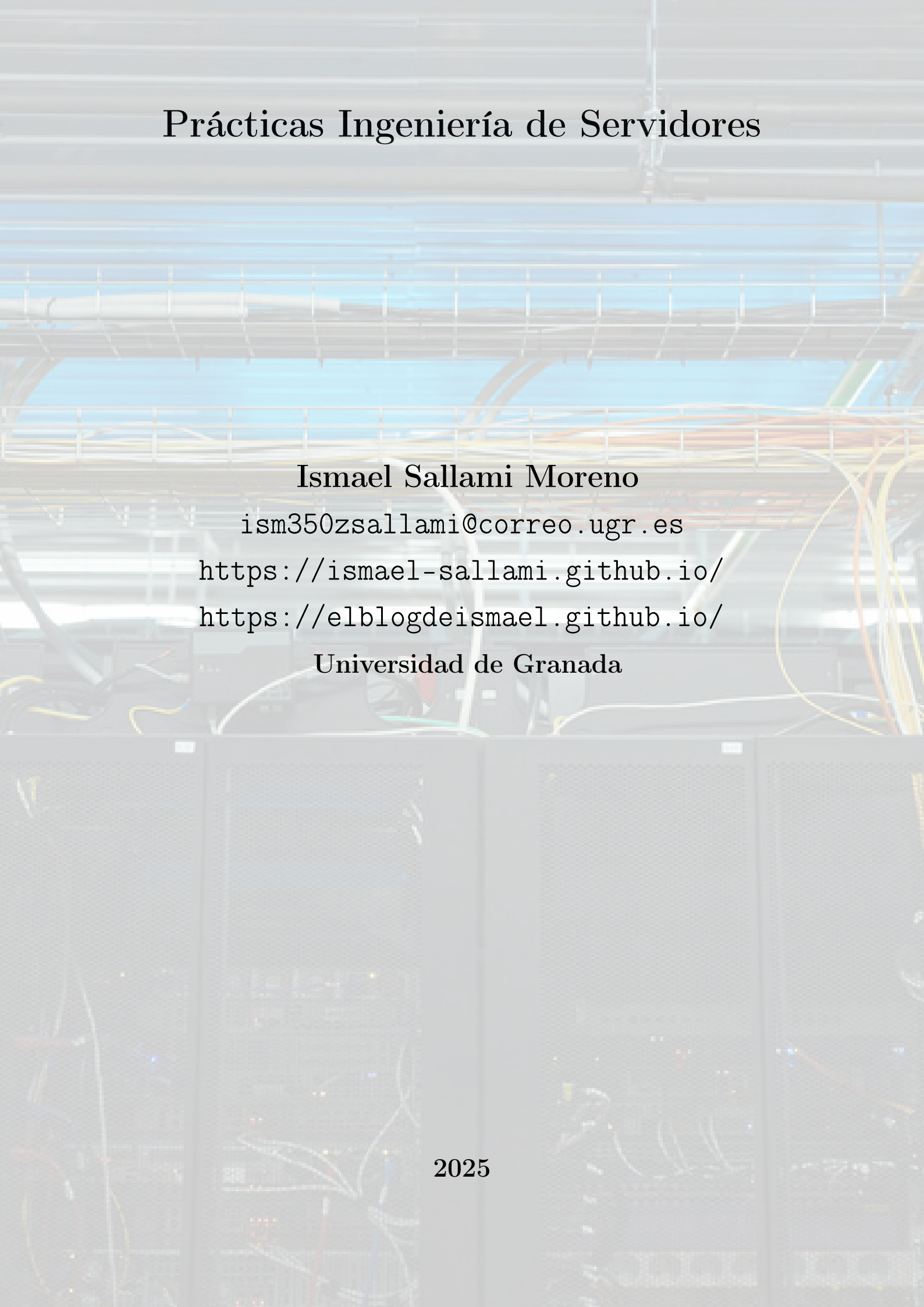
Prácticas Ingeniería de Servidores



Ismael Sallami Moreno

ism350zsallami@correo.ugr.es

https://ismael-sallami.github.io/

https://elblogdeismael.github.io/

Universidad de Granada

2025

Licencia

Índice general

Parte I

Apuntes de Clase

Capítulo 1

Bloque 1

1.1. Ping y SSH

Sobre esta parte no decidí tomar apuntes en clase debido a que son conceptos vistos en la Asignatura de Fundamentos de Redes, por lo que consideré que no era necesario. De todas formas en la parte de Prácticas (Resolución) se comenta paso a paso lo que se hizo en esta parte.

1.2. LVM y RAID

1.2.1. LVM (Logical Volume Manager)

- **Discos y particiones**:
  - `sda`: Primer disco del sistema.
  - Luego pueden existir `sdb`, `sdc`, etc.
- Consideraciones importantes:
  - Si el directorio `/boot` se llena, el sistema podría fallar al arrancar debido a la falta de espacio disponible.
  - Se recomienda evitar el uso de `swap` en sistemas virtualizados, ya que reduce el rendimiento.

