



# **Born2beRoot - Debian**

**YOU CAN DO ANYTHING YOU WANT TO DO**

**VIRTUAL  
MACHINE**

**THIS IS YOUR WORLD**

## Contents

1.	Instalación de sudo.....	5
2.	Resize Logical Volumes (LVM) .....	6
	Parte Obligatoria.....	6
	Parte Obligatoria + Bonus.....	6
	Configuración parte obligatoria + bonus .....	7
	Redimensionamiento de los volúmenes lógicos.....	7
	Configuración del área de intercambio (swap) .....	8
3.	Configuración del sudo .....	9
	Objetivo .....	9
	Creación de la ruta y archivo de log de sudo.....	10
4.	Instalación y Configuración de SSH .....	11
	Instalación del servidor SSH.....	11
	Archivos de configuración .....	11
	Configuración del servidor SSH .....	12
	Reiniciar el servicio SSH.....	13
	Comprobar conexión SSH .....	13
5.	Instalación y configuración de UFW .....	14
6.	Conexión al VirtualBox vía SSH .....	15
	Conexión por fuera de la máquina .....	15
	Conexión SSH.....	17
7.	Políticas de contraseña.....	18
	Instalar la biblioteca de calidad de contraseñas.....	19
	Configurar las reglas de calidad de contraseña .....	19
8.	Creación de usuarios y contraseñas .....	21
	Crear un nuevo usuario .....	21
	Crear grupos personalizados.....	21
9.	Monitorización con CRONTAB CONFIGURATION.....	22
	Crear el script de monitorización .....	22
	Contenido del script .....	22
	Permitir ejecutar el script sin pedir contraseña (sudoers) .....	23
	Programar su ejecución automática con CRONTAB .....	24
10.	Instalación y configuración de llmp stack, lighttpd.....	25
	Instalar Lighttpd (servidor web ligero) .....	26
11.	Instalación y configuración de MariaDB.....	27
	Cliente MariaDB.....	27

Probar el nuevo usuario .....	28
12.    Instalación de <i>PHP</i> .....	29
Editar configuración FastCGI para Lighttpd .....	29
Habilitar FastCGI y reiniciar Lighttpd .....	29
13.    WordPress SETUP .....	30
Instalación .....	30
Configurar WordPress.....	30
Dentro de wp-config.php.....	30
Ajustar permisos.....	31
14.    Configurar un servicio de WordPress a tu elección.....	32
Redis .....	32
Instalación .....	32
Configurar Redis .....	32
Asegurar el directorio de Redis .....	33
Configurar WordPress para usar Redis .....	33
Prueba que Redis funcione correctamente .....	33
Integrar con WordPress.....	34
Otros servicios .....	34
15.    Pasos prohibidos y verificaciones .....	37
15.1.    Instalar interfaz gráfica.....	37
15.2.    Verificar última versión .....	37
15.3.    Verificar particiones (Obligatoria) .....	37
15.4.    Preguntas.....	37
15.5.    Revisar SSH.....	38
15.6.    Revisar UFW o FIREWALLD .....	38
15.7.    Revisión 1.....	38
15.8.    Creación usuario y grupo en tu delante .....	39
15.9.    Script Monitoring.sh.....	39

# 1. Instalación de sudo

- ***su -*** = El guion (-) carga el entorno completo del usuario root (variables, rutas, etc.).
- ***apt update*** = Actualiza la lista de repositorios para asegurarte de que instalas las versiones más recientes.
- ***apt install sudo -y*** = Instalamos el paquete sudo, permite ejecutar comandos como superusuario sin necesidad de iniciar sesión como root.
- ***adduser serromer sudo*** = Agregamos el usuario al grupo sudo, Otorga a "serromer" permisos para usar sudo.
- ***getent group sudo*** = Verificamos que el usuario pertenece al grupo sudo.
- ***reboot*** = Para aplicar correctamente los cambios de grupo.
- ***sudo -V*** = Verificamos la versión de sudo, confirma que el comando sudo está instalado y funcionando.

## ¿Qué es getent?

**getent** significa literalmente: ***GET entries (obtener entradas)***. Es una herramienta del sistema Linux que consulta bases de datos del sistema definidas en el archivo ***/etc/nsswitch.conf***. Sirve para obtener información sobre usuarios, grupos, hosts, servicios, contraseñas, etc. desde las fuentes configuradas (archivos locales, LDAP, NIS, etc.).

Comando	Qué muestra	Ejemplo de salida
<b>getent passwd</b>	Lista todos los usuarios del sistema	serromer:x:1001:1001::/home/serromer:/bin/bash
<b>getent group</b>	Lista todos los grupos del sistema	sudo:x:27:serromer
<b>getent group sudo</b>	Muestra solo el grupo sudo y sus miembros	sudo:x:27:serromer
<b>getent hosts localhost</b>	Muestra información de red del host	127.0.0.1 localhost

## 2. Resize Logical Volumes (LVM)

“Resize Logical Volumes (LVM)” significa **cambiar el tamaño de una partición lógica** en un sistema que usa **LVM (Logical Volume Manager)** — es decir, **ampliar o reducir el espacio disponible** en una parte del disco sin tener que recrear todo el sistema.

De acuerdo con el formato que nos dieron, debe presentarse de esta forma o parecido, para eso debes configurar bien la máquina virtual, para que se parezca los más exacto.

### Parte Obligatoria

```
wil@wil:~$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda        8:0    0   8G  0 disk
└─sda1     8:1    0 487M 0 part  /boot
└─sda2     8:2    0   1K  0 part
└─sda5     8:5    0 7.5G 0 part
  └─sda5_crypt 254:0  0 7.5G 0 crypt
    ├─wil--vg-root 254:1  0 2.8G 0 lvm   /
    ├─wil--vg-swap_1 254:2  0 976M 0 lvm   [SWAP]
    ├─wil--vg-home  254:3  0 3.8G 0 lvm   /home
sr0       11:0   1 1024M 0 rom

wil@wil:~$ _
```

### Parte Obligatoria + Bonus

```
# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda        8:0    0 30.8G 0 disk
└─sda1     8:1    0 500M 0 part  /boot
└─sda2     8:2    0   1K  0 part
└─sda5     8:5    0 30.3G 0 part
  └─sda5_crypt 254:0  0 30.3G 0 crypt
    ├─LVMGroup-root 254:1  0 10G  0 lvm   /
    ├─LVMGroup-swap 254:2  0 2.3G 0 lvm   [SWAP]
    ├─LVMGroup-home 254:3  0 5G   0 lvm   /home
    ├─LVMGroup-var  254:4  0 3G   0 lvm   /var
    ├─LVMGroup-srv  254:5  0 3G   0 lvm   /srv
    ├─LVMGroup-tmp  254:6  0 3G   0 lvm   /tmp
    └─LVMGroup-var--log 254:7  0 4G   0 lvm   /var/log
sr0       11:0   1 1024M 0 rom
```

Cuando creas la máquina virtual, se genera automáticamente con una configuración base. Como todas se parecen bastante, no hay problema — todo tranqui.

Aun así, te recomiendo que al crearla elijas desde el principio los valores correctos de:

- Base Memory (RAM)
- Processors (CPU)
- Disk Size (tamaño del disco)

Así te aseguras de que tu máquina se parezca lo más posible al formato exigido, tanto en la **parte obligatoria** como en la **parte bonus**. Si planeas hacer la parte bonus, lo mejor es que configures el tamaño del disco según los requisitos del bonus desde el inicio, directamente en la configuración de

VirtualBox. De esa forma, evitarás problemas después con el espacio o con tener que redimensionar el disco más adelante.

#### **Recomendación:**

##### **Parte Obligatoria**

Esta configuración es la más pequeña y básica, enfocada en la eficiencia.

- Base Memory (RAM): 1024 MB (o 1 GB)
- Processors (CPU): 1
- Disk Size (Tamaño del disco): 8 GB

##### **Parte Obligatoria + Bonus**

Esta configuración es más grande y compleja, para acomodar múltiples particiones LVM.

- Base Memory (RAM): 2048 MB (o 2 GB)
- Processors (CPU): 2
- Disk Size (Tamaño del disco): 35 GB (Mínimo; 40 GB es más seguro)

## **Configuración parte obligatoria + bonus**

Para obtener el resultado mostrado anteriormente, debemos configurar lo siguiente:

**Recuerda:** todos estos comandos deben ejecutarse como **root**, o bien anteponiendo **sudo** si no lo eres.

### **Redimensionamiento de los volúmenes lógicos**

```
Sudo lvresize -r -L 10G /dev/LVMGroup/root      # 1  
Sudo lvresize -r -L 5G /dev/LVMGroup/home       # 2  
Sudo lvresize -r -L 3G /dev/LVMGroup/var        # 3  
Sudo lvresize -r -L 3G /dev/LVMGroup/srv         # 4  
Sudo lvresize -r -L 3G /dev/LVMGroup/tmp         # 5  
Sudo lvresize -r -L 4G /dev/LVMGroup/var/log     # 6
```

- ✓ **lvresize** → comando para cambiar el tamaño de un volumen lógico (**Logical Volume**).
- ✓ **-r** → opción que ajusta automáticamente el sistema de archivos después del cambio (redimensiona filesystem y LV al mismo tiempo).
- ✓ **-L 10G** → nuevo tamaño que tendrá el volumen (10 gigabytes).
- ✓ **/dev/LVMGroup/root** → ruta del volumen lógico a modificar.

<b>/home</b>	Contiene los archivos personales de los usuarios.	En un entorno de servidor pequeño (como Born2beroot), no se necesita mucho espacio aquí.
<b>/var</b>	Almacena archivos variables: logs, colas de correo, cachés, etc.	3 GB es suficiente para un sistema ligero sin grandes bases de datos.
<b>/srv</b>	Almacena datos de servicios web o FTP.	3 GB es razonable para pruebas o sitios pequeños.
<b>/tmp</b>	Guarda archivos temporales del sistema y de usuarios.	3 GB es adecuado para un sistema sin aplicaciones grandes.
<b>/var/log</b>	Contiene todos los logs del sistema y servicios (sudo, ssh, ufw, lighttpd, etc.).	4 GB da margen suficiente para registrar actividad sin sobrescribir el log.
<b>swap</b>	Configuración del área de intercambio (SWAP).	En entornos de prueba, una regla general es que la swap sea igual o un poco menor que la RAM. Si tienes 2 GB de RAM, 2.29 GB de swap es razonable.

## Configuración del área de intercambio (swap)

**sudo vgs** = Muestra todos los volúmenes groups (VG), su tamaño total y el espacio libre.

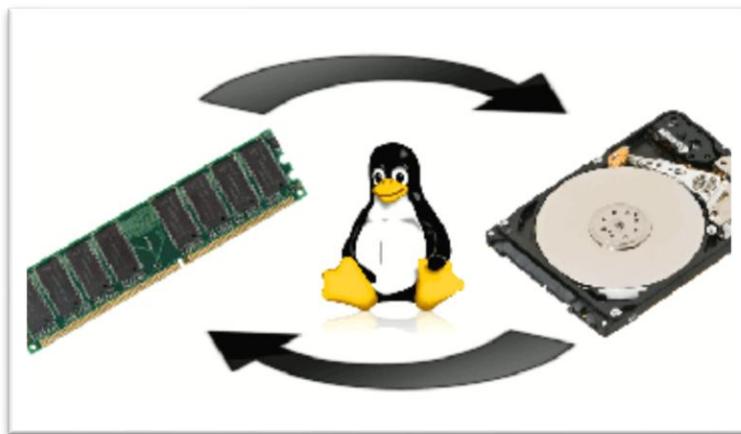
**sudo swapoff -a** = Desactiva temporalmente todas las áreas de intercambio (swap). Es necesario antes de redimensionar el volumen de swap para evitar corrupción de datos.

