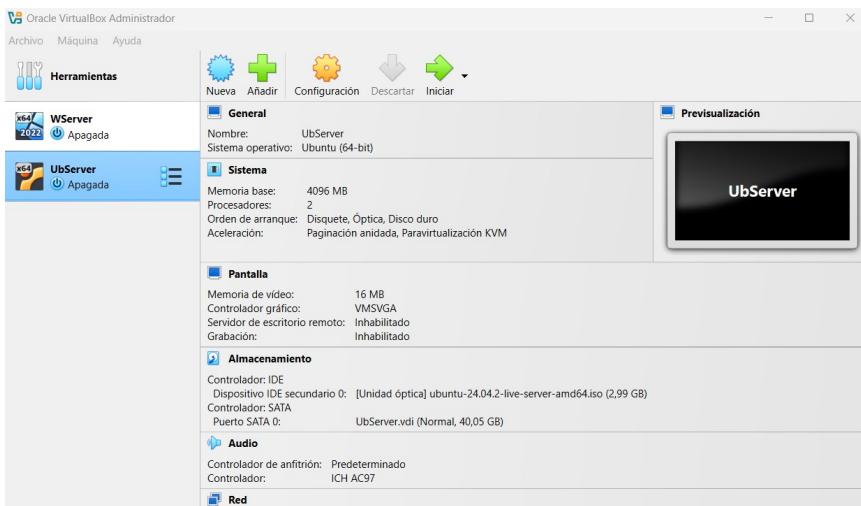


# INSTALACIÓN Y CONFIGURACIÓN DE LINUX SERVER

## Fase 1: Instalación del entorno de pruebas

- ✓ Descargar e instalar Ubuntu Server o Debian en una máquina virtual.
- ✓ Configurar la máquina virtual con 2 núcleos de CPU, 4GB de RAM y 40GB de almacenamiento.



## Fase 2: Configuración del servidor

- ✓ Configurar el sistema con un usuario administrador seguro.
- ✓ Asignar una IP estática en la red local y probar conectividad con ping.
- ✓ Habilitar el acceso remoto mediante SSH y verificar conexión.

Configurar IP estática en la red local y probar conectividad con ping:

Primero ip a para comprobar nuestra red.

Sudo nano /50-cloud-init.yaml para editar el archivo netplan con los siguientes valores:

```
Terminal - raulrp@raulrpser...
Terminal - raulrp@raulrpserver: /etc/netplan
GNU nano 7.2 50-cloud-init.yaml
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [192.168.1.75/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [1.1.1.1, 8.8.8.8]
```

En mi caso tuve que borrar nameservers como addresses para que el apply no me diera error.  
Netplan apply para aplicar los cambios.  
De nuevo ip a para ver que se nos cambió correctamente:

```
Terminal - raulrp@raulrpserver: /etc/netplan
File Edit View Terminal Tabs Help
raulrp@raulrpserver:/etc/netplan$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:33:49:bf brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.75/24 brd 192.168.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 2a0c:5a85:ed03:a000:a00:27ff:fe33:49bf/64 scope global mngtmpaddr noprefixroute
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe33:49bf/64 scope link
        valid_lft forever preferred_lft forever
raulrp@raulrpserver:/etc/netplan$
```

Hacemos ping para ver si funciona:

```
raulrp@raulrpserver:/etc/netplan$ ping -c 4 192.168.1.75
PING 192.168.1.75 (192.168.1.75) 56(84) bytes of data.
64 bytes from 192.168.1.75: icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from 192.168.1.75: icmp_seq=2 ttl=64 time=0.027 ms
64 bytes from 192.168.1.75: icmp_seq=3 ttl=64 time=0.027 ms
64 bytes from 192.168.1.75: icmp_seq=4 ttl=64 time=0.027 ms

--- 192.168.1.75 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3078ms
rtt min/avg/max/mdev = 0.025/0.026/0.027/0.000 ms
```