1.2.2. Directorios base en Linux

- /**boot**: Contiene los archivos de arranque del sistema.
- /**etc**: Almacena los archivos de configuración del sistema operativo.
- /**dev**: Contiene archivos especiales que representan dispositivos del sistema.
- /**mnt**: Punto de montaje temporal para sistemas de archivos.
- /**var**: Contiene datos variables del sistema, como logs y archivos temporales. Puede crecer mucho, por lo que se recomienda monitorearlo.

1.2.3. RAID (Redundant Array of Independent Disks)

RAID es una tecnología que permite combinar múltiples discos para mejorar la redundancia y/o el rendimiento del sistema de almacenamiento.

1.2.4. Ventajas

- Permite unir volúmenes de almacenamiento.
- Mejora el rendimiento si los discos están en buses distintos, ya que permite acceso en paralelo.

1.2.5. Niveles de RAID

- **RAID 0 (Striping)**:
  - Divide los datos en bloques y los distribuye entre varios discos.
  - **Problema**: Si un disco falla, se pierde toda la información.
  - Se usa poco en la práctica debido a su baja robustez.
- **RAID 1 (Mirroring)**:
  - Duplica los datos en dos o más discos.

- Si un disco falla, el otro sigue funcionando con los mismos datos.
- Aporta robustez al sistema.
- **Problema**: Se paga el doble en almacenamiento.
- Se usa frecuentemente en `/boot` para garantizar que el sistema pueda arrancar en caso de fallo de un disco.

- **RAID 5 (Paridad distribuida)**:
  - Se distribuyen los datos y la paridad entre todos los discos.
  - Si un disco falla, se pueden recuperar los datos utilizando la paridad.
  - **Problema**: Puede reducir el rendimiento debido al tiempo necesario para calcular la paridad.
  - **Ventaja**: Equilibra costos, robustez y capacidad de recuperación.
  - Se puede usar un disco de repuesto que entra en acción si uno falla.

1.2.6. Tipos de RAID

- **RAID por Hardware**:
  - Utiliza un controlador RAID físico.
  - Es más eficiente y transparente para el sistema operativo.
- **RAID por Software**:
  - Administrado por el sistema operativo.
  - Puede ser modificado por el administrador, lo que representa un riesgo.
  - Requiere más recursos del sistema.

1.2.7. Ejercicio Opcional: Configuración de RAID1 para /var

1.2.8. Objetivo

El objetivo es proporcionar a `/var` un respaldo frente a fallos mediante la creación de un RAID1. Para ello, debemos montar un RAID1 y mover `/var` dentro de este.

1.2.9. Pasos a seguir

1. **Creación de discos virtuales en la máquina virtual (MV)**
   - Se crean dos discos: `raid1` y `raid2`.
   - Usamos `lsblk` para verificar la existencia de `sdb` y `sdc`.
2. **Instalación de `mdadm`**
   - `sudo dnf provides mdadm` (para verificar qué paquete lo proporciona).
   - `sudo dnf install mdadm` (para instalarlo).
3. **Creación del RAID1**

- `sudo mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb /dev/sdc`
- Aparecerá un *warning* sobre la creación de metadatos, confirmamos con "sí".
- Para monitorear la sincronización del RAID: `watch -n 1 more /proc/mdstat`
- Luego, verificamos con `lsblk`.

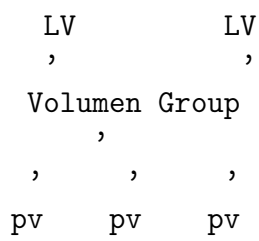
4. **Prueba de fallo de disco**
   - Podemos simular la falla de un disco con: `echo 1 > /dev/sdb`
   - Se observa que `md0` sigue funcionando al ser un *mirror*.

1.2.10. LVM (Logical Volume Manager)

1.2.11. Conceptos Clave

- **LV (Logical Volume)**
- **VG (Volume Group)**
- **PV (Physical Volume)**

La estructura básica de LVM es la siguiente:



Un **Logical Volume (LV)** puede expandirse tomando espacio libre de otros discos o estructuras.

1.2.12. Visualización de LVM

- Para ver los discos que son de tipo LVM, usamos `lsblk` en la columna de *type*.
- Para ver opciones de `pv`, `vg` y `lv`, usamos `tab-completion`.

1.2.13. Consideraciones sobre Volumen Groups

Si un **Volume Group** contiene un disco magnético, un SSD y un RAID, el acceso a los datos puede no ser homogéneo. Por ello, el volumen lógico debe montarse sobre el RAID.

