

Despliegue y preparación de un servidor Linux

Este documento consta con todos los pasos a seguir para desplegar y preparar un servidor Linux para Producción desde cero con los siguientes pasos:

1. Instalación del sistema base.

- **Ubuntu server.**
- **Zona horaria correcta.**
- **Nombre del servidor:** SRV-BASE-AZDINFARISSE.
- **Usuario administrador con contraseña.**
- **Verificación de errores y actualización de paquetes.**

2. Configuración de red y acceso remoto.

- **Asignar una IP estática**
- **Configurar el archivo `/etc/hosts`:** Nombre del servidor correctamente.
- **Instalar y habilitar el servicio SSH.**
- **Verificar la conexión remota desde otro sistema con ssh.**

3. Seguridad mínima obligatoria

- **Instalar y configurar UFW:** Solo tráfico por puerto 22 (SSH) y puerto 80 (HTTP).
- **Nuevo usuario:** Nombre→Desarrollador, con acceso limitado y sin permisos de superusuario.
- **Puerto SSH:** Cambiarlo a 2222 y reforzar la configuración (`/etc/ssh/sshd_config`).
- **Desactivar el acceso SSH del usuario root.**

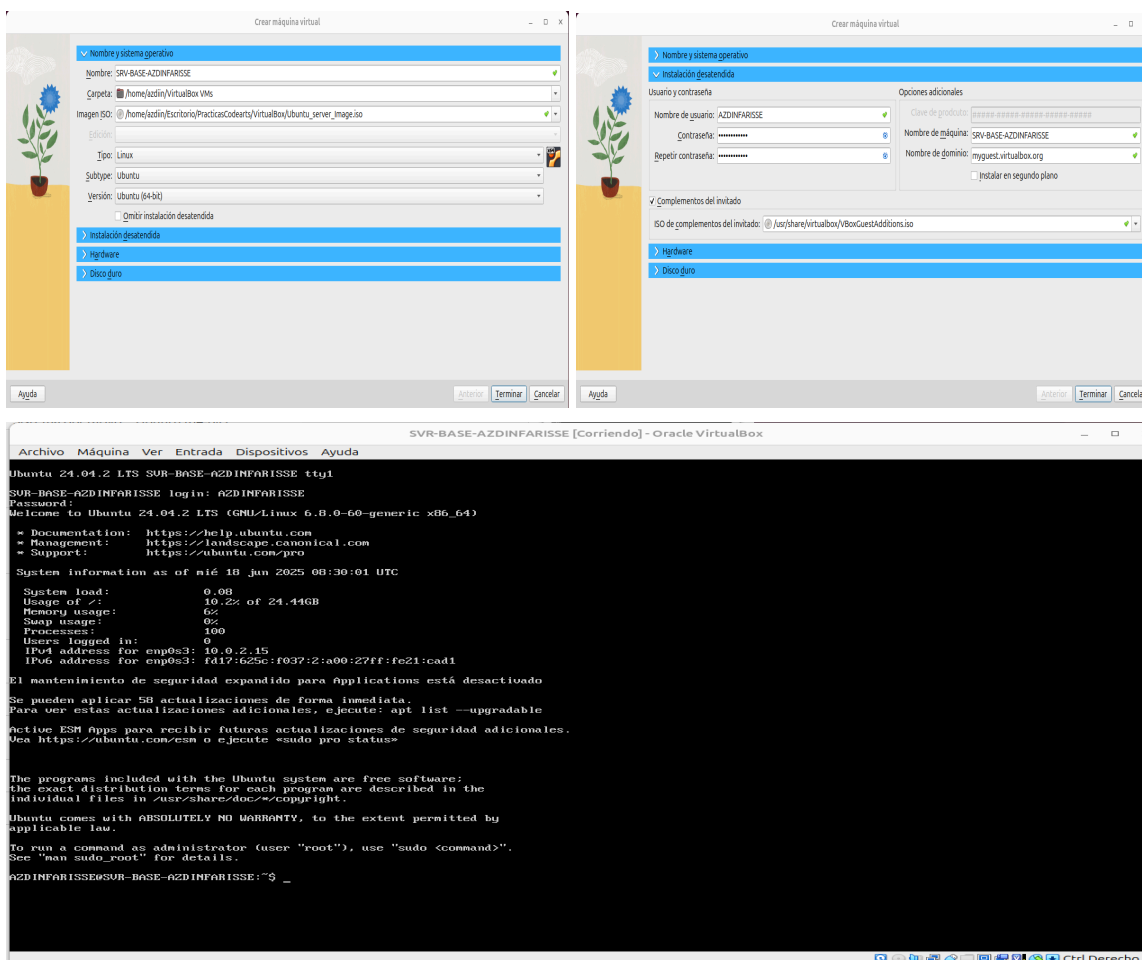
4. Estructura de carpetas y servicios iniciales.