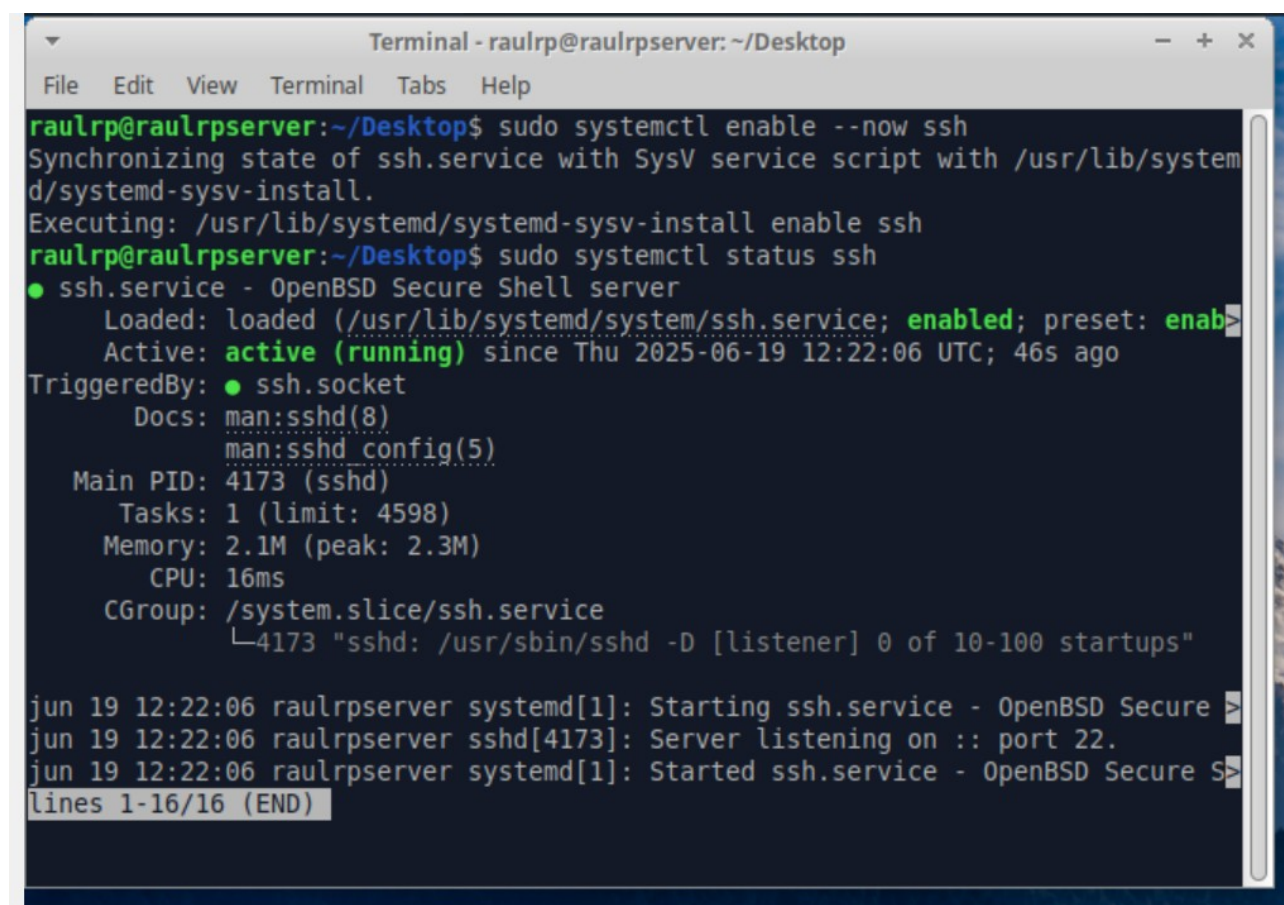
Habilitar el acceso remoto mediante SSH y verificar:

Instalamos SSH con los comandos: apt update y apt install openssh-server

```
Terminal - raulrp@raulrpserver: /etc/netplan
File Edit View Terminal Tabs Help
raulrp@raulrpserver:/etc/netplan$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:9.6p1-3ubuntu13.12).
```

Activamos el servicio SSH con: `sudo systemctl enable --now ssh`

Verificamos con: `sudo systemctl status ssh`



```
Terminal - raulrp@raulrpserver: ~/Desktop
File Edit View Terminal Tabs Help
raulrp@raulrpserver:~/Desktop$ sudo systemctl enable --now ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
raulrp@raulrpserver:~/Desktop$ 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 12:22:06 UTC; 46s ago
     TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
            man:sshd_config(5)
    Main PID: 4173 (sshd)
      Tasks: 1 (limit: 4598)
     Memory: 2.1M (peak: 2.3M)
        CPU: 16ms
       CGroup: /system.slice/ssh.service
              └─4173 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

jun 19 12:22:06 raulrpserver systemd[1]: Starting ssh.service - OpenBSD Secure S>
jun 19 12:22:06 raulrpserver sshd[4173]: Server listening on :: port 22.
jun 19 12:22:06 raulrpserver systemd[1]: Started ssh.service - OpenBSD Secure S>
lines 1-16/16 (END)
```

### Fase 3: Gestión de usuarios y seguridad

- ✓ Crear tres usuarios con diferentes permisos en el servidor.
- ✓ Asignar permisos específicos a cada usuario y probar accesos.
- ✓ Configurar el firewall con UFW y permitir solo las conexiones necesarias.