1.2.14. Creación del LVM sobre el RAID

1. Convertir el RAID en un Physical Volume: `pvcreate /dev/md0`
2. Crear un Volume Group llamado `raid1`: `vgcreate raid1 /dev/md0`
3. Crear un Logical Volume de 10GB dentro del Volume Group: `lvcreate -L 10G -n rvar raid1`

1.2.15. Movimiento de /var al RAID

Después de crear el volumen lógico, debemos mover `/var` dentro del RAID:

- El nombre del volumen dentro del RAID es `/dev/raid1/rvar`.
- También puede aparecer como `/dev/mapper/raid1-rvar`, ya que son sinónimos creados mediante enlaces simbólicos.

1.2.16. Resolución de Problemas

1.2.17. Problema 1: Configurar el sistema de archivos en `rvar`

Actualmente, el volumen lógico `rvar` está vacío y sin un sistema de archivos. Para solucionarlo, seguimos estos pasos:

1. **Seleccionar un sistema de archivos:** Los sistemas de archivos recomendados son **ext4** y **xfs**, ya que son transaccionales y previenen la corrupción de datos en caso de fallos.
2. **Verificar los sistemas de archivos soportados:** Ejecutamos el siguiente comando para listar los sistemas disponibles en el kernel:

   ```
   ls /lib/modules/$(uname -r)/kernel/fs
   ```

3. **Formatear el volumen lógico con `ext4`:**

   ```
   mkfs.ext4 /dev/mapper/raid1-rvar
   ```

1.2.18. Problema 2: Montar y trasladar /var al nuevo volumen

El volumen lógico `rvar` debe ser montado en el sistema y debemos trasladar `/var` sin perder datos. Para ello:

1. **Montar el volumen lógico:** Primero, montamos `rvar` en un directorio temporal:

   ```
   mount /dev/mapper/raid1-rvar /mnt/
   ```

   Podemos verificar con:

```
mount
```

Al revisar `/mnt/`, aparecerá el directorio `lost+found`, indicando que el sistema de archivos está activo.

2. **Cambiar a modo mantenimiento:** Para evitar la pérdida de datos al copiar `/var`, debemos entrar en el **runlevel1**, también conocido como **modo mantenimiento**. Esto se hace con:

```
systemctl isolate runlevel1.target
```

Podemos confirmar el estado con:

```
systemctl status
```

3. **Copiar los datos de /var a /mnt/**: Copiamos todo el contenido de `/var` manteniendo atributos con:

```
cp -a /var/* /mnt/
```

4. **Crear un respaldo de /var:** Antes de reemplazar `/var`, hacemos una copia de seguridad por si algo falla:

```
mv /var /var_old
```

5. **Desmontar /mnt y montar el nuevo /var:**

```
umount /mnt
mkdir /var
mount /dev/mapper/raid1-rvar /var
```

Podemos verificar con:

```
df -h
```

6. **Hacer el montaje permanente en `/etc/fstab`:** Si reiniciamos ahora, la configuración se perdería. Para evitarlo, editamos el archivo `/etc/fstab` y agregamos la siguiente línea:

```
/dev/mapper/raid1-rvar      /var       ext4     defaults     0 0
```

7. **Probar la configuración antes de reiniciar:**

```
mount -a
systemctl daemon-reload
mount
```

Si todo está correcto, reiniciamos el sistema.

1.2.19. Interpretación de `lsblk`

El comando `lsblk` nos permite visualizar la estructura de almacenamiento del sistema. Es importante identificar:

- **Discos físicos y particiones**: Aparecen como `sda`, `sdb`, `sdc`, etc.
- **Volúmenes lógicos**: Se muestran bajo `/dev/mapper/`.
- **SR0**: Indica la unidad de CD-ROM.

1.3. Firewall + SSHD

Ya sabemos de otras Asignaturas que un **firewall** es un sistema que controla el tráfico de red, permitiendo o bloqueando ciertas conexiones. En Linux, el firewall más común es **iptables**.

El comando que tenemos que aprender a gestionar es `firewall-cmd`, tiene diversas opciones como status, state. Otra forma para ver si esta arrancado es `systemctl status firewall`.

Otras opciones: `firewall-cmd --list-all`, en este comando debemos de ver los filtros que estan definidos, en services podemos ver los servicios que están habilitados.

Con sudo firewall-cmd –add-service=http, cuando lo listo otra vez, se añade el puerto http en la opción services. Cuando se hace con firewall-cmd, se hace dinámica, por lo que si reiniciamos esto se pierde. Para que se quede debemos de ejecutar el comando `sudo fire firewall-cmd --runtime-to-permanent`. De forma alternativa, podemos ejecuar primero la opción `--permanent` y luego `--add-port=443/http`, pero si lo listamos no lo lleva a memoria, no hay ninguna opción que lo deje permanente y que lo lleve a memoria. Tenemos dos opciones o trabajar con servicios o llevarlo a memoria.

