the following tasks:

Perform client authentication from machine A to machine B using SSH with the following methods:

1. **Password-based authentication.**
2. **Public key authentication with a null passphrase**, where the users on both machines are identical. Copy the public key using `scp` and append it (without overwriting) to the `authorized_keys` file.
3. **Public key authentication with a non-null passphrase**, where the users on each machine are identical.

Practice 2:

Exercise 1: Perform client authentication from machine A to machine B using SSH with the following methods:

Exercise 1.1: Password-based authentication.

Server's machine(folder: '/etc/ssh/'): \$ sudo nano sshd_config

```
Port 10022
#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 no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes
#AllowGroups ssh_users
# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
```

Server's machine: \$ sudo nano systemctl restart ssh && sudo systemctl status ssh

```
user@server:~$ systemctl start ssh && systemctl status ssh
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'ssh.service'.
Authenticating as: user
Password:
==== AUTHENTICATION COMPLETE ====
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-01-13 14:43:28 UTC; 1min 34s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 629 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 658 (sshd)
       Tasks: 1 (limit: 4546)
      Memory: 3.2M
         CPU: 39ms
    CGroup: /system.slice/ssh.service
            └─658 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 13 14:43:28 server systemd[1]: Starting OpenBSD Secure Shell server...
Jan 13 14:43:28 server sshd[658]: Server listening on 0.0.0.0 port 22.
Jan 13 14:43:28 server sshd[658]: Server listening on :: port 22.
Jan 13 14:43:28 server systemd[1]: Started OpenBSD Secure Shell server.
```

To check this: Client's machine: \$ssh -p 10022 user2@192.168.18.251

```
usuario@user:~$ ssh -p 10022 user2@192.168.18.251
user2@192.168.18.251's password: █
```

Exercise 1.2: Public key authentication with a null passphrase, where the users on both machines are identical. Copy the public key using scp and append it (without overwriting) to the authorized keys file.

Client's machine: \$ sudo ssh-keygen -t rsa -b 2048

```
usuario@user:~$ sudo ssh-keygen -t rsa -b 2048
[+] password for usuario:
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
Your key fingerprint is:
56:htYhuRfSB3uP3z2N9MQnS5R3Dj1PW27bBhWJO25hdmA root@user
Your key's randomart image is:
[RSA 2048]----+
  .      ...|
  o o    E +.|
+ = o . *.B|
  * = o B X*|
+ S . = 0.0|
. o    . * X=|
    o +o*|
      ..|
      |
```

Now we must transfer this key to the server. Do it this:

Client's machine: \$ scp ~/.ssh/id_rsa.pub usuario@192.168.18.251:/tmp/clave_maquina_A.pub

```
usuario@user:~$ sudo scp -P 10022 /home/usuario/.ssh/id_rsa.pub user2@192.168
.18.251:/tmp/clave_maquinaCliente.pub
user2@192.168.18.251's password:
id_rsa.pub                                100% 566   664.8KB/s   00:00
usuario@user:~$ █
```

Unit 4. Web Server Administration

Now, we must aggregate the key of the client to the folders of `authorized_keys`, then we will log in and give the permissions:

Server's machine: `$ ssh user@192.168.18.251`

```
user@server:~$ ssh user@192.168.18.251
user@192.168.18.251's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-130-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

Server's machine: `#mkdir -p ~/.ssh && chmod 700 ~/.ssh`

```
root@server:/home/user# mkdir -p ~/.ssh && chmod 700 ~/.ssh
root@server:/home/user#
```

Now we must copy the key of the client that we saved in `TMP` folder to the correct folder(`authorized_keys`) and give it the correct permissions

Server's machine: `#cat /tmp/clave_maquina_A.pub >> ~/.ssh/authorized_keys && chmod 600 ~/.ssh/authorized_keys`