Crear tres usuarios con diferentes permisos en el servidor:

Vamos a crear los siguientes usuarios:

adminuser: Permisos de administrador.

devuser: Permisos de desarrollo.

guestuser: Permisos limitados.

```
sudo adduser adminuser
```

```
sudo usermod -aG sudo adminuser
```

```
sudo adduser devuser
```

```
sudo adduser guestuser
```



```
Terminal - raulrp@raulrpserver: ~/Desktop
File Edit View Terminal Tabs Help
raulrp@raulrpserver:~/Desktop$ sudo adduser adminuser
[sudo] password for raulrp:
info: Adding user `adminuser' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `adminuser' (1001) ...
info: Adding new user `adminuser' (1001) with group `adminuser (1001)' ...
info: Creating home directory `/home/adminuser' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for adminuser
Enter the new value, or press ENTER for the default
  Full Name []: adminuser
  Room Number []: 1
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
info: Adding new user `adminuser' to supplemental / extra groups `users' ...
info: Adding user `adminuser' to group `users' ...
raulrp@raulrpserver:~/Desktop$ sudo usermod -aG sudo adminuser
```

Asignar permisos:

admin user ya tiene permisos sudo.

Para devuser vamos a darle acceso a /var/www con permisos de lectura/escritura:

Creamos el directorio si no existe con: `sudo mkdir -p /var/www`

Cambiamos el propietario a devuser: `sudo chown devuser:devuser /var/www`

Le damos permisos: `sudo chmod 755 /var/www`

```
Terminal - raulrp@raulrpserver: ~/Desktop
File Edit View Terminal Tabs Help
raulrp@raulrpserver:~/Desktop$ sudo chown devuser:devuser /var/www
raulrp@raulrpserver:~/Desktop$ sudo chmod 755 /var/www
raulrp@raulrpserver:~/Desktop$
```

Para guestuser no hacemos nada porque los permisos por defecto le restringen a /home/guestuser

Para probar los accesos:.

Probar acceso sudo con adminuser

`su - adminuser`

`sudo apt update`

Probar acceso a /var/www con devuser

`su - devuser`

`cd /var/www`

`touch test.txt`

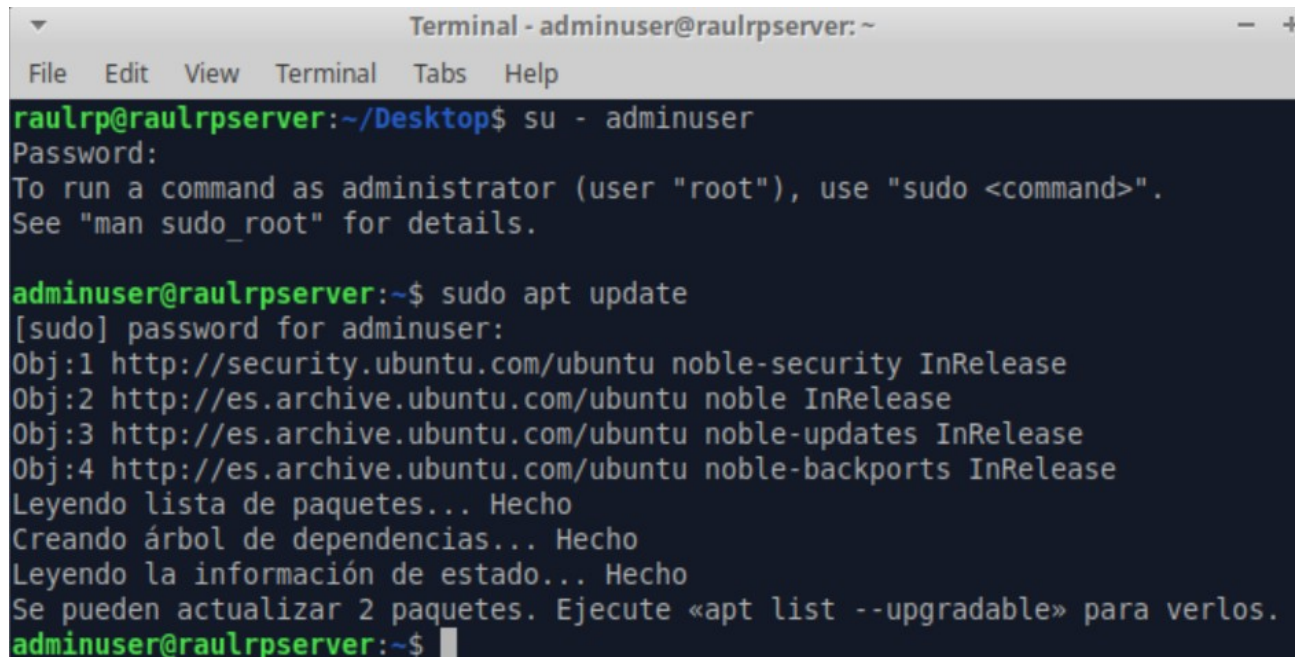
Intentar sudo con devuser (debería fallar)

sudo ls

Probar con guestuser:

su - guestuser

cd /var/www # debería no tener permisos



```
Terminal - adminuser@raulrpserver: ~
File Edit View Terminal Tabs Help

raulrp@raulrpserver:~/Desktop$ su - adminuser
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

adminuser@raulrpserver:~$ sudo apt update
[sudo] password for adminuser:
Obj:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu noble InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease
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.
adminuser@raulrpserver:~$
```



```
adminuser@raulrpserver:~$ su - devuser
Password:
devuser@raulrpserver:~$ cd /var/www
devuser@raulrpserver:/var/www$ touch test.txt
devuser@raulrpserver:/var/www$ sudo ls
[sudo] password for devuser:
devuser is not in the sudoers file.
```