**lvresize -r -L 2.29G /dev/LVMGroup/swap** = Redimensionar el volumen lógico del swap

**sudo mkswap /dev/LVMGroup/swap** = Crea el espacio de intercambio (swap área) en ese volumen lógico. Debe ejecutarse siempre después de redimensionar.

**Lsblk** = Verificación final, para comprobar la estructura de los volúmenes

Este comando mostrará los distintos volúmenes lógicos creados y sus tamaños, confirmando que la configuración se aplicó correctamente.



### 3. Configuración del sudo

#### Objetivo

Personalizar el comportamiento de sudo para:

- Limitar intentos de contraseña
- Registrar toda la actividad
- Requerir un terminal interactivo
- Guardar los logs correctamente

Todo esto se hace en el archivo **/etc/sudoers** o, mejor aún, en un archivo separado dentro de **/etc/sudoers.d** para este proyecto no será necesario trabajar con sudoers.d

Agregamos a **/etc/sudoers** las siguientes configuraciones

- ✓ Defaults **passwd\_tries=3**
- ✓ Defaults **badpass\_message="Password is wrong. Please try again"**
- ✓ Defaults **logfile="/var/log/sudo/sudo.log"**
- ✓ Defaults **log\_input,log\_output**
- ✓ Defaults **iolog\_dir="/var/log/sudo"**
- ✓ Defaults **requiretty**
- ✓ Defaults **secure\_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"**

Línea	Explicación
<code>passwd_tries=3</code>	Limita a 3 intentos de introducir la contraseña antes de que <code>sudo</code> falle.
<code>badpass_message="..."</code>	Muestra un mensaje personalizado cuando la contraseña es incorrecta.
<code>logfile="/var/log/sudo/sudo.log"</code>	Define el archivo donde se guardarán los registros básicos de sudo.
<code>log_input,log_output</code>	Hace que <code>sudo</code> registre todo lo que el usuario escribe y todo lo que el sistema muestra (entradas/salidas). Esto aumenta la trazabilidad.
<code>iolog_dir="/var/log/sudo"</code>	Define el directorio donde se almacenarán los logs detallados de entrada/salida.
<code>requiretty</code>	Obliga a que <code>sudo</code> solo funcione si se ejecuta desde un terminal interactivo (más seguro; evita ejecuciones automáticas o remotas sin sesión real).
<code>secure_path="..."</code>	Asegura que los comandos de <code>sudo</code> se ejecuten solo en rutas de sistema seguras (previene ataques con scripts en rutas inseguras).

```

FILE Machine view Input Devices Help
GNU nano 8.4                               /etc/sudoers.tmp
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults    env_reset
Defaults    mail_badpass
Defaults    passwd_tries=3
Defaults    badpass_message="Password is wrong. Please try again."
Defaults    logfile="/var/log/sudo.log"
Defaults    log_input,log_output
Defaults    iolog_dir="/var/log/sudo"
Defaults    requiretty
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
#
# This fixes CVE-2005-4890 and possibly breaks some versions of kdesu
# (#1011624, https://bugs.kde.org/show_bug.cgi?id=452532)
Defaults    use_pty
#
# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:#sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"
#
# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:#sudo env_keep += "EDITOR"
#
# Completely harmless preservation of a user preference.
#Defaults:#sudo env_keep += "GREP_COLOR"
#
# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:#sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"
```

[ 60 líneas leidas ]

## Creación de la ruta y archivo de log de sudo

- ✓ sudo mkdir -p /var/log/sudo
- ✓ sudo touch /var/log/sudo/sudo.log
- ✓ sudo chmod 750 /var/log/sudo
- ✓ sudo chmod 640 /var/log/sudo/sudo.log
- ✓ sudo chown root:adm /var/log/sudo

Comando	Qué hace
mkdir -p	Crea el directorio (y subdirectorios si no existen).
touch	Crea el archivo vacío para los logs.
chmod 750	Permite acceso solo al root y al grupo adm (seguro).
chmod 640	Permite leer el archivo solo a root y grupo adm, nadie más.
chown root:adm	Asigna propiedad a root y grupo adm, que es el grupo típico para logs del sistema.

## 4. Instalación y Configuración de SSH

### Instalación del servidor SSH

- ✓ **sudo apt install openssh-server -y** = Instala el servicio que permitirá conexiones remotas seguras.
- ✓ **sudo systemctl status ssh** = Verificar que está instalado y corriendo.

```
root@serromer42:~# sudo apt install openssh-server -y
[...]
root@serromer42:~#



File Machine View Input Devices Help
reparando para desempaquetar .../15-libx11-6_2%3a1.8.12-1_amd64.deb ...
esempaquetando libx11-6:amd64 (2:1.8.12-1) ...
eleccionando el paquete libxext6:amd64 previamente no seleccionado.
reparando para desempaquetar .../16-libxext6_2%3a1.3.4-1+b3_amd64.deb ...
esempaquetando libxext6:amd64 (2:1.3.4-1+b3) ...
eleccionando el paquete libxmuu1:amd64 previamente no seleccionado.
reparando para desempaquetar .../17-libxmuu1_2%3a1.1.3-3+b4_amd64.deb ...
esempaquetando libxmuu1:amd64 (2:1.1.3-3+b4) ...
eleccionando el paquete xauth previamente no seleccionado.
reparando para desempaquetar .../18-xauth_1%3a1.1.2-1.1_amd64.deb ...
esempaquetando xauth (1:1.1.2-1.1) ...
onfigurando runit-helper (2.16.4) ...
onfigurando libxau6:amd64 (1:1.0.11-1) ...
onfigurando libxdmcp6:amd64 (1:1.1.5-1) ...
onfigurando libxcb1:amd64 (1.17.0-2+b1) ...
onfigurando libxcb0.10:amd64 (0.10.2-2) ...
onfigurando liburap0:amd64 (7.6.0-36) ...
onfigurando libx11-data (2:1.8.12-1) ...
onfigurando libpam-systemd:amd64 (257.8-1~deb10u2) ...
onfigurando libx11-6:amd64 (2:1.8.12-1) ...
onfigurando libutmpdbe:amd64 (0.73.0-3) ...
onfigurando libfdio2-1:amd64 (1.15.0-1+b1) ...
onfigurando libxmuu1:amd64 (2:1.1.3-3+b4) ...
onfigurando ncurses-term (6.5+20250216-2) ...
onfigurando openssh-client (1:10.0p1-7) ...
creando symlink '/etc/systemd/user/sockets.target.wants/ssh-agent.socket' → '/usr/lib/systemd/user/ssh-agent.socket'.
onfigurando libxext6:amd64 (2:1.3.4-1+b3) ...
onfigurando dbus-user-session (1.16.2-2) ...
onfigurando xauth (1:1.1.2-1.1) ...
onfigurando openssh-ftp-server (1:10.0p1-7) ...
onfigurando openssh-server (1:10.0p1-7) ...
creando config file '/etc/ssh/sshd_config' con nueva versión
creando SSH RSA key; esto puede tardar un poco de tiempo ...
972 SHA256:NNKVjQ2rSXLtH0lljXKueY+00Lay4KpUbgeppMBE root@serromer42 (RSA)
creando SSH2 ECDSA key; esto puede tardar un poco de tiempo ...
56 SHA256:ZX87fFrm2z+yo0OKJc2hNdPr0lyAupizDnSiuj+06Ic root@serromer42 (ECDSA)
creando SSH ED25519 key; esto puede tardar un poco de tiempo ...
56 SHA256:0KT104TmP2/XQOcneUEqLEGLhyssarWNDR0EsCHciLU root@serromer42 (ED25519)
creando usuario 'sshd' (el usuario que ejecuta el servicio) con ID 996 y GID 65534.
creando symlink '/etc/systemd/user/ssh.service' → '/usr/lib/systemd/system/ssh.service'.
creando symlink '/etc/systemd/system/multi-user.target.wants/ssh.service' → '/usr/lib/systemd/system/ssh.service'.
ssh.socket es un enlace simbólico a un servicio estático, no se está ejecutando.
creando symlink '/etc/systemd/system/ssh.service.wants/sshd-keygen.service' → '/usr/lib/systemd/system/sshd-keygen.service';
creando symlink '/etc/systemd/system/sshd.service.wants/sshd-keygen.service' → '/usr/lib/systemd/system/sshd-keygen.service';
creando symlink '/etc/systemd/system/sshd@.service.wants/sshd-keygen.service' → '/usr/lib/systemd/system/sshd-keygen.service';
creando symlink '/etc/systemd/system/ssh.socket.wants/sshd-keygen.service' → '/usr/lib/systemd/system/sshd-keygen.service'.
[...]
```

### Archivos de configuración

Hay **dos archivos** que suelen confundir a todos:

Archivo	Ruta
<b>/etc/ssh/ssh_config</b>	Configura el <b>cliente SSH</b> (cuando tú te conectas a otro equipo).
<b>/etc/ssh/sshd_config</b>	Configura el <b>servidor SSH</b> (cuando otros se conectan a ti). Este es el que necesitas para Born2beroot.

Por lo tanto, **solo debes modificar /etc/ssh/sshd\_config** para el proyecto.

## Configuración del servidor SSH

**sudo nano /etc/ssh/sshd\_config** = Edita el archivo del servidor SSH.

Busca las líneas (puedes usar **Ctrl + W** para buscar en nano) y cámbialas o añádelas si no existen:

- **Port 4242**
- **PermitRootLogin no**
- **PasswordAuthentication yes**

Línea	Qué hace
<b>Port 4242</b>	Cambia el <b>puerto por defecto (22)</b> a <b>4242</b> , como pide el proyecto (mejora básica de seguridad).
<b>PermitRootLogin no</b>	<b>Prohibe que el usuario root se conecte por SSH</b> (solo podrás entrar con tu usuario y usar sudo).
<b>PasswordAuthentication yes</b>	Permite usar contraseña para iniciar sesión (útil para pruebas; algunos sistemas la tienen desactivada por defecto).

```
root@serromer42:/etc/ssh# pwd
/etc/ssh
root@serromer42:/etc/ssh# ls
moduli  ssh_config.d  sshd_config.d      ssh_host_ecdsa_key.pub  ssh_host_ed25519_key.pub  ssh_host_rsa_key.pub
ssh_config  sshd_config  ssh_host_ecdsa_key  ssh_host_ed25519_key    ssh_host_rsa_key
root@serromer42:/etc/ssh# nano ssh_config
```

```
GNU nano 8.4                               sshd_config *

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

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

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

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

Port 4242
#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

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

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody
```