Si queremos conocer los servicios, podemos ejecutar `firewall-cmd --get-services` .

Los firewalls en Linux son muy eficientes debido a que en este SO se trabaja con servicios, por lo que se puede controlar el tráfico de forma muy precisa. Al implementarse a nivel de Kernel ayuda a que sea muy eficiente. *No se va a preguntar nada sobre iptables.*

Debemos de usar el programa `nmap` ya que nos dice que puertos están abietos en un servidor. Para instalarlo, usamos `sudo dnf install nmap`. En el guión debemos de mirar la referenia número 31.

1.3.1. Ejercicio Opcional

Debemos de instalar un servicio HTTP, se recomienda Apache. Podemos escanear los 100 puertos mas usados con el comando `nmap -F localhost`. Para instalar Apache, usamos `sudo dnf install httpd`. Para arrancar el servicio, usamos `sudo systemctl start httpd`. Para comprobar que está arrancado, usamos `sudo systemctl status httpd`. Para que arranque en el inicio, usamos `sudo systemctl enable httpd`. Para comprobar que está escuchando en el puerto 80, usamos `sudo ss -tulnp | grep 80`.

Debemos de poder acceder desde el navegador de la máquina anfitrión.

1.3.2. SSH

Es un programa de terminal remoto, ya lo hemos usado en Asignaturas anteriormente. Antes se usaba Telnet, en este se especifica la dirección IP y el puerto, pero no era seguro, por lo que cualquier man in the middle podía interceptarlo y suplantar la identidad. Además, todas las respuestas que se producían eran en abierto, los `cat`,... En cambio, SSH es seguro, ya que se cifra la información. Este hace lo mismo que Telnet, pero de forma segura. Se recomienda hacerlo `ssh usuario@ip`. O bien sustituir la IP por el nombre del dominio. Para cerrar la conexión, usamos `exit`.

Ya hemos estudiado en otras Asignaturas el tema de criptografía en cuanto al cifrado simétrico y asimétrico. En especial, en Fundamentos de Redes. Por si acaso, vamos a hacer una pequeña introducción. Tenemos dos tipos de cifrado:

- **Cifrado simétrico**: Se usa una clave para cifrar y descifrar. El problema es que si alguien intercepta la clave, puede descifrar todo. Por ello, se usa poco.

- **Cifrado asimétrico**: Se usa una clave pública para cifrar y una privada para descifrar. La clave pública se puede compartir, pero la privada no. Se usa para firmar documentos, ya que si se cifra con la clave privada, solo se puede descifrar con la clave pública.

Debemos de destacar los conceptos de **autenticación** y **autorización**. La autenticación es el proceso de verificar la identidad de un usuario, mientras que la autorización es el proceso de verificar si un usuario tiene permiso para acceder a un recurso. Además, del uso de certificados digitales, que son un tipo de credencial que se usa para autenticar la identidad de un usuario o un servidor. Este se consigue a través de una entidad certificadora, usando una clave pública y privada (Big Brother).

Algoritmos :

- Llave simétrica: DES(tenemos dos llaves, una que es privada y otra que es pública, lo que se cifra con la privada solo se puede descifrar con la pública, por dentro se gestiona en base a números primos muy grandes).

- Llave asimétrica: RSA (Se guarda la privada, mientras que la pública se comparte, estando incluso compartida en la propia web, de manera que la persona que quiera compartir contigo usa la pública para codificar el texto y te lo manda codificado, de esta manera si hay un man in the middle lo ve cifrado, solo la persona con la llave privada puede ver su contenido).

Otro uso de la llave privada es la *firma*, de esta manera se garantiza la confidencialidad y la autenticación del mensaje debido a que es confidencial debido a que solo se puede descifrar con la llave pública, y autenticidad debido a que solo se puede cifrar con la llave privada, la cual solo conoce esa entidad.

Firma Digital: Usando Hash SHA256, se cifra con la llave privada y se envía el mensaje cifrado y el mensaje en abierto. La persona que recibe el mensaje cifrado con la llave privada, lo descifra y lo compara con el mensaje en abierto, si son iguales, se garantiza la autenticidad del mensaje. Ejemplo de ello puede ser el minado de monedas como es el Bitcoin. Además, otro ejemplo de ello es cuando descargamos algo y podemos verificar que es el del propio distribuidor usando el Hash que me proporciona. *Por ende, podemos cosiderar que un HASH + llave privada es el certificado digital.* Para asegurarnos de que la llave pública es de quien dice ser debemos de ver el certificado digital, como hemos mencionado anteriormente. Este proceso para cuando encontramos un certificado digital que es de confianza, ya que si no lo es, no podemos asegurar que la llave pública sea de quien dice ser. Para ver los certificados, vemos que en Firefox, accedemos a settings y buscamos *certificados*, podemos ver las entidades que nuestro browser reconoce.