Configurar el firewall con UFW y permitir solo las conexiones necesarias:

Instalar UFW: sudo apt install ufw

Activar UFW pero antes permitir SSH: sudo ufw allow OpenSSH

O también: sudo ufw allow 22/tcp

Activar UFW: sudo ufw enable

Ver estado del firewall: sudo ufw status verbose

```
Terminal - raulrp@raulrpserver: ~/Desktop
File Edit View Terminal Tabs Help

raulrp@raulrpserver:~/Desktop$ sudo apt install ufw
[sudo] password for raulrp:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.2-6).
ufw set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
raulrp@raulrpserver:~/Desktop$ sudo ufw allow OpenSSH
Rules updated
Rules updated (v6)
raulrp@raulrpserver:~/Desktop$ sudo ufw enable
Firewall is active and enabled on system startup
raulrp@raulrpserver:~/Desktop$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To Action From
--
22/tcp (OpenSSH) ALLOW IN Anywhere
22/tcp (OpenSSH (v6)) ALLOW IN Anywhere (v6)
```

#### Fase 4: Documentación y presentación

- ✓ Elaborar un informe en Google Docs detallando los pasos de instalación y configuración.
- ✓ Presentar en Google Slides las ventajas de Linux Server frente a otras soluciones.
- ✓ Justificar si Linux Server es una opción viable para Codearts Solutions.

#### Justificación:

##### Estabilidad y fiabilidad

Linux Server es reconocido por su alta estabilidad, incluso en entornos de producción a largo plazo.

Servidores Linux pueden funcionar durante años sin reiniciarse, ideal para servicios en línea 24/7.

##### Seguridad

Linux tiene una arquitectura sólida y una comunidad activa que lanza actualizaciones de seguridad frecuentes.

Ofrece herramientas como iptables, ufw, fail2ban, y AppArmor para endurecer el sistema.

##### Costo

Linux es gratuito y de código abierto, lo que reduce gastos en licencias.

Ideal para una empresa como Codearts Solutions si busca escalabilidad sin incurrir en altos costos de infraestructura.

## Flexibilidad y personalización

Al ser de código abierto, se puede adaptar a cualquier necesidad (servidores web, bases de datos, aplicaciones backend, contenedores, etc.).

Compatible con tecnologías como Docker, Kubernetes, NGINX, Node.js, Python, etc.

Soporte para desarrollo y despliegue

Linux es el sistema preferido por la mayoría de entornos de desarrollo moderno y DevOps.

Herramientas como Git, Jenkins, Ansible, y CI/CD pipelines se integran nativamente.

Comunidad y documentación

Hay amplia documentación y soporte comunitario.

Cualquier problema común tiene solución fácilmente localizable en foros como Stack Overflow, Ubuntu Forums, etc.

Casos de uso similares

Empresas líderes como Google, Facebook, Amazon y Netflix usan Linux Server para gran parte de su infraestructura.

Casos donde no sería ideal

Si Codearts usara software muy específico solo disponible en Windows Server (como ciertas versiones de SQL Server o software de escritorio especializado), no sería ideal usar solo Linux.

También si el equipo no tiene experiencia con administración de sistemas Linux, habría una curva de aprendizaje inicial.

## Conclusión

Linux Server es altamente viable para Codearts Solutions, especialmente si:

Se busca estabilidad, seguridad y bajo costo.

Se trabaja en entornos de desarrollo moderno, web, backend o cloud.

Se tiene (o se planea adquirir) conocimiento básico en administración de sistemas Linux.

Si Codearts Solutions quiere eficiencia, escalabilidad y flexibilidad sin costos de licencia, Linux Server es una excelente elección.

## Reto Día 7: Despliegue y Preparación de un Servidor Linux para Producción

### Fase 1: Instalación del sistema base

✓ Instalar Ubuntu Server o Debian desde ISO en una máquina virtual.

✓ Configurar durante la instalación:

- Zona horaria correcta

- Nombre del host: srv-base-[nombreAlumno]

- Usuario administrador personalizado con contraseña segura

✓ Verificar que el sistema arranca sin errores y actualiza sus paquetes (apt update && apt upgrade).