```
cat /tmp/clave_maquinaCliente.pub >> ~/.ssh/authorized_keys && chmod 600 ~/.ssh/authorized_keys
```

You should remember to delete the file in `/tmp` to avoid a security issue.

Server's machine: `#rm /tmp/clave_maquinaCliente.pub`

```
root@server:/home/user# rm /tmp/clave_maquinaCliente.pub
```

Finally, we will say to the server that takes this file.

Server's machine(folder: `/etc/ssh/`): `$ sudo nano sshd_config`

```
#MaxSessions 10

PubkeyAuthentication yes
AuthorizedKeysFile      .ssh/authorized_keys
```


Unit 4. Web Server Administration

Server's machine: `$ sudo nano systemctl restart ssh && sudo systemctl status ssh`

```
user@server:~$ systemctl start ssh && systemctl status ssh
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'ssh.service'.
Authenticating as: user
Password:
==== AUTHENTICATION COMPLETE ====
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-01-13 14:43:28 UTC; 1min 34s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 629 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 658 (sshd)
      Tasks: 1 (limit: 4546)
     Memory: 3.2M
        CPU: 39ms
    CGroup: /system.slice/ssh.service
            └─658 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 13 14:43:28 server systemd[1]: Starting OpenBSD Secure Shell server...
Jan 13 14:43:28 server sshd[658]: Server listening on 0.0.0.0 port 22.
Jan 13 14:43:28 server sshd[658]: Server listening on :: port 22.
Jan 13 14:43:28 server systemd[1]: Started OpenBSD Secure Shell server.
```

To check the results, go to the client and connect to the server:

Client's machine: `$ssh usuario@192.168.18.251`

```
usuario@user:~$ ssh user@192.168.18.251
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-130-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

user@server:~$
```

Practice 3: SSH

At the end of this practice, you must submit a script that explains and demonstrates the process and steps followed to complete each of the following tasks:

In this practice, you will use the `scp` command to copy files and directories between the client and server.

1. In your HOME directory, create a folder named `dir1`. Inside it:
 - Create another folder called `dir2`.
 - Create two text files named `f1.txt` and `f2.txt`.
 - Inside `dir2`, create a file named `f3.txt`.
2. Copy `f1.txt` and `f2.txt` to the server.
3. Copy the entire `dir2` folder to the server.
4. Modify `f1.txt` on the server and then copy it back to the client.

Practice 3:

Exercise 1:*In your HOME directory, create a folder named dir1. Inside it:*

Client's machine: \$sudo mkdir dir1:

```
usuario@user:~$ sudo mkdir dir1
[sudo] password for usuario:
usuario@user:~$
```

Exercise 1.1:*Create another folder called dir2.*

Client's machine:\$sudo mkdir dir2

```
usuario@user:~$ sudo mkdir dir2
usuario@user:~$
```

Exercise 1.2:*Create two text files named f1.txt and f2.txt.*

```
usuario@user:~/dir1$ sudo touch f1.txt f2.txt
usuario@user:~/dir1$
```

Exercise 1.3:*Inside dir2, create a file named f3.txt.*

```
usuario@user:~/dir1/dir2$ sudo touch f3.txt
usuario@user:~/dir1/dir2$
```

Exercise 2:*Copy f1.txt and f2.txt to the server.*


```
usuario@user:~$ scp /home/usuario/dir1/f1.txt /home/usuario/dir1/f2.txt user@
192.168.18.251:~/
f1.txt          100%  0    0.0KB/s   00:00
f2.txt          100%  0    0.0KB/s   00:00
usuario@user:~$
```

Exercise 3:*Copy the entire dir2 folder to the server.*

```
usuario@user:~$ scp -r /home/usuario/dir1/dir2/ user@192.168.18.251:~/
f3.txt          100%  0    0.0KB/s   00:00
usuario@user:~$
```

Exercise 4:*Modify f1.txt on the server and then copy it back to the client.*