Advertencia

El tema de certificados y demás al ser materia de Fundamentos de Redes, no se va a preguntar en el examen.

Nos conectamos a un ordenador remoto mediante ssh, de esta manera le decimos que nos queremos conectar, el te envía la llave pública, es decir, se va a su configuración, recupera la llave y te la envía. A continuación, le enviamos nuestras creedenciales cifradas con la llave pública, y cuando le llega, las descifra con la llave privada y comprueba si son correctas. Si lo son, nos deja entrar. Si no lo son, nos dice que no podemos entrar. Además, de esta manera se prevee que un *man-in-the-middle* no pueda interceptar las credenciales.

Podemos meterle una entrada en el fichero de host, de esta manera cuando nos conectamos a un servidor, nos conectamos a una IP, pero si le metemos una entrada en el fichero de host, le decimos que esa IP es un nombre, de esta manera cuando nos conectamos a ese nombre, nos conectamos a esa IP. Para ello, debemos de editar el fichero `/etc/hosts` y añadir la IP y el nombre del servidor, así es más cómdo, para ello en Linux es `ssh usuario@nombre`.

Cuando ejecutamos el comando para conectarnos, nos imprime un mensaje con el HASH, y nos pregunta que si nos lo creemos, si en vez de una llave pública esta reconocida en nuestras llaves y ve que tenemos el certificado no nos preguntaría. En todos lo equipos se crea el directorio ssh en home, dentro de este tenemos un archivo que se llama known_hosts, el cual contiene las llaves públicas de los ordenadores que ya has confiado, de manera que no nos va a preguntar cuando nos conectemos de nuevo, si queremos que en la primera vez tampoco pregunta la copiamos y pegamos, entre otras opciones. *Nota: Por defecto se crea ese directorio cuando se conetca por ssh.*

Este proceso es muy costos en términos de cómputo, por lo que se usa lo más mínimo posible. Por esto se usa un proceso de handshaking, de esta manera el equipo le manda una propuesta de llave simétrica y de esta manera ya se usa en adelante, ya que el uso de clave privada y pública es muy costoso.

¿Cómo puedo acceder sin contraseña? Para ello nos autenticamos con llave pública y llave privada. En el caso de que decidamos hacerlo asi cuando le mandamos que soy 'nombre', recupera la llave de el directorio .ssh y le manda un mensaje de firmame esto con la clave privada de dicho usuario, y de esta manera se asegura de que efectivamente es el usuario que dice ser. Además, en el directorio home solo puede escribir el usuario y el root.

En la MV:

- ssh-keygen: crea la llave privada dentro de home, luego nos pregunta si queremos protegerla con contraseña, si no queremos, le damos a enter, *pero siempre debe de hacerse.* Luego si nos vamos al directorio vemos dos documentos la llave privada y la pública. Si queremos que no nos pregunte siempre la contraseña, usamos el comando `ssh-agent`, de esta manera el la escribe por ti, hasta que usamos el comando `exit`, en este caso se olvida.

Para usar la pública:

- Accedemos al directorio /home/.ssh

- usamos el comando ssh-copy-id usuario@ip/hostname (copia mi identidad, se puede hacer con ctrl+c/v), y luego nos hace una prueba para que le demostremos que de verdad somos el usuario al que nos queremos conectar, para ello le mandamos la contraseña.

- Luego comprobamos que podemos entrar sin contraseña.

- En la máquina donde nos hemos conectado, vemos un fichero llamado *id_ rsa.pub*, y si lo listamos debe de ser el mismo que la clave pública de la máquina inicial, desde la que nos conectamos a esta.

Nota: El directorio .ssh se crea en el home cuando es el cliente, mientras que en el servidor se crea en el /etc/ssh, también podemos crearlo nosotros a priori.

Ahora podemos ejecuar un comando remoto, por ejemplo: `ssh usuario@ip comando`, de esta manera se ejecuta el comando en la máquina remota. En este punto comienza la automatización de configuración remota. *Ansible son scripts pero ejecutados de manera remota.* Además, podemos copiar archivos usando el comando scp (secure copy), de esta manera se copia el archivo de una máquina a otra. *Nota: scp es un comando de ssh.* El comando es `scp origen destino`, por ejemplo: `scp /home/usuario/archivo usuario@ip:/home/usuario/`, de esta manera se copia el archivo de la máquina local a la remota. Si queremos copiar un directorio, usamos el comando `scp -r origen destino`. Además podemos conectarnos mediante *sftp* (secure file transfer protocol), de esta manera se conecta a la máquina remota y podemos copiar archivos de manera interactiva. *Nota: sftp es un comando de ssh.* Para conectarnos, usamos el comando `sftp usuario@ip`, y luego podemos usar los comandos `put` y `get` para copiar archivos de una máquina a otra.