La instalación, verificación de errores en el arranque y actualización de paquetes ya la tengo hecha en este sistema y en otros retos anteriores, pero voy a proceder a cambiar el nombre del host y a cambiar la zona horaria:

Cambiar la zona horaria:

Con el comando: `timedatectl`, veremos nuestra zona horaria actual, pero con: `timedatectl list-timezones`, listaremos las zonas horarias disponibles, por ejemplo podríamos buscar la de Madrid con: `timedatectl set-timezones | grep Madrid` o mejor, cambiar la zona horaria directamente con: `sudo timedatectl set-timezone Europe/Madrid`



```
Terminal - raulrp@raulrpserver: ~/Desktop
File Edit View Terminal Tabs Help
raulrp@raulrpserver:~/Desktop$ sudo timedatectl set-timezone Europe/Madrid
[sudo] password for raulrp:
raulrp@raulrpserver:~/Desktop$ timedatectl
      Local time: jue 2025-06-19 15:08:42 CEST
      Universal time: jue 2025-06-19 13:08:42 UTC
      RTC time: jue 2025-06-19 13:08:42
      Time zone: Europe/Madrid (CEST, +0200)
System clock synchronized: yes
      NTP service: active
      RTC in local TZ: no
raulrp@raulrpserver:~/Desktop$
```

Cambiar el nombre del host (hostname):

Ver el hostname actual:

hostnamectl

Cambiar el hostname:

sudo hostnamectl set-hostname srv-base-RaulRecuerdo

Cambiar el archivo /etc/hosts (recomendado):

sudo nano /etc/hosts

Y asegúrate de que la línea del localhost tenga el nuevo nombre, por ejemplo:

127.0.0.1 localhost

127.0.1.1 srv-base-RaulRecuerdo

Verificar el cambio:

hostnamectl

```
Terminal - raulrp@raulrpserver: ~/Desktop
File Edit View Terminal Tabs Help
raulrp@raulrpserver:~/Desktop$ hostnamectl
Static hostname: raulrpserver
      Icon name: computer-vm
      Chassis: vm
      Machine ID: 33392ae51c5943e8bf6d0fd14c6ecb6d
      Boot ID: 51f8811291f84408826aa495acb2612c
Virtualization: oracle
Operating System: Ubuntu 24.04.2 LTS
      Kernel: Linux 6.8.0-60-generic
      Architecture: x86-64
Hardware Vendor: innotek GmbH
Hardware Model: VirtualBox
Firmware Version: VirtualBox
      Firmware Date: Fri 2006-12-01
      Firmware Age: 18y 6month 2w 4d
raulrp@raulrpserver:~/Desktop$ sudo hostnamectl set-hostname srv-base-RaulRecuerdo
raulrp@raulrpserver:~/Desktop$ sudo nano /etc/hosts
```



```
raulrp@raulrpserver:~/Desktop$ sudo hostnamectl set-hostname srv-base-RaulRecuer
0
raulrp@raulrpserver:~/Desktop$ sudo nano /etc/hosts
raulrp@raulrpserver:~/Desktop$ hostnamectl
  Static hostname: srv-base-RaulRecuerdo
        Icon name: computer-vm
        Chassis: vm
        Machine ID: 33392ae51c5943e8bf6d0fd14c6ecb6d
        Boot ID: 51f8811291f84408826aa495acb2612c
  Virtualization: oracle
  Operating System: Ubuntu 24.04.2 LTS
        Kernel: Linux 6.8.0-60-generic
  Architecture: x86_64
  Hardware Vendor: innotek GmbH
  Hardware Model: VirtualBox
  Firmware Version: VirtualBox
        Firmware Date: Fri 2006-12-01
        Firmware Age: 18y 6month 2w 4d
raulrp@raulrpserver:~/Desktop$
```

## Fase 2: Configuración de red y acceso remoto

- ✓ Asignar una IP estática válida en la red local.
- ✓ Configurar el archivo /etc/hosts correctamente con el nombre del servidor.
- ✓ Instalar y habilitar el servicio SSH.
- ✓ Verificar la conexión remota desde otro sistema con ssh.

Todo esto ya está hecho en esta práctica más arriba del documento.

## Fase 3: Seguridad mínima obligatoria

- ✓ Instalar y configurar UFW para que:
  - Solo permita tráfico por puerto 22 (SSH) y puerto 80 (HTTP).
- ✓ Crear un nuevo usuario llamado desarrollador, con acceso limitado y sin permisos de superusuario.
- ✓ Cambiar el puerto por defecto de SSH a 2222 y reforzar la configuración (/etc/ssh/sshd\_config).
- ✓ Desactivar el acceso SSH del usuario root.