```
root@server:/home/user# scp /home/user/f1.txt usuario@192.168.18.100:~/dir1/f1.txt
usuario@192.168.18.100's password:
f1.txt          100% 19    7.5KB/s   00:00
```



Unit 4. Web Server Administration

Practice 4: SSH

At the end of this practice, you must submit a script that explains and demonstrates the process and steps followed to complete each of the following tasks:

In this practice, you will configure **port forwarding** using SSH so that DNS queries (port 53) over the **TCP protocol** are securely transmitted through the port forwarding.

You can use your own DNS server or the classroom's DNS server (ask your instructor for access).

To test the port forwarding, ensure the DNS query commands specify TCP usage with:

- `dig -tcp target_system`
- `host -T target_system`

Practice 4:

Exercise 1:*In this practice, you will configure port forwarding using SSH so that DNS queries (port 53) over the TCP protocol are securely transmitted through the port forwarding. You can use your own DNS server or the classroom's DNS server (ask your instructor for access).*

Client's machine: `$sudo ssh -L 53:192.168.18.251:53 user@192.168.18.251`

```
usuario@user:~$ sudo ssh -L 53:192.168.18.251:53 user@192.168.18.251
user@192.168.18.251's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-130-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

Client's machine: `$ dig -p 53 @localhost google.com +tcp`

```
user@server:~$ sudo dig -p 53 @localhost google.com +tcp
[sudo] password for user:
; <<>> DiG 9.18.28-0ubuntu0.22.04.1-Ubuntu <<>> -p 53 @localhost google.c
om +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 17161
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; COOKIE: aff37500cded7e2d01000000678e70f9eafcfaa1bf96c14a (good)
;; QUESTION SECTION:
;google.com.                IN      A
;; ANSWER SECTION:
google.com.                 5       IN      CNAME   forcesafesearch.google.co
m.
forcesafesearch.google.com. 26387   IN      A       216.239.38.120
;; Query time: 56 msec
;; SERVER: 127.0.0.1#53(localhost) (TCP)
;; WHEN: Mon Jan 20 15:51:21 UTC 2025
;; MSG SIZE rcvd: 113
```

Client's machine: `$host -T google.com localhost`

```
user@server:~$ host -T google.com localhost
Using domain server:
Name: localhost
Address: 127.0.0.1#53
Aliases:

google.com is an alias for forcesafesearch.google.com.
forcesafesearch.google.com has address 216.239.38.120
forcesafesearch.google.com has IPv6 address 2001:4860:4802:32::78
```

Practice 5: SSH

At the end of this practice, you must

Unit 4. Web Server Administration

submit a script that explains and demonstrates the process and steps followed to complete each of the following tasks:

In this practice, you will create a **tunnel** to establish a secure connection between systems. As discussed, the information transmitted through the tunnel's associated IPs will be encrypted using the SSH connection.

1. The tunnel will connect your SSH server and a system acting as a client.
2. **Network configuration for the tunnel:**
 - Network: 10.1.1.0 / 30 (subnet mask 255.255.255.252). This configuration allows only two IPs: one for the server and one for the client.
3. To test the tunnel's functionality:
 - On the **server**, execute: ping 10.1.1.2.
 - On the **client**, execute: ping 10.1.1.1.
4. Access the web server running on the SSH server using the URL:

http://10.1.1.1

Practice 5:

Exercise 1: The tunnel will connect your SSH server and a system acting as a client.

Server's machine(folder: '/etc/ssh/'): \$ sudo nano sshd_config

```
#PidFile /run/sshd.pid
#MaxStartups 10:30:100
PermitTunnel yes
#ChrootDirectory none
#VersionAddendum none
```

Client's machine: \$ ssh -f -w 0:0 user@192.168.18.251 -N

```
usuario@user:~$ sudo ssh -f -w 0:0 user@192.168.18.251 -N
user@192.168.18.251's password:
usuario@user:~$ █
```