1.3.3. Ejercicio Opcional

En el ejercicio debemos de dar acceso mediante ssh usando la llave pública y privada, debemos de modificar el puerto para conectarnos en vez del 22 para el ssh usar otro que sea mayor que el 1024, para ello podemos asegurarnos de que no estamos colisionando con un puerto conocido buscando en Innternet.

1. Accedemos al contenido del demonio del ssh que se encuentra en /etc/ssh, una vez dentro nos pone el comando que debemos de ejecuar `semanage...` (aparece en el fichero de ssh de la máquina Rocky), para que se apliquen los cambios debemos de reiniciar el servicio, para ello usamos el comando `sudo systemctl restart sshd` o bien reiniciar la máquina.

2. Además debemos de modificar el firewall para permitir ese puerto, para ello usamos el comando `sudo firewall-cmd --add-port=puerto/tcp --permanent` , y luego reiniciamos el servicio con `sudo systemctl restart firewalld`.

3. Luego debemos de copiar la llave pública a la máquina remota, para ello usamos el comando `ssh-copy-id usuario@ip`, y luego nos pide la contraseña, una vez introducida, ya podemos conectarnos sin contraseña.

4. Para comprobar que todo está correcto, usamos el comando `ssh usuario@ip -p puerto`, de esta manera nos conectamos a la máquina remota.

Con el parámetro -p le especificamos el puerto al que nos queremos conectar, de esta manera nos conectamos al puerto que hemos especificado, ya que de manera predeterminada se conecta al puerto 22.

Parte II

Prácticas

Capítulo 2

Bloque 1

2.1. Ejercicio 1 Opcional

Cuando se hace referencia a la imagen X, se refiere a la imagen 2.X(hasta el punto de Firewall)

El alumno/a debe ser capaz de presentar un MV con la configuración descrita en este apartado. La configuración debe ser permanente, es decir, en todo caso, tras reiniciar el equipo, la configuración será la esperada. Para validar la configuración de red, el alumno/a debe ser capaz de:

- Hacer ping desde el equipo anfitrión a la MV y viceversa.
- Hacer ping desde la MV a cualquier equipo accesible públicamente en Internet por FQHN o IP.
- Conectar por ssh desde el equipo anfitrión a la MV .

2.1.1. Solución

Una vez que hayamos instalado el SO que se nos pide correctamente. Debemos de realizar una serie de ajustes previos:

- Añadir nuestro usuario, para ello debemos de ejecutar lo siguientes comandos (iniciando como usuario root):
  - `sudo useradd nombre_de_usuario`
  - `sudo passwd nombre_de_usuario`
  - `sudo usermod -aG wheel nombre_de_usuario` para que pueda usar el comando sudo.
- Configurar la red NAT y una de tipo Host-Only, para ello en Herramientas en la VM debemos seleccionar la opción de Red y añadir una nueva interfaz de red de tipo Host-Only, y paso seguido configurar la red NAT (ver Figura 1 y 2)

- Comprobar que el servicio SSH está instalado, por defecto se suele instalar, para asegurarnos debemos de ejecutar el comando `sudo systemctl status ssh`. En el caso de que no venga instalado debemos de ejecutar el comando `sudo dnf install -y openssh-server openssh-clients`[1] (Ver Figura 5).
- Cambiar la variable PS1 como se nos pedía, para ello debemos de editar el fichero de bashrc y exportar la variable PS1 con el valor que se nos pedía:
  - `PS1='\u@\h:\t:\w\$ '` (Ver Figura 5), donde:
    - `\u`: Nombre del usuario actual.
    - `\h`: Nombre del hostname (nombre del sistema).
    - `\t`: Hora actual en formato de 24 horas (HH:MM:SS).
    - `\w`: Directorio de trabajo actual.
    - `$`: Símbolo del prompt, que será $ para un usuario normal y # para root.