## Reiniciar el servicio SSH

```
File Machine View Input Devices Help
root@serromer42:/etc/ssh# systemctl status ssh
  ssh.service - OpenBSD Secure Shell server
    Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
      Active: active (running) since Sun 2025-11-02 16:49:41 CET; 5min ago
    Invocation: 400dd1f794144f8ba0181acb89c06657
      Docs: man:sshd(8)
             man:sshd_config(5)
    Main PID: 1719 (sshd)
      Tasks: 1 (limit: 1106)
     Memory: 1.3M (peak: 2M)
        CPU: 43ms
       CGroup: /system.slice/ssh.service
                 └─1719 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Nov 02 16:49:41 serromer42 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Nov 02 16:49:41 serromer42 sshd[1719]: Server listening on 0.0.0.0 port 22.
Nov 02 16:49:41 serromer42 sshd[1719]: Server listening on :: port 22.
Nov 02 16:49:41 serromer42 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
root@serromer42:/etc/ssh# service ssh status
  ssh.service - OpenBSD Secure Shell server
    Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
      Active: active (running) since Sun 2025-11-02 16:49:41 CET; 5min ago
    Invocation: 400dd1f794144f8ba0181acb89c06657
      Docs: man:sshd(8)
             man:sshd_config(5)
    Main PID: 1719 (sshd)
      Tasks: 1 (limit: 1106)
     Memory: 1.3M (peak: 2M)
        CPU: 43ms
       CGroup: /system.slice/ssh.service
                 └─1719 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Nov 02 16:49:41 serromer42 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Nov 02 16:49:41 serromer42 sshd[1719]: Server listening on 0.0.0.0 port 22.
Nov 02 16:49:41 serromer42 sshd[1719]: Server listening on :: port 22.
Nov 02 16:49:41 serromer42 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
root@serromer42:/etc/ssh#
```

## Comprobar conexión SSH

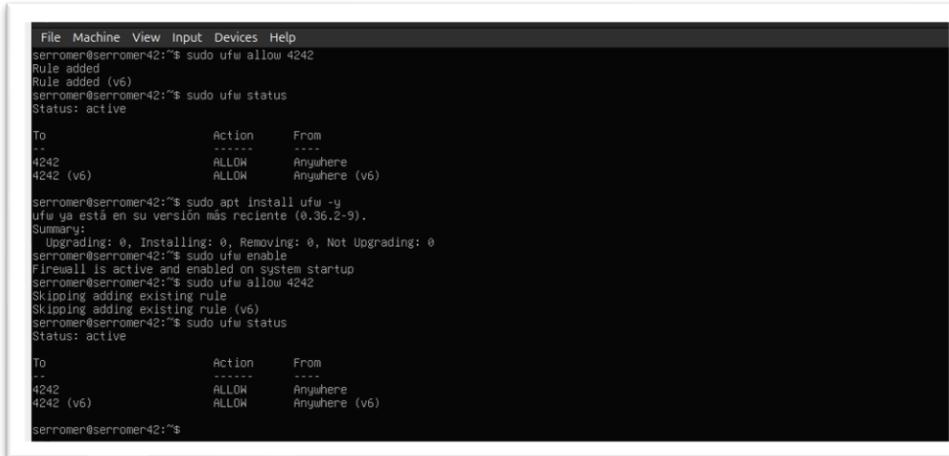
**ssh tu\_usuario@localhost -p 4242** = Prueba que puedes conectarte desde tu propia máquina (localhost).

## 5. Instalación y configuración de UFW

**UFW (Uncomplicated Firewall)** es una herramienta sencilla que gestiona **iptables**, el firewall nativo de Linux. Permite **controlar qué conexiones entran o salen** del sistema de forma fácil.

Por defecto, **todas las conexiones entrantes están bloqueadas** y solo las salientes están permitidas, hasta que tú las habilitas.

- ✓ **sudo apt install ufw -y** = Esto descarga e instala UFW y crea su servicio systemd (ufw.service).
- ✓ **sudo ufw enable** = Esto activa UFW con las reglas por defecto (bloquear todo entrante y permitir todo saliente).
- ✓ **sudo ufw allow 4242** = Esto agrega una regla que permite el tráfico entrante por el puerto 4242 (el que configuraste para SSH).
- ✓ **sudo ufw status verbose** = Ver estado detallado.



A terminal window showing the configuration of UFW. The user runs several commands: sudo ufw allow 4242, sudo ufw status, sudo apt install ufw -y, sudo ufw enable, and sudo ufw allow 4242 again. The output shows the creation of rules, the activation of UFW, and the final status.

```
File Machine View Input Devices Help
serromer@serromer42:~$ sudo ufw allow 4242
Rule added
Rule added (v6)
serromer@serromer42:~$ sudo ufw status
Status: active

To           Action    From
--           --       --
4242          ALLOW    Anywhere
4242 (v6)    ALLOW    Anywhere (v6)

serromer@serromer42:~$ sudo apt install ufw -y
ufw ya está en su versión más reciente (0.36.2-2).
Skipping upgrade: 0, Installing: 0, Removing: 0, Not Upgrading: 0
serromer@serromer42:~$ sudo ufw enable
Firewall is active and enabled on system startup
serromer@serromer42:~$ sudo ufw allow 4242
Skipping adding existing rule
Skipping adding existing rule (v6)
serromer@serromer42:~$ sudo ufw status
Status: active

To           Action    From
--           --       --
4242          ALLOW    Anywhere
4242 (v6)    ALLOW    Anywhere (v6)

serromer@serromer42:~$
```

## 6. Conexión al VirtualBox vía SSH

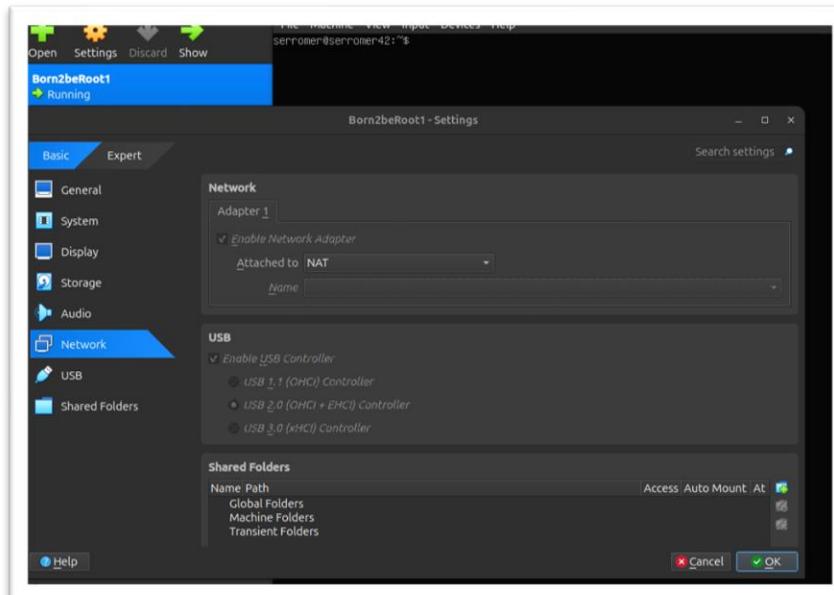
La conexión dentro del terminal de la máquina virtual debe estar activa en el puerto **4242**, es decir:  
**ssh <usuario>@127.0.0.1 -p 4242** = Esta conexión se realiza **desde dentro de la máquina virtual**.

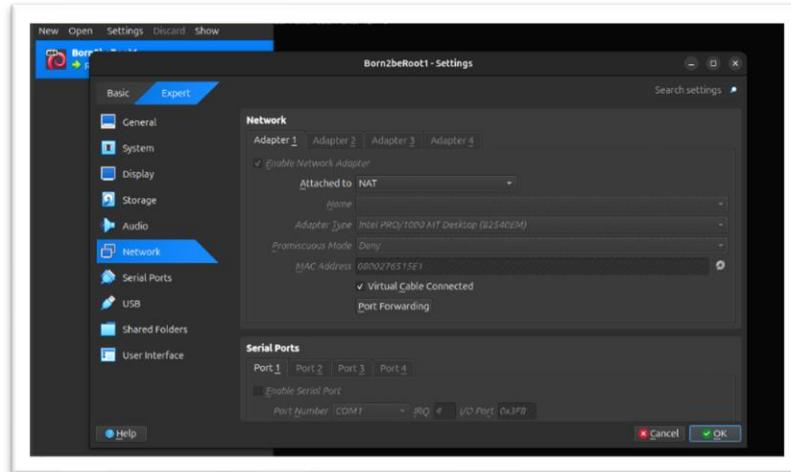
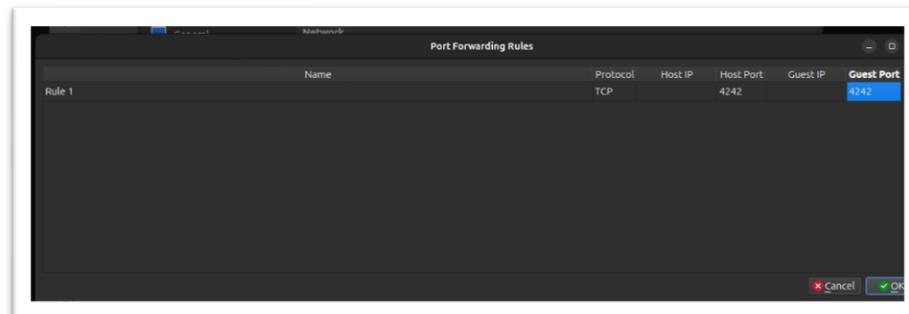
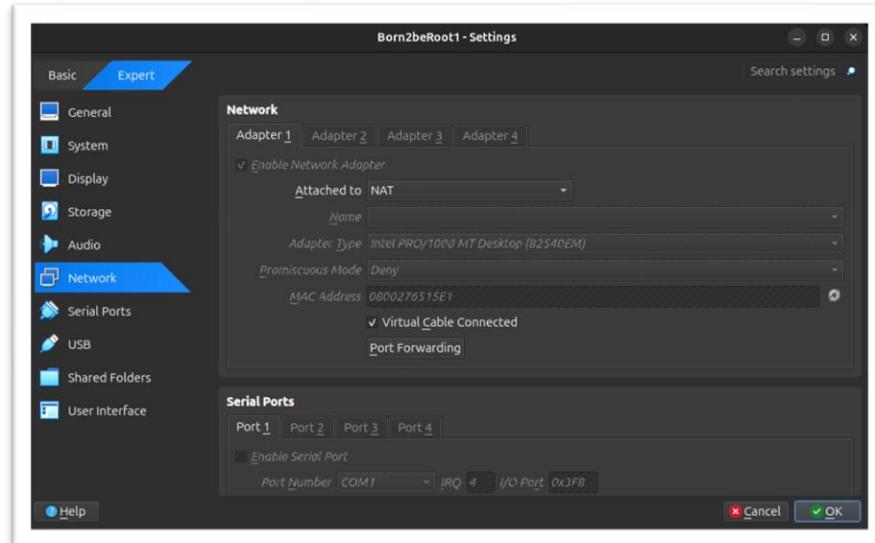
```
File Machine View Input Devices Help
serromer@serromer42:~$ ssh serromer@127.0.0.1 -p 4242
The authenticity of host '[127.0.0.1]:4242 ([127.0.0.1]:4242)' can't be established.
ED25519 key fingerprint is SHA256:0kTl04fMb2/XQQCnEUEqLEGhysSaPWNDR0EscMiClU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[127.0.0.1]:4242' (ED25519) to the list of known hosts.
serromer@127.0.0.1's password:
Linux serromer42 6.12.48+deb13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.48-1 (2025-09-20) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

### Conexión por fuera de la máquina





Si no tienes instalado net-tools, instálalo y luego verifica si el puerto 4242.

- ✓ ***sudo apt install net-tools*** = Es un conjunto de utilidades de red clásicas de Linux (como ifconfig, netstat, route, etc.).
- ✓ ***netstat -tuln | grep 4242*** = Muestra información sobre las conexiones de red, puertos en uso y servicios que están escuchando.
  - **-t** → muestra conexiones TCP.
  - **-u** → muestra conexiones UDP.
  - **-l** → muestra solo los puertos que están en escucha (“listening”).
  - **-n** → muestra direcciones y puertos en formato numérico (no los resuelve a nombres).

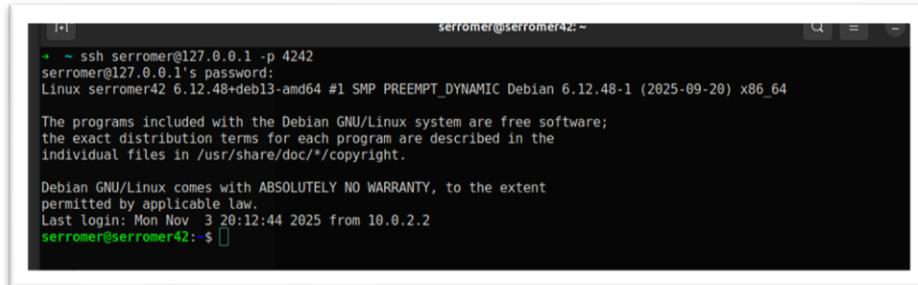
- ✓ **ss -tuln | grep 4242** = ss (de socket statistics) muestra información sobre los sockets y conexiones de red del sistema.



The image shows two separate terminal windows. The top window is titled 'sergio-alejandro@sergiodevelop:~' and displays the command 'netstat -tuln | grep 4242'. The output shows a single socket entry: 'tcp 0 0 0.0.0.0:4242 0.0.0.0:\* LISTEN'. The bottom window is titled 'sergio-alejandro@sergiodevelop:~' and displays the command 'ss -tuln | grep -i 4242'. The output is identical to the first window, showing the same socket entry.