- **Crear una estructura de carpetas en `/srv/` :**
 - ☐ `/srv/www` → para proyectos web.
 - ☐ `/srv/repositorios` → para guardar código fuente.
 - ☐ `/srv/docs` → para documentación técnica interna.
- **Establecer permisos específicos:**
 - ☐ El usuario desarrollador puede escribir solo en `/srv/www`.
 - ☐ Solo el usuario administrador puede acceder a `/srv/repositorios`
- **Instalar Apache2 y colocar página de prueba en `/srv/www`.**

Ahora que ya tenemos claros los pasos, procedemos al despliegue y preparación de nuestro Servidor Linux.

1. Instalación del sistema base.

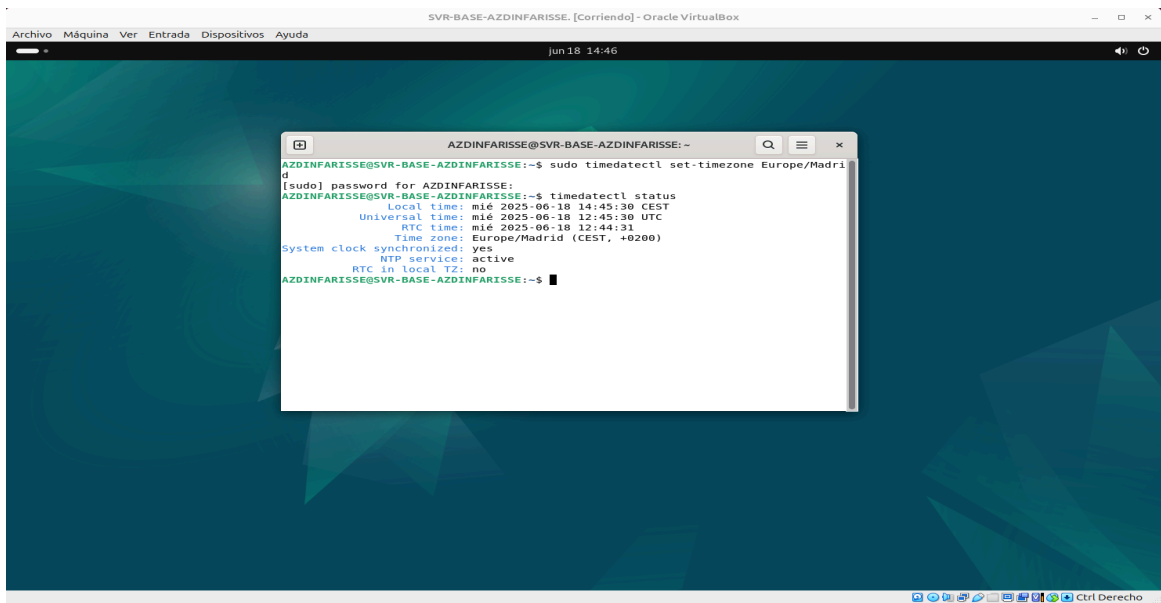
Instalamos nuestro Servidor Ubuntu en una VirtualBox y configuramos todos los parámetros solicitados



En este momento le he añadido la interfaz gráfica.