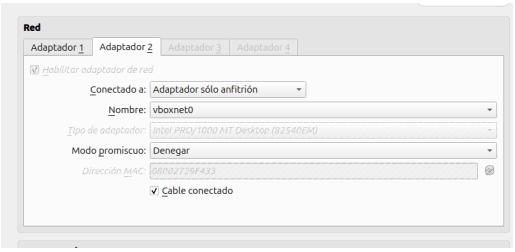


Figura 2.1: Configuración de la red NAT y Host-Only

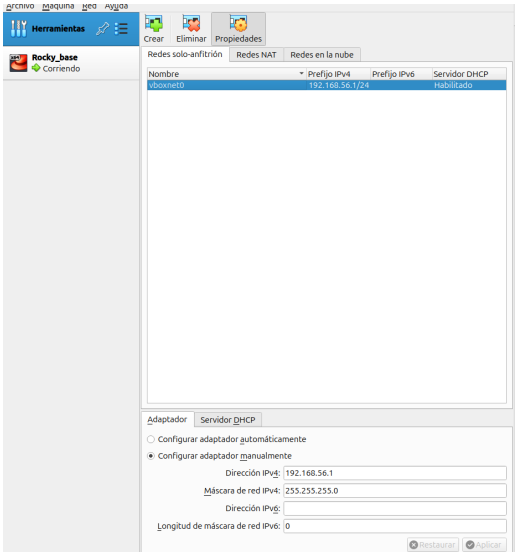


Figura 2.2: Configuración de VirtualBox para la red de Host-Only

Además se nos pide que la Ip de Host Only sea estática, para ello vamos a asegurarnos usando la herramienta *nmtui*, en la que vamos a ver si es estática o no la ip. Como podemos ver en la siguiente imagen esta configurada como ip automática, que viene siendo lo mismo que dinámica por lo que debemos de cambiarlo a manual para poder configurar la ip estática. (Ver Figura 3 y 4). Para ver que efectivamente la ip cambió, podemos verlo en la Figura 5.

Una vez hayamos cambiado la ip estática, debemos de verificar que efectivamente se ha cambiado y para ello usamos el comando `ip a` y vemos que efectivamente se ha cambiado la ip a la que hemos asignado. (Ver Figura 7).

Llegado a este punto vamos a realizar un ping a la máquina anfitriona y viceversa, para ello usamos el comando `ping -c <número de pings>ip_de_la_maquina` y vemos que efectivamente hay conexión entre ambas máquinas. (Ver Figura 8, 9 y

[1] Incluimos clients para añadir el servicio de cliente.