```
sergio-alejandro@sergiodevelop:~$ netstat -tuln | grep 4242
tcp 0 0 0.0.0.0:4242 0.0.0.0:* LISTEN
sergio-alejandro@sergiodevelop:~$ ss -tuln | grep -i 4242
tcp LISTEN 0 10 0.0.0.0:4242 0.0.0.0:*
sergio-alejandro@sergiodevelop:~$
```

## Conexión SSH



The image shows a terminal window titled 'serromer@serromer42:~'. It displays the command 'ssh serromer@127.0.0.1 -p 4242'. The session starts with a password prompt for 'serromer@127.0.0.1'. After logging in, it shows the system information: 'Linux serromer42 6.12.48+deb13-amd64 #1 SMP PREEMPT\_DYNAMIC Debian 6.12.48-1 (2025-09-20) x86\_64'. It then displays the standard Debian copyright notice. Finally, it shows the last login information: 'Last login: Mon Nov 3 20:12:44 2025 from 10.0.2.2'. The prompt 'serromer@serromer42:' is visible at the bottom.

```
serromer@serromer42:~$ ssh serromer@127.0.0.1 -p 4242
serromer@127.0.0.1's password:
Linux serromer42 6.12.48+deb13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.48-1 (2025-09-20) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Nov 3 20:12:44 2025 from 10.0.2.2
serromer@serromer42:~$
```

## 7. Políticas de contraseña

**sudo nano /etc/login.defs** = Dentro de este archivo, defines los parámetros globales para todas las contraseñas del sistema.

```
serromer@serromer42:~$ ls
serromer@serromer42:~$ sudo nano /etc/login.defs
[sudo] contraseña para serromer:
```

Accedemos al archivo de configuración y realizamos los siguientes ajustes:

- La contraseña expirará cada **30 días**.
- Se establece un período mínimo de **2 días** antes de permitir un nuevo cambio de contraseña.
- Se mostrará un **aviso 7 días** antes de que la contraseña expire.

PASS_MAX_DAYS	30
PASS_MIN_DAYS	2
PASS_WARN_AGE	7

Parámetro	Significado
PASS_MAX_DAYS 30	La contraseña expira cada 30 días. El usuario debe cambiarla.
PASS_MIN_DAYS 2	El usuario no puede volver a cambiarla hasta pasados 2 días. Evita cambiarla muchas veces seguidas.
PASS_WARN_AGE 7	El sistema avisa 7 días antes de que expire la contraseña.

Estos valores se aplican automáticamente a **los nuevos usuarios** creados después del cambio.

Para aplicarlos a usuarios existentes, hay que usar el comando **chage**.

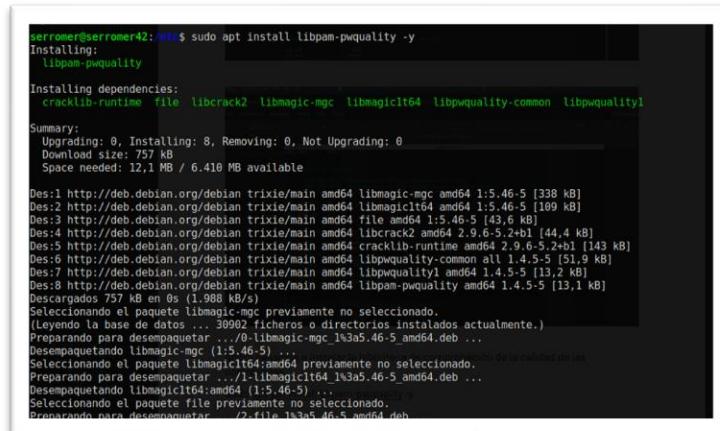
```
GNU nano 8.4          serromer@serromer42: /etc
HOME_MODE      0700
# Password aging controls:
#           Maximum number of days a password may be used.
#           Minimum number of days allowed between password changes.
#           Number of days warning given before a password expires.
#
PASS_MAX_DAYS  30
PASS_MIN_DAYS  2
PASS_WARN_AGE  7
#           Entramos al archivo y creamos las siguientes configuraciones:
#           La contraseña expira en 30 días con 2 días mínimo de cambio límite y 7 días de aviso.

# Min/max values for automatic uid selection in useradd(8)
#
UID_MIN        1000
UID_MAX        60000
# System accounts
#SYS_UID_MIN   101
#SYS_UID_MAX   999
# Extra per user uids
SUB_UID_MIN    100000
SUB_UID_MAX    600100000
SUB_UID_COUNT  65536
#
# Min/max values for automatic gid selection in groupadd(8)
#
GID_MIN        1000
GID_MAX        60000
```

## Instalar la biblioteca de calidad de contraseñas

Esta biblioteca hace que **las contraseñas sean fuertes** (mínimo de longitud, mezcla de letras, números, etc.).

- ✓ **sudo apt install libpam-pwquality -y** = Esta instala el módulo **PAM (Pluggable Authentication Module)** que se usa **en /etc/pam.d/common-password**.



```
serromer@serromer42:~$ sudo apt install libpam-pwquality -y
Installing:
  libpam-pwquality

Installing dependencies:
  cracklib-runtime file libcrack2 libmagic-mgc libmagic1t64 libpwquality-common libpwquality1

Summary:
  Upgrading: 0, Installing: 8, Removing: 0, Not Upgrading: 0
  Download size: 757 KB
  Space needed: 12,1 MB / 6.410 MB available

Des:1 http://deb.debian.org/debian trixie/main amd64 libmagic-mgc amd64 1:5.46-5 [338 kB]
Des:2 http://deb.debian.org/debian trixie/main amd64 libmagic1t64 amd64 1:5.46-5 [109 kB]
Des:3 http://deb.debian.org/debian trixie/main amd64 file amd64 1:5.46-5 [43,6 kB]
Des:4 http://deb.debian.org/debian trixie/main amd64 libcrack2 amd64 2.9.6-5.2+b1 [44,4 kB]
Des:5 http://deb.debian.org/debian trixie/main amd64 cracklib-runtime amd64 2.9.6-5.2+b1 [143 kB]
Des:6 http://deb.debian.org/debian trixie/main amd64 libpwquality-common all 1.4.5-5 [51,9 kB]
Des:7 http://deb.debian.org/debian trixie/main amd64 libpwquality1 amd64 1.4.5-5 [13,2 kB]
Des:8 http://deb.debian.org/debian trixie/main amd64 libpam-pwquality amd64 1.4.5-5 [13,1 kB]
Descargados 757 KB en 0s (1.988 kB/s)
Selezionando el paquete libmagic-mgc previamente no seleccionado.
Leyendo la base de datos... 30902 ficheros o directorios instalados actualmente.)
Preparando para despaquetar .../libmagic-mgc_1%3a5.46-5_amd64.deb ...
Despaquetando libmagic-mgc (1:5.46-5) ...
Selezionando el paquete libmagic1t64:amd64 previamente no seleccionado.
Preparando para despaquetar .../libmagic1t64_1%3a5.46-5_amd64.deb ...
Despaquetando libmagic1t64:amd64 (1:5.46-5) ...
Selezionando el paquete file previamente no seleccionado.
Preparando para despaquetar .../file_1%3a5.46-5_amd64.deb ...

```



```
serromer@serromer42:~$ ls -l | grep -i pam/*
-rw-r--r-- 1 root root 552 jun 29 19:40 pam.conf
drwxr-xr-x 2 root root 4096 nov  3 20:25 pam.d
serromer@serromer42:~$ cd pam.d
serromer@serromer42:~/pam.d$ ls
chfn  common-account  common-session      login    passwd    sshd  sudo-i
chpasswd common-auth   common-session-noninteractive newusers runuser  su   su-l
chsh  common-password cron          other     runuser-l sudo
serromer@serromer42:~/pam.d$ sudo nano Scommon-password
```

## Configurar las reglas de calidad de contraseña

**PAM (Pluggable Authentication Modules)** es el sistema modular de autenticación de Linux.

Cada servicio (login, sudo, passwd, ssh, etc.) consulta sus módulos PAM para decidir si se permite o no el acceso.

- ✓ **sudo nano /etc/pam.d/common-password** = Edita el archivo PAM.

Busca la línea que contiene **pam\_pwquality.so** y cámbiala por esta (o agrégala si no existe):

- ✓ **password requisite pam\_pwquality.so retry=3 minlen=10 ucredit=-1 lcredit=-1 dcredit=-1 maxrepeat=3 reject\_username difok=7 enforce\_for\_root**

Parámetro	Significado
<b>retry=3</b>	Permite 3 intentos para escribir una contraseña válida.
<b>minlen=10</b>	Longitud mínima: 10 caracteres.
<b>ucredit=-1</b>	Requiere al menos una letra mayúscula.
<b>lcredit=-1</b>	Requiere al menos una letra minúscula.
<b>dcredit=-1</b>	Requiere al menos un número.
<b>maxrepeat=3</b>	No permite repetir un mismo carácter más de 3 veces seguidas.
<b>reject_username</b>	Impide usar el nombre de usuario dentro de la contraseña.
<b>difok=7</b>	La nueva contraseña debe diferir al menos en 7 caracteres de la anterior.

```

GNU nano 8.4
/etc/pam.d/common-password
common-password *

# /etc/pam.d/common-password - password-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
# used to change user passwords. The default is pam_unix.

# Explanation of pam_unix options:
# The "yescrypt" option enables
# hashed passwords using the yescrypt algorithm, introduced in Debian
# 11. Without this option, the default is Unix crypt. Prior releases
# used the option "sha512"; if a shadow password hash will be shared
# between Debian 11 and older releases replace "yescrypt" with "sha512"
# for compatibility. The "obscure" option replaces the old
# "OBSCURE_CHECKS ENAB" option in login.defs. See the pam_unix manpage
# for other options.

# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password    requisite          pam_pwquality.so retry=3 minlen=10 ucredit=-1 lcredit=-1 dcredit=-1 maxrepeat=3 reject_username difok=7 enforce_for_root
password    [success=1 default=ignore]  pam_unix.so obscure use_authtok try_first_pass yescrypt
# here's the fallback if no module succeeds
password    requisite          pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password    required           pam_permit.so
# and here are more per-package modules (the "Additional" Block)
# end of pam-auth-update config

```

Este cambio de password es para tanto tu usuario como root.

- ✓ ***passwd <nombre\_usuario>*** = Te pedirá una nueva contraseña que cumpla con las reglas anteriores.
- ✓ ***sudo chage -l <nombre\_usuario>*** = Verificar las fechas de expiración y límites.

```

serromer@serromer42:~$ passwd
Cambiando la contraseña de serromer.
Actualización de la contraseña.
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña es idéntica a la anterior
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña es demasiado parecida a la anterior
Vuelva a escribir la nueva contraseña:
Vuelva a escribir la nueva contraseña:
actualizada correctamente
serromer@serromer42:~$ 

```

```

serromer@serromer42:~$ sudo chage -M 30 -m 2 -W 7 serromer
serromer@serromer42:~$ sudo chage -l serromer
Último cambio de contraseña
La contraseña caduca : dic 03, 2025
Contraseña inactiva : nunca
La cuenta caduca : nunca
Número de días mínimo entre cambio de contraseña : 2
Número de días máximo entre cambio de contraseña : 30
Número de días de aviso antes de que caduque la contraseña : 7
serromer@serromer42:~$ sudo passwd root
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
actualizada correctamente
serromer@serromer42:~$ sudo chage -l root
Último cambio de contraseña
La contraseña caduca : nunca
Contraseña inactiva : nunca
La cuenta caduca : nunca
Número de días mínimo entre cambio de contraseña : 0
Número de días máximo entre cambio de contraseña : 99999
Número de días de aviso antes de que caduque la contraseña : 7
serromer@serromer42:~$ sudo chage -M 30 -m 2 -W 7 root
serromer@serromer42:~$ sudo chage -l root
Último cambio de contraseña
La contraseña caduca : dic 03, 2025
Contraseña inactiva : nunca
La cuenta caduca : nunca
Número de días mínimo entre cambio de contraseña : 2
Número de días máximo entre cambio de contraseña : 30
Número de días de aviso antes de que caduque la contraseña : 7
serromer@serromer42:~$ 

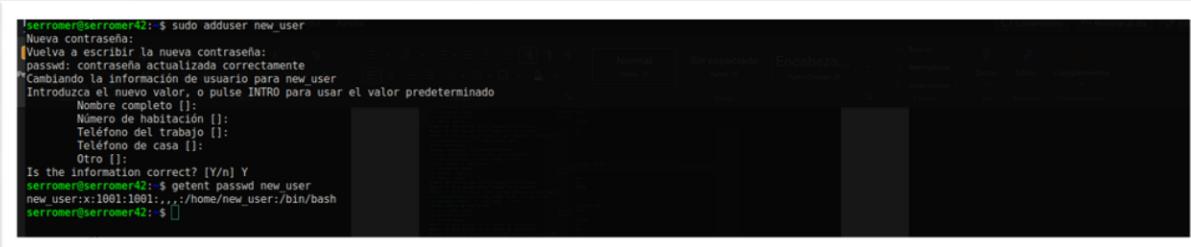
```

## 8. Creación de usuarios y contraseñas

### Crear un nuevo usuario

**sudo adduser new\_user** = Crear un nuevo usuario (Ejm: Minim0Regla#)

- Contraseña (que debe cumplir con las políticas PAM configuradas antes).
- Datos opcionales (puedes dejarlos vacíos presionando Enter).
- Crea el usuario en **/etc/passwd**
- Asigna un directorio personal **/home/new\_user**
- Crea su grupo personal
- Pregunta y configura la contraseña



```
serromer@serromer42:~$ sudo adduser new_user
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
password: contraseña actualizada correctamente
Comenzando la configuración del usuario para new_user
Introduzca el nuevo valor, o pulse INTRO para usar el valor predeterminado
Nombre completo []:
Número de habitación []:
Teléfono del trabajo []:
Teléfono de casa []:
Otro []:
Is the information correct? [Y/n] Y
serromer@serromer42:~$ getent passwd new_user
new_user:x:1001:1001:::/home/new_user:/bin/bash
serromer@serromer42:~$
```

### Crear grupos personalizados

**sudo addgroup new\_group** = Crear grupos personalizados

- **addgroup** crea nuevos grupos en el sistema (almacenados en **/etc/group**).
- Los grupos se usan para asignar permisos o roles a conjuntos de usuarios.

**sudo adduser [usuario] [grupo]** = Añade al usuario indicado al grupo.

**sudo getent passwd new\_user** = Muestra datos del usuario (UID, GID, shell, home, etc.).

### Agregando al grupo

- ✓ sudo addgroup user42
- ✓ sudo addgroup evaluating
- ✓ sudo adduser serromer user42
- ✓ sudo adduser serromer evaluating



```
serromer@serromer42:~$ sudo addgroup user42
serromer@serromer42:~$ sudo addgroup evaluating
serromer@serromer42:~$ sudo adduser serromer user42
-bash: sudo: orden no encontrada
serromer@serromer42:~$ sudo adduser serromer user42
serromer@serromer42:~$ sudo adduser serromer evaluating
fatal: The group 'evaluating' does not exist.
serromer@serromer42:~$ sudo adduser serromer evaluating
serromer@serromer42:~$ getent group user42
user42:x:1002:serromer
serromer@serromer42:~$ getent group evaluating
evaluating:x:1003:serromer
serromer@serromer42:~$
```

## 9. Monitorización con CRONTAB CONFIGURATION

### Crear el script de monitorización

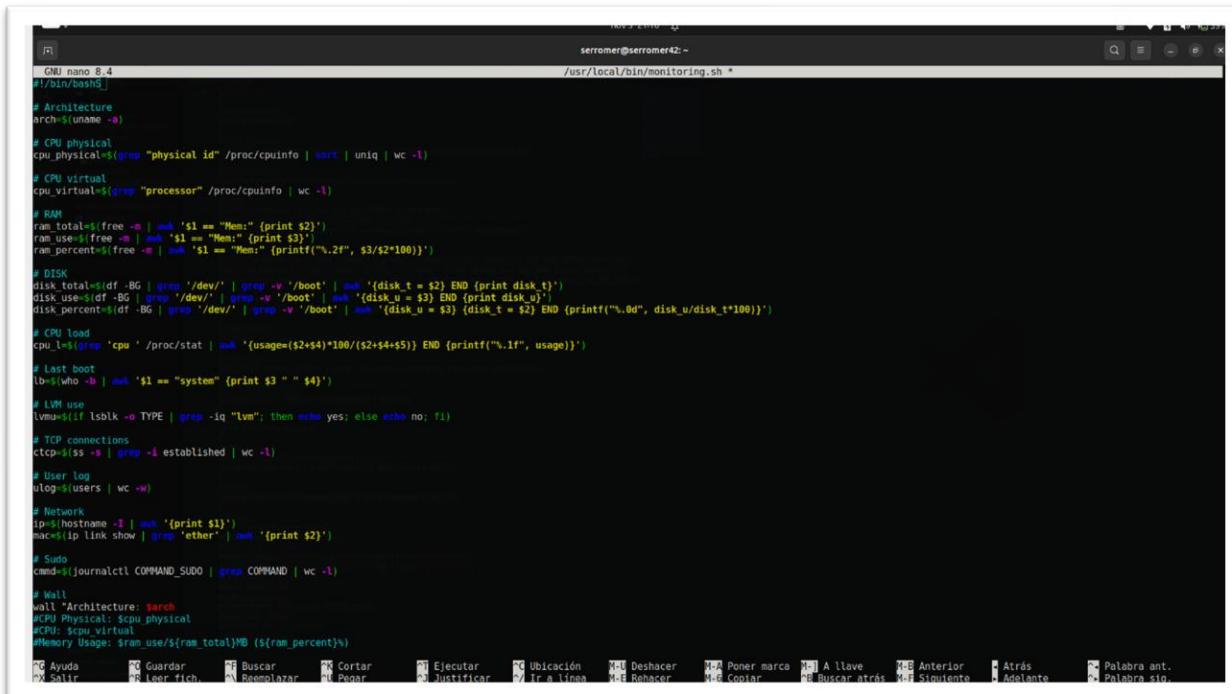
`sudo touch /usr/local/bin/monitoring.sh`

`sudo chmod 755 /usr/local/bin/monitoring.sh`

- `touch` → crea un archivo vacío (si no existe).
- `/usr/local/bin` → es la ubicación estándar para scripts personalizados del sistema.

### Contenido del script

`sudo nano /usr/local/bin/monitoring.sh` = Edita el archivo.



```
GNU nano 8.4
#!/bin/bash$1
# Architecture
arch=$(uname -a)

# CPU physical
cpu_physical=$(grep "physical id" /proc/cpuinfo | sort | uniq | wc -l)

# CPU virtual
cpu_virtual=$(grep "processor" /proc/cpuinfo | wc -l)

# RAM
ram_total=$(free -m | awk '$1 == "Mem:" {print $2}')
ram_use=$(free -m | awk '$1 == "Mem:" {print $3}')
ram_percent=$(( $(echo "scale=2; ${ram_use} / ${ram_total} * 100" | bc) )) % 

# DISK
disk_total=$(df -BG | grep '/dev/' | grep -v '/boot' | awk '{disk_t = $2} END {print disk_t}')
disk_use=$(df -BG | grep '/dev/' | grep -v '/boot' | awk '{disk_u = $3} END {print disk_u}')
disk_percent=$(( $(echo "scale=2; ${disk_u} / ${disk_t} * 100" | bc) )) %

# CPU load
cpu_load=$(grep 'cpu ' /proc/stat | awk '{usage=($2+$4)*100/($2+$4+$5)} END {printf("%.1f", usage)}')

# Last boot
lb=$(who -b | awk '$1 == "system" {print $3 " " $4}')

# LVM use
lvm=$(if lsb_release -o | grep -iq "lvm"; then echo yes; else echo no; fi)

# TCP connections
tcp=$(ss -a | grep -E established | wc -l)

# User log
ulog=$(users | wc -w)

# Network
ip=$(hostname -I | awk '{print $1}')
mac=$(ip link show | grep 'ether' | awk '{print $2}')

# Sudo
cmd=$(journalctl --since 1min --unit=system | grep COMMAND | wc -l)

# Wall
wall "Architecture: $arch
#CPU Physical: $cpu_physical
#CPU: $cpu_virtual
#Memory Usage: $ram_percent%"

# Ayuda      Guardar      Buscar      Cortar      Ejecutar      Ubicación      Deshacer      Poner marca      A llave      Anterior      Atrás      Palabra ant.
# Salir      Leer fich.  Desenlazar  Pegar      Justificar     Ir a línea      Rehacer      Copiar      Buscar atrás  Siguiente      Adelante      Palabra sig."
```

Variable	Descripción
<b>ARCH</b>	Muestra arquitectura del sistema y versión del kernel.
<b>CPU_PHYS</b>	Número de CPUs físicas.
<b>VCPU</b>	Número de procesadores virtuales (núcleos).
<b>MEM_USED</b>	Muestra el uso actual de memoria RAM.
<b>DISK_USED</b>	Muestra uso total del disco.
<b>CPU_LOAD</b>	Carga actual del procesador.
<b>LAST_BOOT</b>	Fecha y hora del último arranque.
<b>LVM_USE</b>	Indica si el sistema usa LVM (yes/no).
<b>TCP_CON</b>	Número de conexiones TCP activas.
<b>USER_LOG</b>	Usuarios conectados.
<b>IP / MAC</b>	Dirección IP y MAC del equipo.

## Permitir ejecutar el script sin pedir contraseña (sudoers)

***sudo visudo*** = Edita el archivo de configuración sudoers.

*serromer ALL=(ALL) NOPASSWD:/usr/local/bin/monitoring.sh* = Al final, añade esta línea.

**`sudo systemctl enable cron.service`** = Activar el servicio.

- ✓ Permite que el usuario **serromer** pueda ejecutar ese script con sudo sin tener que escribir la contraseña.
  - ✓ Esto es necesario porque el cron se ejecutará como **root** y debe poder correrlo sin interrupción.