UFW ya está instalado, procedo a la parte de solo permitir el tráfico por puerto 22 (SSH) y puerto 80 (HTTP):

Permitir solo puertos 22 (SSH) y 80 (HTTP):

```
sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw allow 22/tcp
sudo ufw allow 80/tcp
```

Habilitar UFW:

```
sudo ufw enable
```

sudo ufw status verbose

```
Terminal - raulrp@raulrpserver: ~/Desktop
File Edit View Terminal Tabs Help
raulrp@raulrpserver:~/Desktop$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
raulrp@raulrpserver:~/Desktop$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
raulrp@raulrpserver:~/Desktop$ sudo ufw allow 22/tcp
Rule added
Rule added (v6)
raulrp@raulrpserver:~/Desktop$ sudo ufw allow 80/tcp
Rule added
Rule added (v6)
raulrp@raulrpserver:~/Desktop$ sudo ufw enable
Firewall is active and enabled on system startup

raulrp@raulrpserver:~/Desktop$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To Action From
--
22/tcp (OpenSSH) ALLOW IN Anywhere
22/tcp ALLOW IN Anywhere
80/tcp ALLOW IN Anywhere
22/tcp (OpenSSH (v6)) ALLOW IN Anywhere (v6)
22/tcp (v6) ALLOW IN Anywhere (v6)
80/tcp (v6) ALLOW IN Anywhere (v6)

raulrp@raulrpserver:~/Desktop$
```

Crear un nuevo usuario llamado desarrollador sin permisos de superusuario:

sudo adduser desarrollador

sudo deluser desarrollador sudo

```
Terminal - raulrp@raulrpserver: ~/Desktop
File Edit View Terminal Tabs Help
raulrp@raulrpserver:~/Desktop$ sudo adduser desarrollador
info: Adding user `desarrollador' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `desarrollador' (1004) ...
info: Adding new user `desarrollador' (1004) with group `desarrollador (1004)' ...
..
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 []:
    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' ...
raulrp@raulrpserver:~/Desktop$ sudo deluser desarrollador sudo
fatal: The user `desarrollador' is not a member of group `sudo'.
raulrp@raulrpserver:~/Desktop$
```

Cambiar el puerto SSH a 2222 y reforzar /etc/ssh/sshd\_config:

Editar el archivo de configuración SSH:

```
sudo nano /etc/ssh/sshd_config
```

Realiza los siguientes cambios:

Port 2222

PermitRootLogin no

PasswordAuthentication yes

AllowUsers desarrollador

Reiniciar el servicio SSH:

```
sudo systemctl restart ssh
```

```
Terminal - raulrp@raulrpserver: ~/Desktop @raulrpserver: ~/Desktop
File Edit View Terminal Tabs Help
GNU nano 7.2 /etc/ssh/sshd_config *
#Port 222
#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
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^M Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

Actualizar reglas UFW para el nuevo puerto:

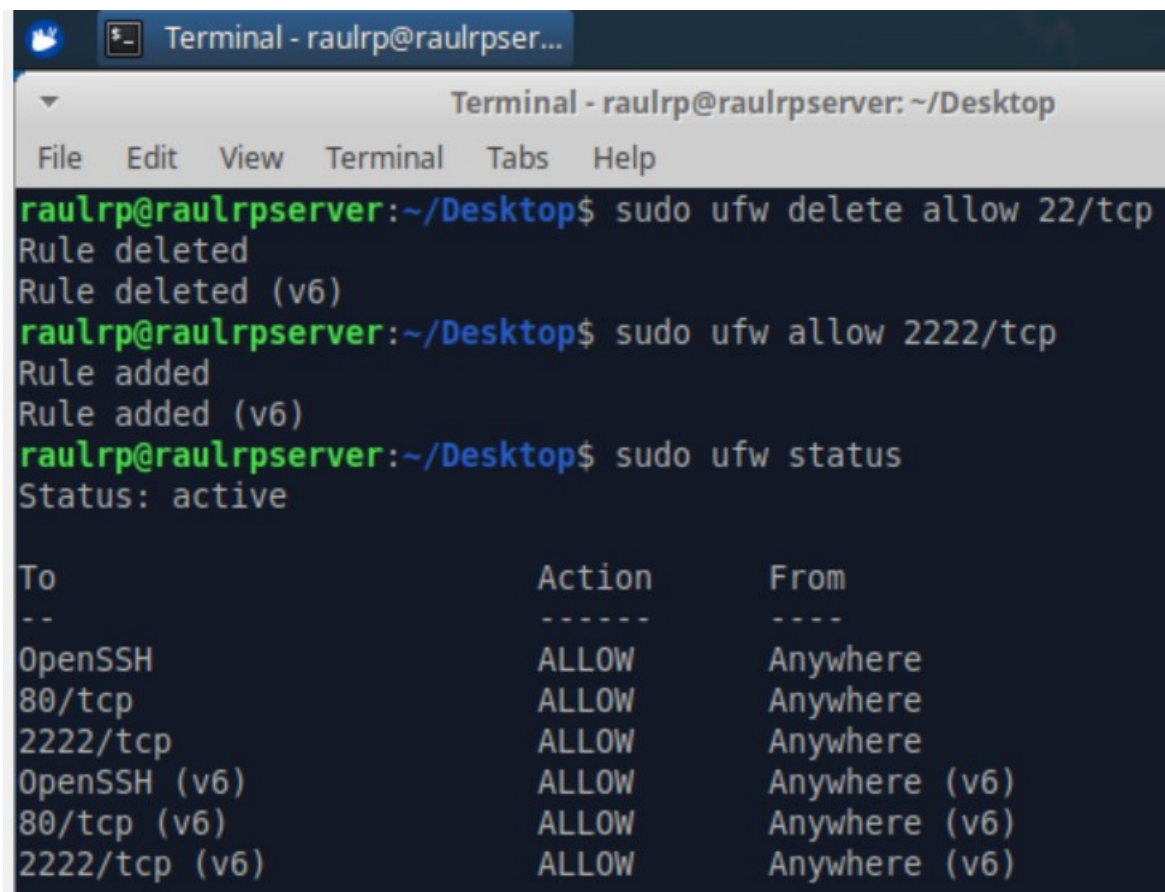
Eliminar puerto 22 y permitir 2222:

```
sudo ufw delete allow 22/tcp
```

```
sudo ufw allow 2222/tcp
```

Verifica el estado:

```
sudo ufw status
```



```
Terminal - raulrp@raulrpserver: ~/Desktop
File Edit View Terminal Tabs Help
raulrp@raulrpserver:~/Desktop$ sudo ufw delete allow 22/tcp
Rule deleted
Rule deleted (v6)
raulrp@raulrpserver:~/Desktop$ sudo ufw allow 2222/tcp
Rule added
Rule added (v6)
raulrp@raulrpserver:~/Desktop$ sudo ufw status
Status: active

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

Verificación final:

Asegúrate de poder iniciar sesión con el usuario desarrollador:

```
ssh desarrollador@IP -p 2222
```

#### Fase 4: Estructura de carpetas y servicios iniciales

✓ Crear una estructura de carpetas en /srv/ con los siguientes directorios:

- /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 el servidor web Apache2 o NGINX (a elegir) y colocar una página de prueba en /srv/www.

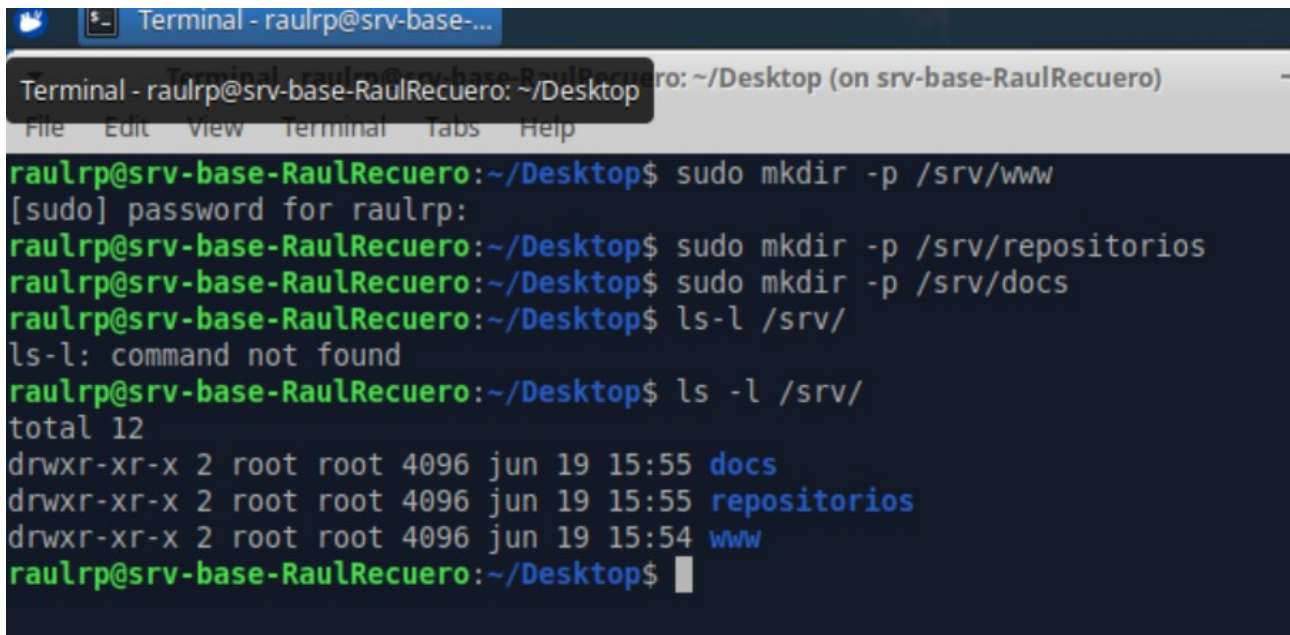


Crear la estructura de carpetas en /srv/:

```
sudo mkdir -p /srv/www  
sudo mkdir -p /srv/repositorios  
sudo mkdir -p /srv/docs
```