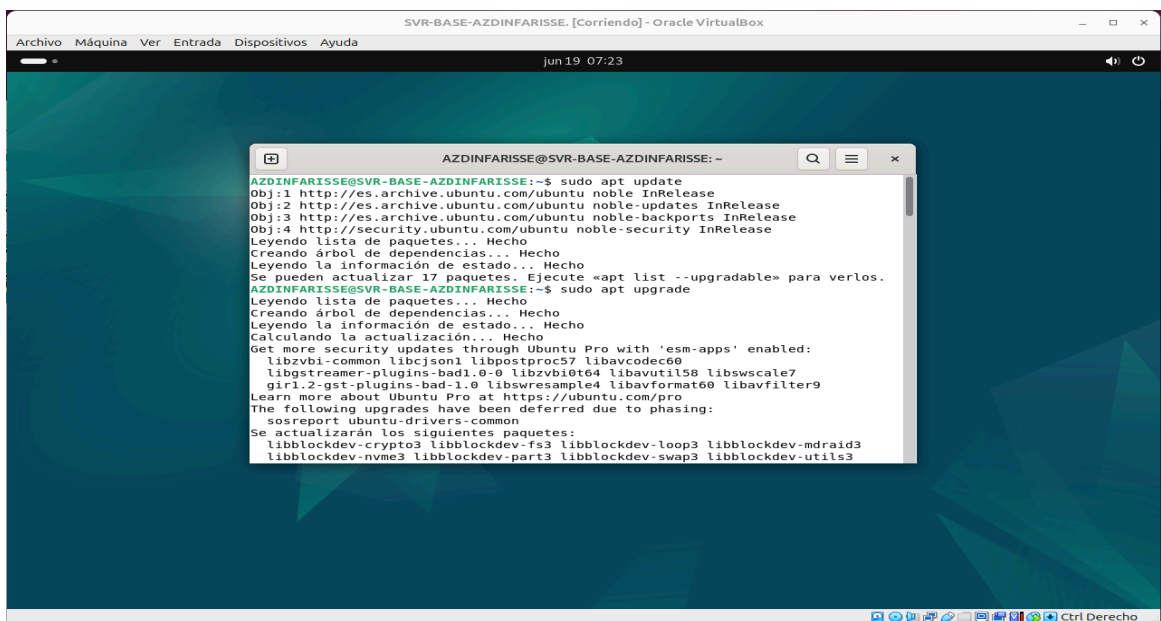
Para configurar la zona horaria correcta, hemos de introducir el comando:
sudo timedatectl set-timezone Europe/Madrid

Y verificamos que la zona horaria está correcta con el comando:
timedatectl status



Para actualizar los paquetes de nuestro servidor lo haremos a través de los comandos:

```
sudo apt update
sudo apt upgrade
```



2. Configuración de red y acceso remoto.

Para asignar una **IP estática** a nuestro servidor tenemos que dirigirnos a: **/etc/netplan** y hacer un **ls** para saber el nombre del archivo .yaml de esta carpeta. Una vez lo sabemos introducimos el comando:

sudo nano "nombre del archivo.yaml" en mi caso → **sudo nano 50-cloud-init.yaml**

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /etc/netplan
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ cd /etc/netplan
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ ls
50-cloud-init.yaml
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ sudo nano 50-cloud-init.yaml
```

Dentro de este archivo hemos de configurar la ip estática y quedaría así:

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /etc/netplan
GNU nano 7.2 50-cloud-init.yaml
network:
  version: 2
  ethernet:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.1.20/24
      nameservers:
        addresses:
          - 8.8.8.8
          - 1.1.1.1
      routes:
        - to: 0.0.0.0/0
          via: 192.168.1.1
```

Ahora guardamos el archivo y para que esta configuración se aplique introducimos:

sudo netplan apply

Y para comprobar que se ha introducido correctamente:

ip route

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /etc/netplan
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ cd /etc/netplan
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ ls
50-cloud-init.yaml
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ sudo nano 50-cloud-init.yaml
[sudo] password for AZDINFARISSE:
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ sudo netplan apply
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ ip route
default via 192.168.1.1 dev enp0s3 proto static
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.20
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$
```

Ahora que ya tenemos nuestra IP estática configurada, vamos a configurar el archivo /etc/hosts correctamente con el nombre del servidor.

Para ello introducimos:

sudo nano /etc/hosts y modificamos el archivo con los siguientes parámetros:

127.0.0.1 localhost

127.0.1.1 SVR-BASE-AZDINFARISSE

192.168.1.20 SVR-BASE-AZDINFARISSE

Guardamos el archivo y comprobamos que se ha cambiado el nombre correctamente:

cat /etc/hostname

Y hacemos un **ping SVR-BASE-AZDINFARISSE** para comprobar que todo está correcto.

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /etc/netplan
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ sudo nano /etc/hosts
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ cat /etc/hostname
SVR-BASE-AZDINFARISSE
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ ping SVR-BASE-AZDINFARISSE
PING SVR-BASE-AZDINFARISSE (127.0.1.1) 56(84) bytes of data:
64 bytes from SVR-BASE-AZDINFARISSE (127.0.1.1): icmp_seq=1 ttl=64 time=0.016 ms
64 bytes from SVR-BASE-AZDINFARISSE (127.0.1.1): icmp_seq=2 ttl=64 time=0.014 ms
64 bytes from SVR-BASE-AZDINFARISSE (127.0.1.1): icmp_seq=3 ttl=64 time=0.011 ms
64 bytes from SVR-BASE-AZDINFARISSE (127.0.1.1): icmp_seq=4 ttl=64 time=0.010 ms
64 bytes from SVR-BASE-AZDINFARISSE (127.0.1.1): icmp_seq=5 ttl=64 time=0.014 ms
^C64 bytes from SVR-BASE-AZDINFARISSE (127.0.1.1): icmp_seq=6 ttl=64 time=0.011
ms
64 bytes from SVR-BASE-AZDINFARISSE (127.0.1.1): icmp_seq=7 ttl=64 time=0.014 ms
^C
--- SVR-BASE-AZDINFARISSE ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6153ms
rtt min/avg/max/mdev = 0.010/0.012/0.016/0.002 ms
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$
```

Ahora vamos a instalar y habilitar el servicio SSH en nuestro servidor introduciendo el comando:

sudo apt install openssh-server

Para comprobar que el servicio esté activo:

sudo systemctl status ssh

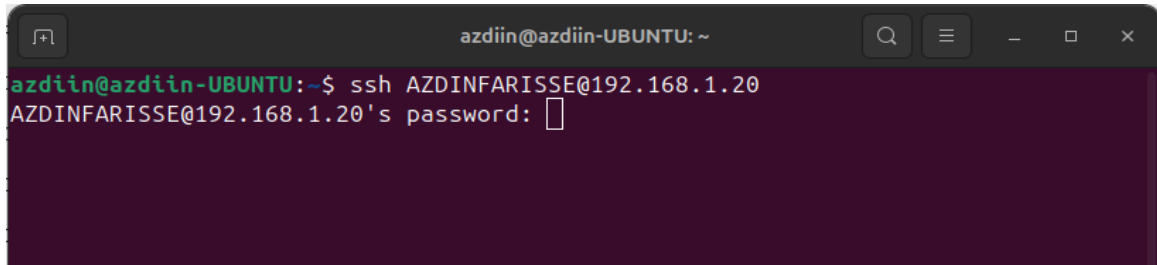
```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /etc/netplan
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-06-19 08:45:03 CEST; 6min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 7927 (sshd)
    Tasks: 1 (limit: 4605)
   Memory: 1.3M (peak: 4.1M)
      CPU: 43ms
   CGroup: /system.slice/ssh.service
           └─7927 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

jun 19 08:45:03 SVR-BASE-AZDINFARISSE systemd[1]: Starting ssh.service - OpenBSD Secu
jun 19 08:45:03 SVR-BASE-AZDINFARISSE sshd[7927]: Server listening on 0.0.0.0 port 22.
jun 19 08:45:03 SVR-BASE-AZDINFARISSE sshd[7927]: Server listening on :: port 22.
jun 19 08:45:03 SVR-BASE-AZDINFARISSE systemd[1]: Started ssh.service - OpenBSD Secu
jun 19 08:49:24 SVR-BASE-AZDINFARISSE sshd[9171]: Accepted password for AZDINFARISSE
jun 19 08:49:24 SVR-BASE-AZDINFARISSE sshd[9171]: pam_unix(sshd:session): session ope
lines 1-18/18 (END)
```

Ahora vamos a verificar la conexión remota desde la terminal de otro sistema. Para ello introducimos en el otro sistema el comando:

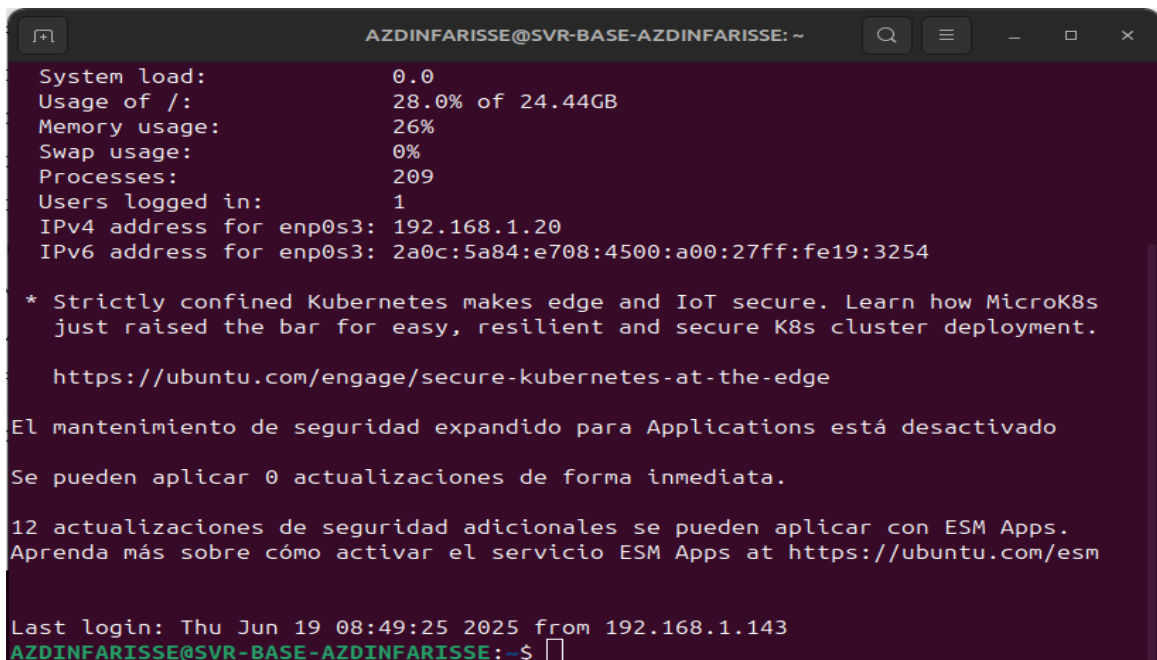
ssh usuario@IP_DEL_SERVIDOR, siendo en este caso:

ssh AZDINFARISSE@192.168.1.20



```
azdiin@azdiin-UBUNTU: ~  
azdiin@azdiin-UBUNTU:~$ ssh AZDINFARISSE@192.168.1.20  
AZDINFARISSE@192.168.1.20's password: 
```

y podemos comprobar que la conexión remota funciona correctamente.



```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~  
System load: 0.0  
Usage of /: 28.0% of 24.44GB  
Memory usage: 26%  
Swap usage: 0%  
Processes: 209  
Users logged in: 1  
IPv4 address for enp0s3: 192.168.1.20  
IPv6 address for enp0s3: 2a0c:5a84:e708:4500:a00:27ff:fe19:3254  
  
* 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 0 actualizaciones de forma inmediata.  
  
12 actualizaciones de seguridad adicionales se pueden aplicar con ESM Apps.  
Aprenda más sobre cómo activar el servicio ESM Apps at https://ubuntu.com/esm  
  
Last login: Thu Jun 19 08:49:25 2025 from 192.168.1.143  
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ 
```

3. Seguridad mínima obligatoria

Comandos para instalar y configurar **UFW**:

sudo apt install -y ufw → Instalación.

sudo ufw allow 22/tcp → Firewall permite tráfico al puerto 22 (SSH).

sudo ufw allow 80/tcp → Firewall permite tráfico al puerto 80 (HTTP).

sudo ufw enable → Para activar el firewall.

sudo ufw status → Para comprobar el estado del firewall.