```
serromer@serromer42:~$ sudo systemctl enable cron.service
Synchronizing state of cron.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable cron
serromer@serromer42:~$
```

Ahora reiniciamos el servicio y la máquina para que los cambios se hagan válidos.

***sudo /usr/local/bin/monitoring.sh*** = Probar que funciona manualmente.

```
serromer@serromer42:~$ sudo /usr/local/bin/monitoring.sh
serromer@serromer42:~$ sudo /usr/local/bin/monitoring.sh
serromer@serromer42:~$ sudo /usr/local/bin/monitoring.sh
serromer@serromer42:~$ [1]
File Machine View Input Devices Help
Mensaje de difusión general (broadcast) de root@serromer42 (pts/1) (Mon Nov 3
root@serromer42:~$ [1]
*** System Report:
Architecture : Linux serromer42 6.12.40+deb13-amd64 #1 SMP PREEMPT_DYNAMIC
NIC Debian 6.12.40-1 (2025-09-28) x86_64 GNU/Linux
CPU Physical : 1
CPU Cores (virt.) : 2
CPU Frequency : 2433/967MHz (25.13x)
Disk Usage : 2/210B (6%)
CPU Load : 100.0%
Swap Free : 914MB 2025-11-03
LVM In use :
TCP Connections : 2 ESTABLISHED
Logged Users : 0
IP Address : 10.0.2.15 | MAC 00:00:27:65:15:e1
Sudo Commands : 0
Generated on Jun 03 nov 2025 21:09:03 CET
```

## Programar su ejecución automática con CRONTAB

**Crontab** es una herramienta del sistema Linux que **permite ejecutar comandos o scripts automáticamente en momentos específicos** (cada minuto, hora, día, semana, etc.). Es como un reloj automático del sistema que ejecuta tareas por ti en los horarios que definas.

El servicio **cron** está **siempre activo** en segundo plano (como un *daemon*).

Cada minuto, cron revisa **todas las tablas de tareas (crontab)** de todos los usuarios, y ejecuta los comandos que correspondan según la hora actual.

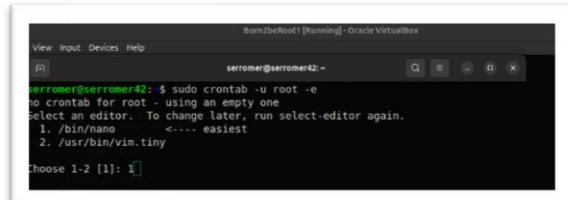
**CRONTAB = Cron Table = Tabla de tareas programadas en el tiempo.**

Comando	Descripción
<b>crontab -e</b>	Edita el crontab del usuario actual
<b>sudo crontab -u root -e</b>	Edita el crontab de root
<b>crontab -l</b>	Muestra las tareas programadas
<b>crontab -r</b>	Elimina todas las tareas del crontab
<b>systemctl status cron</b>	Verifica si el servicio cron está activo

**sudo crontab -u root -e** = Edita el crontab de root

**\*/10 \* \* \* \* /usr/local/bin/monitoring.sh** = Añade esta línea al final.

Campo	Significado
<b>*/10</b>	Cada 10 minutos
<b>*</b>	Cada hora
<b>*</b>	Cada día
<b>*</b>	Cada mes
<b>*</b>	Cualquier día de la semana
<b>/usr/local/bin/monitoring.sh</b>	Comando o script a ejecutar



Lo siguiente se ejecutará el script cada 10 minutos, mostrando la información del sistema en pantalla con wall:

A screenshot of a terminal window titled "GNOME Terminal - /tmp/crontab.1JgLNv/crontab \*". It shows the contents of a crontab file being edited with nano. The file contains several comments at the top, followed by a line: "# 0 5 \* \* 1 tar -zcf /var/backups/home.tgz /home/" and then the line "#!/bin/sh". At the bottom of the file, there is a new line added: "\*10 \* \* \* \* /usr/local/bin/monitoring.sh".

## 10. Instalación y configuración de llmp stack, lighttpd

El **LLMP stack** es el conjunto de tecnologías que trabajan juntas para construir y servir una aplicación web completa en Linux.

Mi servidor está basado en un **stack LLMP**, que significa Linux, **Lighttpd**, **MariaDB** y **PHP**. Linux es el sistema base, **Lighttpd** el servidor web, PHP el lenguaje que genera las páginas dinámicas, y **MariaDB** la base de datos. Todos están conectados mediante **FastCGI** y **PHP-FPM**. Sin esta integración, el servidor no podría ejecutar aplicaciones web dinámicas.

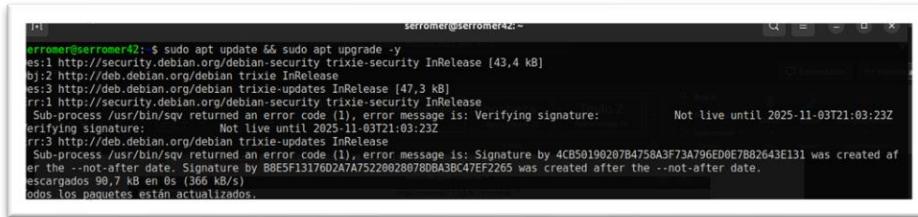
Letra	Significado	Qué hace
L	Linux	El sistema operativo base
L	Lighttpd	El servidor web (sirve las páginas)
M	MariaDB	El sistema de base de datos
P	PHP	El lenguaje de programación del lado del servidor

Diferencia con otros stacks			
Stack	Servidor Web	Base de Datos	Lenguaje
LAMP	Apache	MySQL/MariaDB	PHP
LLMP	Lighttpd	MariaDB	PHP
LEMP	Nginx	MariaDB/MySQL	PHP
MEAN	Node.js	MongoDB	JavaScript

Vamos a explicarlo con la versión **LLMP** (más ligera y moderna).

**sudo apt update && sudo apt upgrade -y** = Actualiza el sistema

Antes de instalar servicios, siempre actualizo el sistema para evitar conflictos de dependencias.