Verificamos que se hayan creado:

```
ls -l /srv/
```

A terminal window titled "Terminal - raulrp@srv-base-..." showing the execution of several commands. The user runs 'sudo mkdir -p /srv/www', followed by 'sudo mkdir -p /srv/repositorios' and 'sudo mkdir -p /srv/docs'. Then they run 'ls-l /srv/' which results in a 'command not found' error. Finally, they run 'ls -l /srv/' which displays the permissions and ownership for the three newly created directories: 'docs', 'repositorios', and 'www'. All three directories are owned by 'root' and have permissions 'drwxr-xr-x'.

```
Terminal - raulrp@srv-base-...  
Terminal - raulrp@srv-base-RaulRecuero: ~/Desktop (on srv-base-RaulRecuero)  
File Edit View Terminal Tabs Help  
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo mkdir -p /srv/www  
[sudo] password for raulrp:  
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo mkdir -p /srv/repositorios  
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo mkdir -p /srv/docs  
raulrp@srv-base-RaulRecuero:~/Desktop$ ls-l /srv/  
ls-l: command not found  
raulrp@srv-base-RaulRecuero:~/Desktop$ ls -l /srv/  
total 12  
drwxr-xr-x 2 root root 4096 jun 19 15:55 docs  
drwxr-xr-x 2 root root 4096 jun 19 15:55 repositorios  
drwxr-xr-x 2 root root 4096 jun 19 15:54 www  
raulrp@srv-base-RaulRecuero:~/Desktop$
```

Crear usuarios y asignar permisos:

Supongamos que tenemos dos usuarios:

desarrollador: usuario que podrá escribir en /srv/www

administrador: usuario que podrá acceder a /srv/repositorios

Si no existen los usuarios, los creamos:

```
sudo adduser desarrollador  
sudo adduser administrador
```

Asignar permisos:

Primero, establecemos los propietarios de las carpetas:

```
sudo chown desarrollador:desarrollador /srv/www  
sudo chown administrador:administrador /srv/repositorios
```

Ahora, ajustamos los permisos:

```
sudo chmod 755 /srv/www  
sudo chmod 700 /srv/repositorios
```

Verificamos:

ls -ld /srv/\*

```
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo chown desarrollador:desarrollador /srv/www
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo chown administrador:administrador /srv/repositorios
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo chmod 755 /srv/www
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo chmod 700 /srv/repositorios
raulrp@srv-base-RaulRecuero:~/Desktop$ ls -ld /srv/*
drwxr-xr-x 2 root root 4096 jun 19 15:55 /srv/docs
drwx----- 2 administrador administrador 4096 jun 19 15:55 /srv/repositorios
drwxr-xr-x 2 desarrollador desarrollador 4096 jun 19 15:54 /srv/www
raulrp@srv-base-RaulRecuero:~/Desktop$
```

Instalar el servidor web Apache2 o NGINX:

Elijo Apache2

sudo apt update

sudo apt install apache2 -y

Verificamos el estado del servicio:

sudo systemctl status apache2

```
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset:
   Active: active (running) since Thu 2025-06-19 16:05:22 CEST; 24s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 8715 (apache2)
    Tasks: 55 (limit: 4598)
   Memory: 5.3M (peak: 5.6M)
      CPU: 26ms
   CGroup: /system.slice/apache2.service
           └─8715 /usr/sbin/apache2 -k start
             └─8717 /usr/sbin/apache2 -k start
               └─8718 /usr/sbin/apache2 -k start

jun 19 16:05:22 srv-base-RaulRecuero systemd[1]: Starting apache2.service - The
jun 19 16:05:22 srv-base-RaulRecuero apachectl[8714]: AH00558: apache2: Could n
jun 19 16:05:22 srv-base-RaulRecuero systemd[1]: Started apache2.service - The
lines 1-16/16 (END)
```

Configurar Apache para servir contenido desde /srv/www:

Por defecto, Apache sirve desde /var/www/html. Vamos a cambiarlo.

Crear un archivo de prueba en /srv/www:

Como desarrollador:

sudo -u desarrollador nano /srv/www/index.html

```
Terminal - raulrp@srv-base-RaulRecuero: ~/Desktop (on srv-base-Rau
File Edit View Terminal Tabs Help
GNU nano 7.2 /srv/www/index.html
<!DOCTYPE html>
<html>
<head><title>Servidor funcionando</title></head>
<body><h1>Apache está funcionando en /srv/www</h1></body>
</html>
```

Modificar la configuración de Apache:  
Creamos un nuevo VirtualHost apuntando a /srv/www:

```
sudo nano /etc/apache2/sites-available/srv-www.conf
```

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /srv/www

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

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Desactivamos el sitio por defecto y activamos el nuevo:

```
sudo a2dissite 000-default.conf
sudo a2ensite srv-www.conf
sudo systemctl reload apache2
```

```
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo a2ensite srv-www.conf
Enabling site srv-www.
To activate the new configuration, you need to run:
    systemctl reload apache2
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl reload apache2
raulrp@srv-base-RaulRecuero:~/Desktop$
```

Abre navegador y visita la IP del servidor.