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /etc/netplan
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ sudo apt install -y ufw
sudo ufw allow 22/tcp
sudo ufw allow 80/tcp
sudo ufw enable
sudo ufw status
```

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /etc/netplan
sudo ufw allow 80/tcp
sudo ufw enable
sudo ufw status
[sudo] password for AZDINFARISSE:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
ufw ya está en su versión más reciente (0.36.2-6).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
Rules updated
Rules updated (v6)
Rules updated
Rules updated (v6)
Firewall is active and enabled on system startup
Status: active

To Action From
--
22/tcp ALLOW Anywhere
80/tcp ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)
80/tcp (v6) ALLOW Anywhere (v6)
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$
```

Ahora crearemos un nuevo usuario llamado desarrollador con el comando:
sudo adduser desarrollador

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /etc/netplan
22/tcp (v6)          ALLOW          Anywhere (v6)
80/tcp (v6)          ALLOW          Anywhere (v6)

AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ sudo adduser desarrollador
info: Adding user `desarrollador' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `desarrollador' (1001) ...
info: Adding new user `desarrollador' (1001) with group `desarrollador (1001)' ...
info: Creating home directory `/home/desarrollador' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for desarrollador
Enter the new value, or press ENTER for the default
    Full Name []: Desarrollador
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
info: Adding new user `desarrollador' to supplemental / extra groups `users' ...
info: Adding user `desarrollador' to group `users' ...
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$
```

Ahora vamos a cambiar el puerto por defecto de SSH al **2222** y a reforzar la configuración (/etc/ssh/sshd_config) y a desactivar el acceso SSH del usuario root. Para ello introduciremos el comando:

sudo nano /etc/ssh/sshd_config

Y así se nos abrirá el archivo config en el que debemos de buscar el **puerto 22** y cambiarlo por el **2222** y buscar la línea **PermitRootLogin** y modificarla a **no**. Tanto la línea del puerto como la otra están con **#** (comentario), así que habrá que eliminar la **#** para que pueda leerse correctamente.

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /etc/netplan
GNU nano 7.2 /etc/ssh/sshd_config

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

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

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

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

Ahora para que estos cambios hagan efecto reiniciamos SSH con el comando:
sudo systemctl restart ssh

Ahora eliminaremos el puerto 22 para que no tener conflictos con:

sudo ufw delete allow 22/tcp

y añadiremos el puerto 2222 con:

sudo ufw allow 2222/tcp

y comprobamos el estado del firewall con:

sudo ufw status

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /etc/netplan
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ sudo nano /etc/ssh/sshd_config
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ sudo nano /etc/ssh/sshd_config
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ sudo systemctl restart ssh
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ sudo ufw allow 2222/tcp
Rule added
Rule added (v6)
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ sudo ufw delete allow 22/tcp
Command 'sud' not found, but there are 15 similar ones.
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ sudo ufw delete allow 22/tcp
Rule deleted
Rule deleted (v6)
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$ sudo ufw status
Status: active

To Action From
--
80/tcp ALLOW Anywhere
2222/tcp ALLOW Anywhere
80/tcp (v6) ALLOW Anywhere (v6)
2222/tcp (v6) ALLOW Anywhere (v6)

AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/netplan$
```

4. Estructura de carpetas y servicios iniciales

Para crear una estructura de carpetas en /srv/ con los directorios:

sudo mkdir -p /srv/www

sudo mkdir -p /srv/repositorios

sudo mkdir -p /srv/docs

Permisos usuario desarrollador solo escritura en /srv/repositorios:

sudo chown desarrollador:desarrollador /srv/www

sudo chmod 755 /srv/www

Permisos usuario administrador para acceder a /srv/repositorios:

sudo chown AZDINFARISSE:AZDINFARISSE /srv/repositorios

sudo chmod 700 /srv/repositorios

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /srv
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/srv$ sudo mkdir -p /srv/www
sudo mkdir -p /srv/repositorios
sudo mkdir -p /srv/docs
sudo chown desarrollador:desarrollador /srv/www
sudo chmod 755 /srv/www
sudo chown AZDINFARISSE:AZDINFARISSE /srv/repositorios
sudo chmod 700 /srv/repositorios
```

Ahora instalaremos el servidor web **Apache2** y colocaremos una página de prueba en `/srv/www`.

Para instalar el servidor Apache2 introducimos el comando:

sudo apt update

sudo apt install apache2

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
$ sudo apt update
$ sudo apt install apache2
Obj:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu noble InRelease
Des:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Obj:4 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease
Descargados 126 kB en 1s (139 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 2 paquetes. Ejecute «apt list --upgradable» para verlos.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-data apache2-utils
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-data apache2-utils
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
Se necesita descargar 351 kB de archivos.
Se utilizarán 1.675 kB de espacio de disco adicional después de esta operación.
¿Debes continuar? [S/n] e
```

Ahora lo activamos con:

sudo systemctl enable apache2

y lo iniciamos con:

sudo systemctl start apache2

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...

Service restarts being deferred:
  systemctl restart gdm.service
  systemctl restart systemd-logind.service

No containers need to be restarted.

User sessions running outdated binaries:
  AZDINFARISSE @ user manager service: gnome-session-b[2744], systemd[2589]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo systemctl start apache2
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$
```

Ahora creamos la página con el siguiente comando:

echo '<!DOCTYPE html><html><head><title>Prueba

Apache</title></head><body><h1>¡Apache funcionando desde

/srv/www!</h1></body></html>' | sudo tee /srv/www/index.html

Ahora le daremos permisos a Apache para que pueda leer /srv/www:

```
sudo chown -R desarrollador:www-data /srv/www
```

```
sudo chmod -R 755 /srv/www
```

y reiniciamos apache para aplicar los cambios:

```
sudo systemctl restart apache2
```

En este punto se comprueba que todo ha salido bien accediendo desde el navegador a:

<http://localhost>

Al entrar me estaba dando fallo y después de buscar información encontré que me faltaba modificar el archivo de configuración de apache

(**/etc/apache2/sites-available/000-default.conf**) y añadir las siguientes líneas:

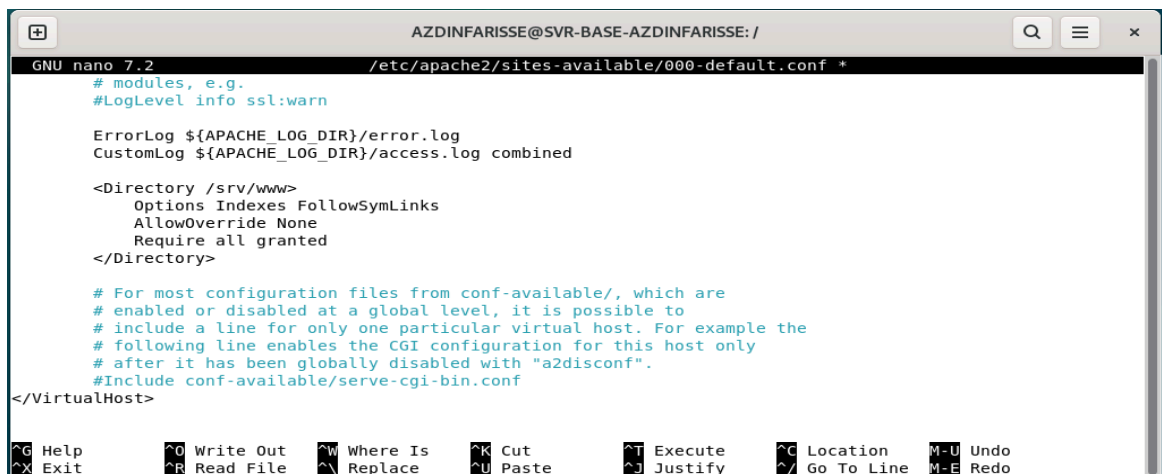
```
<Directory /srv/www>
```

```
Options Indexes FollowSymLinks
```

```
AllowOverride None
```

```
Require all granted
```

```
</Directory>
```



```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
GNU nano 7.2 /etc/apache2/sites-available/000-default.conf *
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

<Directory /srv/www>
Options Indexes FollowSymLinks
AllowOverride None
Require all granted
</Directory>

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Volvemos a probar a entrar a <http://localhost> y vemos que ahora sí nos accede a la página correctamente.