```
serromer@serromer42:~$ sudo apt update && sudo apt upgrade -y
[...]
errormer@serromer42:~$
```

## Instalar Lighttpd (servidor web ligero)

**Lighttpd** no incluye **PHP** ni **MariaDB**, así que se instala manualmente. Para que **Lighttpd** pueda ejecutar scripts PHP, configureré **FastCGI (Fast Common Gateway Interface)**, que actúa como puente entre el servidor y **PHP-FPM**. Si no instalara **PHP-FPM**, el servidor solo serviría archivos estáticos, sin ejecutar código dinámico.

**FastCGI** es una tecnología para que los servidores web ejecuten programas externos (por ejemplo, un script PHP o Python), **esto mantiene procesos PHP abiertos** en segundo plano, en resumen, **FastCGI** es un **puente** entre el servidor web (**Lighttpd**) y el intérprete de PHP (**PHP-FPM**).

**PHP-FPM (FastCGI Process Manager)** es el programa que maneja esos procesos **FastCGI** de PHP. Recibe peticiones desde **Lighttpd**. Ejecuta el código PHP, devuelve el resultado (HTML) al servidor web. Si no instalo **FastCGI** o **PHP-FPM**, entonces **Lighttpd no podrá ejecutar archivos .php**.

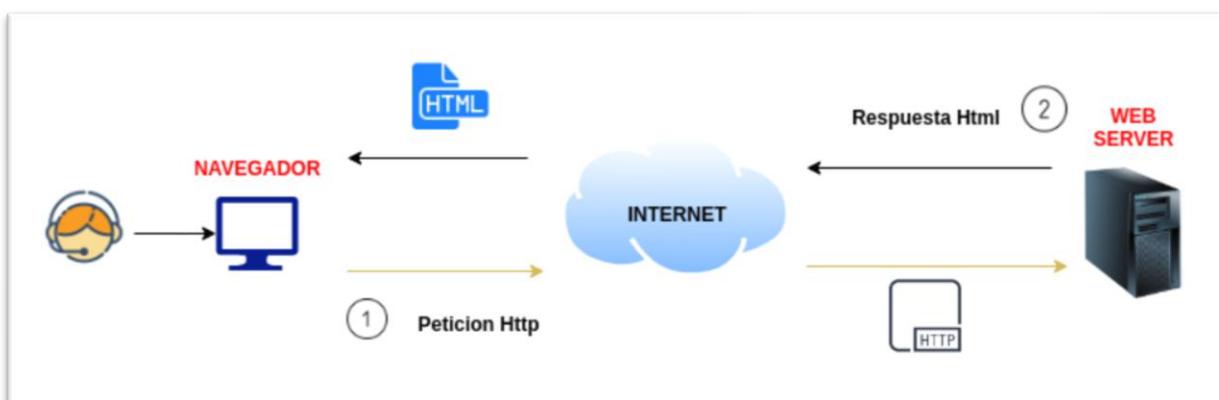
**sudo apt install lighttpd -y** = Instalación del servidor.

- **Lighttpd** (“Lighty”) es un servidor web rápido y de bajo consumo, alternativa a Apache.
- Por defecto sirve archivos desde **/var/www/html**.
- El servicio se inicia automáticamente tras la instalación.
  - ✓ **sudo systemctl status lighttpd** = Ver estado del servidor.
  - ✓ **sudo systemctl reload lighttpd** = Reinicia el servicio.
  - ✓ **sudo systemctl enable lighttpd** = Activa el servicio.
  - ✓ **sudo systemctl start lighttpd** = Empieza el servicio.
  - ✓ **curl http://localhost** = Deberías ver el texto por defecto del servidor web.

**sudo ufw allow 80** = Le estás diciendo al firewall que deje pasar conexiones web normales hacia tu servidor (**Lighttpd, Apache o Nginx**).

**sudo ufw status** = Muestra las reglas activas del firewall.

- ✓ **allow 80**, porque cuando instalas tu servidor web (**Lighttpd, Apache o Nginx**):
- ✓ El servicio escucha en el puerto 80 (**HTTP**) por defecto.
- ✓ Si el firewall está activo (**UFW habilitado**), ese tráfico se bloquearía si no lo autorizas.



## 11. Instalación y configuración de MariaDB

**MariaDB** es una base de datos relacional (RDBMS), **libre y compatible con MySQL**. Incluye el **motor de base de datos** (`mysqld`) y el **cliente de línea de comandos** (`mariadb`), es el motor de base de datos que forma parte del stack LLMP.

<code>sudo apt install mariadb-server -y</code>	= Instalar MariaDB
<code>sudo systemctl status mariadb</code>	= Verifica que el servicio esté activo.
<code>sudo mysql_secure_installation</code>	= Configurar la seguridad inicial

Durante este asistente:

- ✓ **Set root password?** → N (normalmente ya está configurada internamente en Debian)
- ✓ **Remove anonymous users?** → Y
- ✓ **Disallow root login remotely?** → Y
- ✓ **Remove test database?** → Y
- ✓ **Reload privilege tables?** → Y

### Explicación:

- Elimina usuarios anónimos.
- Impide el acceso remoto del usuario root (solo localhost).
- Borra bases de prueba que podrían ser vulnerables.
- Recarga los permisos actualizados

Ejecuto `mysql_secure_installation` para reforzar la seguridad inicial de MariaDB eliminando accesos anónimos y remotos.

### Cliente MariaDB

- ✓ **sudo mariadb** = Entrar al cliente MariaDB, abre el intérprete SQL en consola.
- ✓ **sudo mysql** = En sistemas más nuevos
- ✓ **CREATE DATABASE serromer;** = Crea una nueva base de datos llamada serromer.
- ✓ **show databases;** = Comprobar que se creó.
- ✓ **GRANT ALL ON serromer.\* TO 'serromer'@'localhost' IDENTIFIED BY 'Fuerza#123A' WITH GRANT OPTION;** = Crear un usuario con permisos.
- ✓ **FLUSH PRIVILEGES;** = Aplicar cambios, recarga la tabla de permisos para aplicar los cambios inmediatamente.
- **GRANT ALL ON serromer.\*** → da acceso completo a todas las tablas de la base serromer.
- **'serromer'@'localhost'** → el usuario solo puede conectarse desde el mismo sistema.
- **IDENTIFIED BY 'Fuerza#123A'** → establece la contraseña del usuario.
- **WITH GRANT OPTION** → le permite otorgar permisos a otros usuarios (opcional, pero común en el proyecto).

### **Probar el nuevo usuario**

**mariadb -u serromer -p** = Salir del cliente y volver a entrar con el nuevo usuario  
**SHOW DATABASES;** = Si aparece tu base de datos → todo funciona correctamente.

## 12. Instalación de PHP

Instalamos **PHP** junto con las extensiones necesarias para comunicación con **MariaDB** y procesamiento web. Luego configuré el módulo **FastCGI** de **Lighttpd** para que ejecute scripts PHP mediante **PHP-FPM**, asegurando compatibilidad de versiones. Finalmente, probé la configuración con un archivo **info.php** que confirma el correcto enlace entre **PHP** y **Lighttpd**.

Primero instalas, **PHP** con soporte para **MySQL** y otras extensiones comunes, necesarias para conectar el backend **PHP** con **MariaDB** y manejar archivos o imágenes.

**sudo apt install php-fpm php-mysql php-curl php-gd php-zip -y** = Instalar **PHP** y extensiones necesarias

**dpkg -l | grep php** = Muestra todos los paquetes PHP instalados.

**php -v** = Muestra la versión actual de PHP y confirma que se instaló correctamente.

- ✓ **php-fpm** → (**FastCGI Process Manager**): permite que PHP trabaje con Lighttpd o Nginx.
- ✓ **php-mysql** → Extensión que permite a PHP conectarse con MariaDB/MySQL.
- ✓ **php-curl** → Para hacer peticiones HTTP desde PHP.
- ✓ **php-gd** → Para manejar imágenes (usado por muchos CMS).
- ✓ **php-zip** → Para trabajar con archivos ZIP.

**php -v** = si ves (**PHP 8.2.12 (fpm-fcgi)**) -> Entonces la ruta de configuración será:

/etc/php/8.2/fpm/pool.d/www.conf

### Editar configuración FastCGI para Lighttpd

Configuré **PHP-FPM** como backend **FastCGI** para **Lighttpd**, lo que permite que el servidor web ejecute código PHP dinámico. Luego verifiqué la versión de PHP instalada y la ruta del socket para asegurar compatibilidad.

**sudo nano /etc/lighttpd/conf-available/15-fastcgi-php.conf** = Abre el archivo

**ls /etc/lighttpd/conf-available/** = Si el nombre no existe, busca el que tenga “**fastcgi**” o “**php**” en el directorio.

- ✓ Dentro del archivo, debe tener una línea similar a: “**socket**” => “**/run/php/php8.2-fpm.sock**”,
- ✓ Asegúrate de que la versión (**php8.2**) coincida con la que viste en **php -v**.

### Habilitar FastCGI y reiniciar Lighttpd

- ✓ **sudo lighty-enable-mod fastcgi**
- ✓ **sudo lighty-enable-mod fastcgi-php**
- ✓ **sudo systemctl reload lighttpd**
- ✓ **sudo systemctl status lighttpd**

### Probar PHP (Opcional)

**sudo nano /var/www/html/info.php** = Crea un archivo de prueba.

<http://localhost/info.php> = Ejecutas en navegador, acuerda agregar algo a **info.php**

## 13. WordPress SETUP

El último paso del proyecto **Born2beroot** es la instalación y configuración de WordPress, que sirve como prueba final de que tu **LLMP stack(Linux + Lighttpd + MariaDB + PHP)** funciona correctamente.

**WordPress** es un **CMS (Content Management System)**, o sea, un **Sistema de Gestión de Contenidos**. Su función principal es **permitir crear, administrar y publicar páginas web fácilmente**, sin tener que programar en HTML, PHP o SQL manualmente. Podría haber programado todo manualmente en PHP y SQL, pero con **WordPress** aprovecho una plataforma ya estructurada que se ejecuta sobre mi stack LLMP.

Descargué e instalé **WordPress** dentro del directorio **/var/www/html**. Configuré el archivo **wp-config.php** con los datos de mi base **MariaDB**, creada previamente. Luego asigné los permisos al usuario **www-data** para que **Lighttpd** pueda acceder a los archivos. Finalmente, recargué el servicio **Lighttpd** y verifiqué la instalación desde el navegador. Esto demuestra que mi **stack LLMP (Linux, Lighttpd, MariaDB, PHP)** funciona correctamente.

### Instalación

**sudo apt install wget -y** = wget permite descargar archivos desde Internet directamente desde la terminal (como el paquete de WordPress).

**sudo wget https://wordpress.org/latest.tar.gz -P /var/www/html** = Descargar la última versión de WordPress.

- ✓ **-P ruta:** indica que el archivo se guardará directamente en el directorio web.

**sudo tar -xvzf /var/www/html/latest.tar.gz -C /var/www/html** = Descomprimir WordPress

**sudo rm -rf /var/www/html/latest.tar.gz** = Eliminar el archivo comprimido

- ✓ **-C** indica en qué carpeta lo quieras descomprimir (tu raíz web).

**sudo cp -r /var/www/html/wordpress/\* /var/www/html** = Copia todo el contenido de la carpeta wordpress a la raíz (/var/www/html).

**sudo rm -rf /var/www/html/WordPress** = Borramos la carpeta vacía WordPress para dejar todo.

### Configurar WordPress

**sudo cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php** = Copias el archivo de configuración de ejemplo (wp-config-sample.php) a uno real (wp-config.php).

**sudo nano /var/www/html/wp-config.php** = Luego lo editas para conectar WordPress con tu base de datos.

#### Dentro de wp-config.php

Debes editar las líneas, WordPress usará estos datos para conectarse automáticamente a MariaDB a través de PHP.

```
define( 'DB_NAME', 'serromer' );
define( 'DB_USER', 'serromer' );
define( 'DB_PASSWORD', 'Fuerza#123A' );
define( 'DB_HOST', 'localhost' );
```

<b>DB_NAME</b>	Tu base de datos (la que creaste en MariaDB).
<b>DB_USER</b>	El usuario con permisos sobre esa base.
<b>DB_PASSWORD</b>	La contraseña de ese usuario.
<b>DB_HOST</b>	El servidor de la base de datos (en este caso, localhost).

## Ajustar permisos

`sudo chown -R www-data:www-data /var/www/html`

`sudo chmod -R 755 /var/www/html`

`sudo systemctl reload lighttpd` = Recarga la configuración del servidor web para que reconozca los nuevos archivos.

<http://localhost:8080> = Verificar en el navegador, el puerto que tengas configurado en **Lighttpd**, como 4242 si lo cambiaste.

- ✓ **www-data** es el usuario y grupo que usa **Lighttpd**.
- ✓ **chown** cambia el propietario de los archivos (para que el servidor web tenga acceso).
- ✓ **chmod 755** da permisos de lectura y ejecución a todos, escritura solo al dueño.



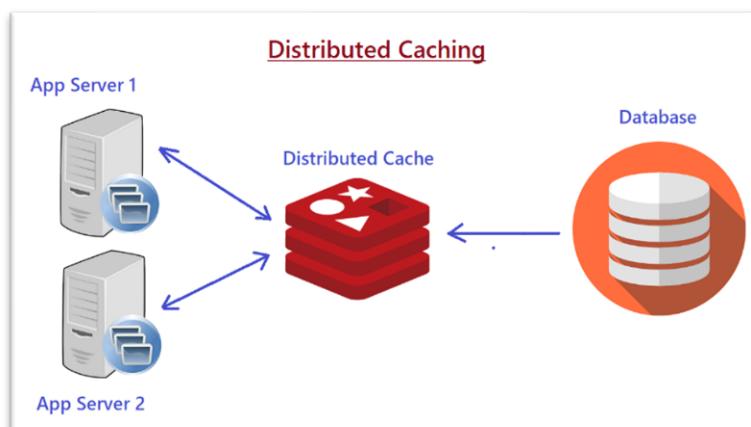
## 14. Configurar un servicio de WordPress a tu elección

### Redis

He configurado **Redis** como servicio adicional para mi servidor **WordPress**. **Redis** actúa como un sistema de caché en memoria, reduciendo las consultas directas a la base de datos y mejorando el rendimiento.

**Redis (Remote Dictionary Server)** es una **base de datos en memoria** (RAM) de tipo **key-value**, extremadamente rápida. Se usa principalmente como **sistema de caché** para acelerar aplicaciones como WordPress. **Redis** guarda datos temporalmente en memoria (RAM) para evitar que WordPress consulte la base de datos **MariaDB** cada vez que alguien carga una página.

Lo conecté mediante la extensión **php-redis** y configuro las credenciales en **wp-config.php**. Finalmente, activé la integración desde el plugin **Redis Object Cache**. Puedo comprobar su funcionamiento con el comando **redis-cli ping**, que devuelve **PONG**.



### Instalación

**sudo apt install redis-server -y** = Instala el servicio Redis en tu sistema.

**sudo apt install php-redis -y** = Extensión PHP para que WordPress pueda comunicarse con Redis.

**sudo systemctl restart lighttpd** = Reinicias **Lighttpd** para que cargue la extensión.

### Configurar Redis

**sudo nano /etc/redis/redis.conf** = Abre el archivo de configuración.

**requirepass Fuerza#123A** = Busca y modifica las siguientes líneas, añade una contraseña para proteger el acceso a Redis (importante por seguridad).

### Activa snapshots (copias automáticas del estado de Redis)

Save 900 1  
Save 300 10  
Save 60 10000

- ✓ **Guarda una copia del estado** si hubo 1 cambio en 900 segundos, 10 cambios en 300 segundos, o 10.000 cambios en 60 segundos.
- ✓ **Sirve como respaldo en caso de apagado del servidor.**

### Asegurar el directorio de Redis

**sudo mkdir -p /var/lib/redis** = Crea el directorio donde Redis guardará sus snapshots.

**sudo chown redis:redis /var/lib/redis** = Asigna permisos al usuario redis (igual que con www-data para Lighttpd).

**sudo systemctl restart redis-server** = Reinicia el servicio para aplicar los cambios.

### Configurar WordPress para usar Redis

**sudo nano /var/www/html/wp-config.php** = Abre tu configuración de WordPress.

Añade estas líneas debajo de las configuraciones de base de datos:

```
define('WP_REDIS_HOST', '127.0.0.1');  
define('WP_REDIS_PASSWORD', 'Fuerza#123A');
```

- ✓ **WP\_REDIS\_HOST** indica la dirección del servidor Redis (en este caso, local).
- ✓ **WP\_REDIS\_PASSWORD** es la contraseña que configuraste en **redis.conf**.

### Prueba que Redis funcione correctamente

**redis-cli** = Abre el cliente Redis

**auth Fuerza#123A** = Ejecuta

**ping**

**PONG** = Resultado esperado, esto confirma que Redis está funcionando y acepta conexiones con autenticación.

## Integrar con WordPress

Para que **WordPress** use **Redis**, necesitas un **plugin** que conecte su sistema de caché con el servicio Redis.

**`http://localhost:8080/wp-admin`** = Accede al panel de WordPress desde tu navegador

**Plugins** → Añadir nuevo

**Redis Object Cache** = Busca e instala

**Ajustes → Redis** = Una vez instalado, ve a

**Enable Object Cache** = Haz clic

- ✓ Activa la caché de objetos en WordPress.
- ✓ Los datos dinámicos (consultas, resultados, sesiones) se guardan temporalmente en Redis.
- ✓ Acelera las páginas hasta 2–3 veces.

## Otros servicios

Servicio	Función principal	Qué hace / Para qué sirve	Relación con WordPress o servidor
<b>Redis</b>	Caché en memoria	Guarda datos temporales (consultas, sesiones, objetos) en RAM.	Acelera WordPress y reduce carga de base de datos.
<b>Memcached</b>	Caché en memoria distribuida	Similar a Redis, pero más simple; guarda pares clave-valor.	Alternativa a Redis; mejora rendimiento.
<b>Fail2Ban</b>	Seguridad	Analiza logs y bloquea IPs que intentan ataques (SSH, WordPress, etc.).	Protege tu servidor contra fuerza bruta.
<b>ClamAV</b>	Antivirus	Escanea archivos del servidor en busca de malware.	Protege sitios o subidas de archivos en WordPress.
<b>NTP / Chrony</b>	Sincronización de hora	Mantiene la hora del servidor exacta.	Evita errores de logs y certificados.
<b>Postfix</b>	Servidor de correo	Envía correos desde tu servidor (p. ej., notificaciones WordPress).	Permite enviar correos de contacto o recuperación.
<b>Certbot (Let's Encrypt)</b>	HTTPS gratuito	Instala certificados SSL para tener HTTPS.	Mejora seguridad y SEO del sitio WordPress.
<b>phpMyAdmin</b>	Interfaz web para MySQL/MariaDB	Permite gestionar bases de datos desde navegador.	Facilita la administración de WordPress sin terminal.
<b>ElasticSearch</b>	Búsquedas avanzadas	Indexa contenido para mejorar búsquedas internas.	Mejora la búsqueda en sitios grandes o e-commerce.
<b>Prometheus + Grafana</b>	Monitorización	Recolecta métricas del sistema y las muestra gráficamente.	Te deja monitorear el rendimiento de tu servidor.
<b>OpenVPN / WireGuard</b>	VPN	Crea una red privada segura para acceder al servidor.	Aumenta la seguridad de acceso al entorno.
<b>Docker</b>	Contenedores	Permite correr servicios aislados y reproducibles.	Facilita la gestión y despliegue de entornos WordPress.
<b>Nginx (en lugar de Lighttpd)</b>	Servidor web	Alternativa moderna y potente a Lighttpd o Apache.	Puede usarse como proxy inverso o reemplazo completo.
<b>Monit</b>	Supervisión de servicios	Monitorea y reinicia automáticamente servicios caídos.	Asegura que Redis, MariaDB o Lighttpd estén siempre activos.
<b>ufw / iptables</b>	Firewall	Controla los puertos abiertos del servidor.	Protege acceso al SSH, HTTP, y servicios.

## 15. Generando SIGNATURE.TXTDAD

La entrega consiste en un único archivo llamado signature.txt en la raíz de tu repositorio Git. Este archivo debe contener la firma SHA1 de tu disco virtual de la máquina virtual (VM) utilizada para el proyecto.

### 1. Ubicación del Disco Virtual

Primero, debes encontrar dónde está guardado el archivo de disco de tu máquina virtual. La ruta depende de tu sistema operativo y software de virtualización:

- **Windows (VirtualBox):** %HOMEDRIVE%\%HOMEPATH%\VirtualBox VMs\
- **Linux (VirtualBox):** ~/VirtualBox VMs/
- **Mac M1 (UTM):** ~/Library/Containers/com.utmapp.UTM/Data/Documents/
- **macOS (VirtualBox):** ~/VirtualBox VMs/

El archivo de disco suele tener la extensión **.vdi** (VirtualBox) o **.qcow2** (UTM).

### 2. Obtención de la Firma SHA1

Una vez localizado el archivo del disco virtual (ej. rocky\_serv.vdi), debes calcular su **firma digital** utilizando el algoritmo **SHA1**. Esto se hace mediante un comando en la terminal:

Sistema Operativo	Ejemplo de Comando	Archivo de Disco
<b>Windows</b>	certUtil -hashfile **rocky_serv.vdi** sha1	.vdi
<b>Linux</b>	sha1sum **rocky_serv.vdi**	.vdi
<b>Mac M1 (UTM)</b>	shasum **rocky.utm/Images/disk-0.qcow2**	.qcow2
<b>macOS</b>	shasum **rocky_serv.vdi**	.vdi

El resultado será una cadena de 40 caracteres hexadecimales (ejemplo: 6e657c4619944be17df3c31faa030c25e43e40af).

### 3. Entrega

1. **Crea** un archivo llamado **signature.txt**.
2. **Pega** el resultado de la firma SHA1 dentro de este archivo.
3. **Coloca** signature.txt en la **raíz** de tu repositorio Git.
4. **Sube y confirma** (push) el archivo a Git.

### 4. Evaluación

Durante la defensa o revisión, se ejecutará el mismo comando de firma SHA1 sobre tu máquina virtual **física** (la que uses para la defensa). El resultado debe ser **exactamente idéntico** al contenido de tu archivo signature.txt. **Si no coinciden, la nota es 0.**

## Consejos Clave

- **Verifica Dblemente el Archivo:** Asegúrate de que el archivo del que obtienes la firma SHA1 es **el correcto y final** de tu proyecto.
- **Cuidado con las Modificaciones:** **Cualquier cambio** en la VM (como encenderla, instalar un paquete, crear un archivo, etc.) después de obtener la firma **cambiará la firma SHA1** del archivo de disco.

**Solución Recomendada:** Una vez que el proyecto esté **finalizado y antes de la entrega, apaga** la VM, obtén la firma SHA1, crea signature.txt y **no vuelvas a modificar esa VM** hasta la defensa.

### Alternativas para Evitar el Cambio de Firma:

**Duplicar la VM:** Crea una copia de la VM final, y usa la copia para cualquier prueba o modificación posterior, dejando la original inalterada para la evaluación. **Usar Save State:** Guarda el estado (save state) en lugar de apagarla, aunque duplicarla es más seguro.

- **Revisa el Formato:** signature.txt **solo debe contener la cadena SHA1** (40 caracteres), sin espacios, saltos de línea, nombres de archivo ni el comando utilizado.
- **Ubicación de Entrega:** El archivo **signature.txt** debe estar en el **directorio principal** de tu repositorio Git.

## 16. Pasos prohibidos y verificaciones

### 16.1. Instalar interfaz gráfica



Como consiste en configurar un servidor, deberás instalar el número mínimo de servicios. Por este motivo, una interfaz gráfica no tiene sentido. Está prohibido por tanto instalar X.org o cualquier servidor gráfico equivalente. En caso de hacerlo, tu nota será 0.

### 16.2. Verificar última versión

Deberás elegir como sistema operativo la última versión estable de **Debian** (no testing/unstable), o la última versión estable de **Rocky**. Se recomienda encarecidamente **Debian** si no tienes experiencia en administración de sistemas.

### 16.3. Verificar particiones (Obligatoria)

```
wil@wil:~$ lsblk
NAME           MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda            8:0    0   8G  0 disk
└─sda1          8:1    0 487M  0 part  /boot
  └─sda2          8:2    0   1K  0 part
    └─sda5          8:5    0 7.5G  0 part
      └─sda5_crypt 254:0    0 7.5G  0 crypt
        ├─wil--vg-root 254:1    0 2.8G  0 lvm   /
        ├─wil--vg-swap_1 254:2    0 976M  0 lvm   [SWAP]
        └─wil--vg-home 254:3    0 3.8G  0 lvm   /home
sr0           11:0    1 1024M 0 rom
wil@wil:~$ _
```

### 16.4. Preguntas



Durante la defensa, se te harán unas preguntas sobre el sistema operativo que has elegido. Debes saber, por lo tanto, las diferencias entre aptitude y apt, o qué son SELinux y AppArmor. En definitiva, ¡entiende lo que estás utilizando!

## 16.5. Revisar SSH

El servicio SSH se ejecutará obligatoriamente en el puerto 4242 de tu máquina virtual. Por seguridad, no debe ser posible conectarte a través de SSH como root.



El uso de SSH será comprobado durante la defensa creando un nuevo usuario. Por lo tanto, debes entender cómo funciona.

## 16.6. Revisar UFW o FIREWALLD

Debes configurar tu sistema operativo con el firewall UFW, (o firewalld en Rocky) dejando solamente el puerto 4242 abierto en tu máquina virtual.



Tu firewall debe estar activo cuando ejecutes la máquina virtual.  
Para Para Rocky, debes usar firewalld en lugar de UFW

## 16.7. Revisión 1

- El **hostname** de tu máquina virtual debe ser tu login terminado en 42 (por ejemplo, wil42). Deberás modificar este **hostname** durante tu evaluación.
- Debes implementar una política de contraseñas fuerte.
- Debes instalar y configurar **sudo** siguiendo reglas estrictas.
- Además del usuario root, un usuario con tu login como nombre debe existir.
- Este usuario debe pertenecer a los grupos **user42** y **sudo**.

Para configurar una política de contraseñas fuerte, deberás cumplir los siguientes requisitos:

- Tu contraseña debe expirar cada 30 días.
- El número mínimo de días permitido antes de modificar una contraseña deberá ser 2.
- El usuario debe recibir un mensaje de aviso 7 días antes de que su contraseña expire.
- Tu contraseña debe tener como mínimo 10 caracteres de longitud. Debe contener una mayúscula, una minúscula y un número. Por cierto, no puede tener más de 3 veces consecutivas el mismo carácter.



Después de preparar tus archivos de configuración, deberás cambiar la contraseña de todas las cuentas presentes en la máquina virtual, root incluida.

Para configurar una contraseña fuerte para tu grupo sudo, debes cumplir con los siguientes requisitos:

- Autenticarte con sudo debe estar limitado a tres intentos en el caso de introducir una contraseña incorrecta.
- Un mensaje personalizado de tu elección debe mostrarse en caso de que la contraseña introducida sea incorrecta cuando se utilice sudo.
- Para cada comando ejecutado con sudo, tanto el input como el output deben quedar archivados en el directorio /var/log/sudo/.
- El modo TTY debe estar activado por razones de seguridad.
- Por seguridad, los directorios utilizables por sudo deben estar restringidos. Por ejemplo:  
`/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin`

## 16.8. Creación usuario y grupo en tu delante



Durante la defensa, deberás crear un usuario y asignárselo a un grupo.

## 16.9. Script Monitoring.sh