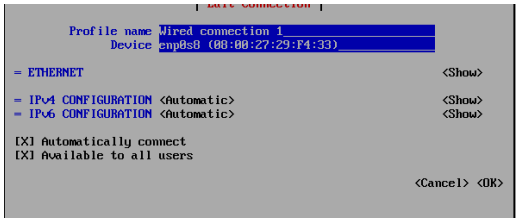


Figura 2.3: Con nmtui vemos que es dinámica

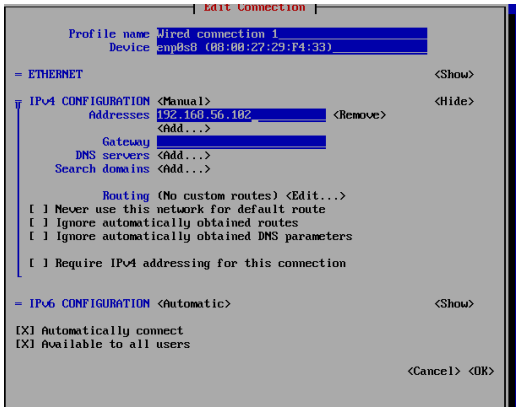


Figura 2.4: Cambiamos a manual y asignamos una ip estática válida

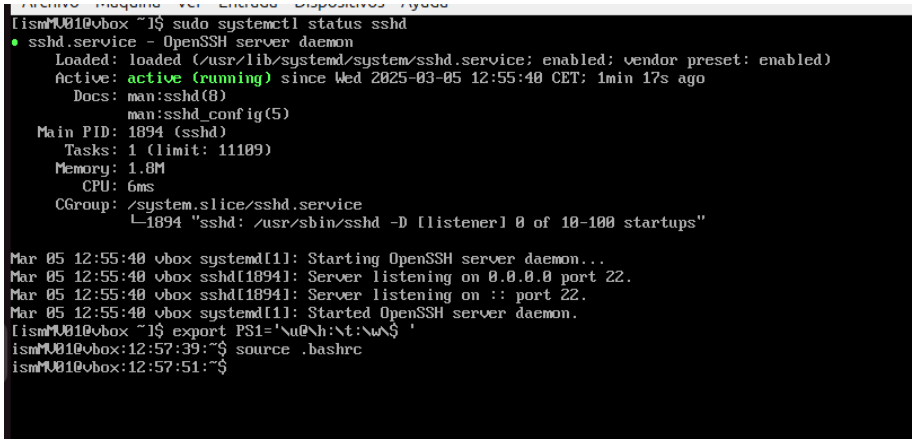


Figura 2.5: Sshd y variable PS1

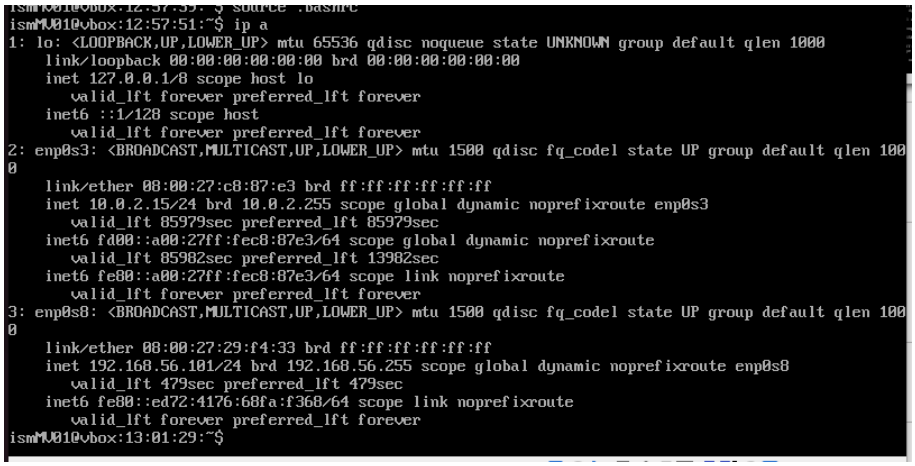


Figura 2.6: Resultado de ip a

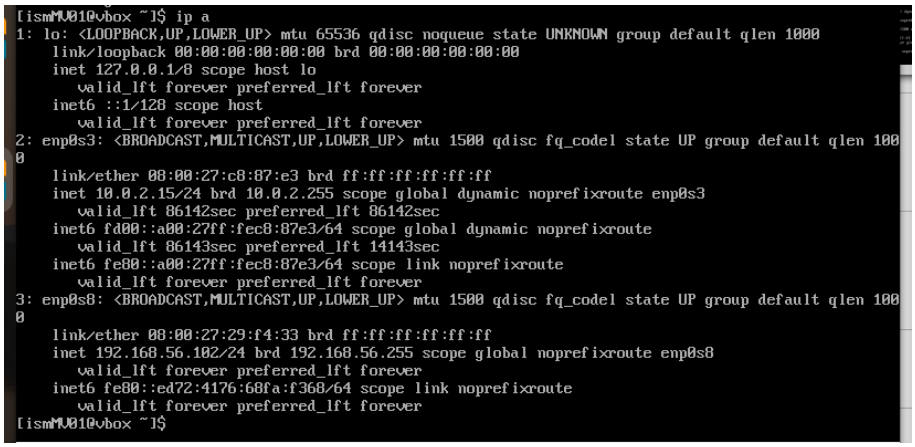


Figura 2.7: Resultado de ip a para verificar el cambio de ip

10). Además, vemos que gracias al *NAT* podemos hacer ping a cualquier máquina accesible en internet[2]. (Ver Figura 11).

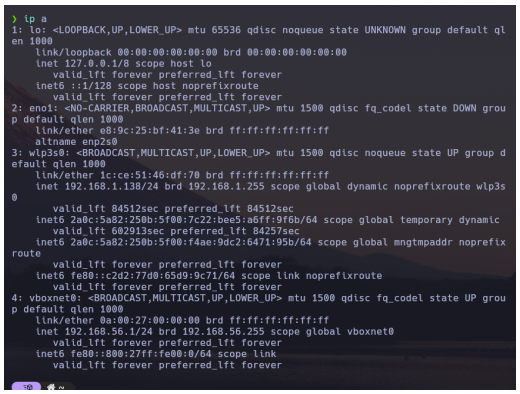


Figura 2.8: Resultado del comando de *ip a* en la máquina anfitriona para ver la ip

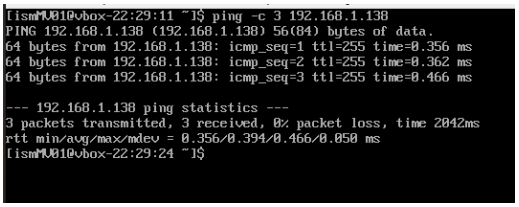


Figura 2.9: Ping a la máquina anfitriona

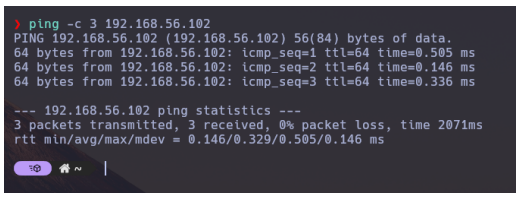


Figura 2.10: Ping de la máquina anfitriona a la máquina virtual

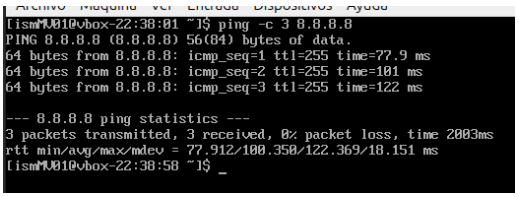


Figura 2.11: Ping a un servidor público (Google)

En cuanto al servicio ssh, debemos de ver el estado del servicio sshd con el comando `sudo systemctl status sshd` y vemos que esta corriendo. En este punto desde el anfitrión podemos introducirt la línea de comando `ssh ismMV01@192.168.56.102` y vemos que efectivamente todo funciona correctamente. (Ver Figura 12 y 13).

[2]Cabe destacar que durante el desarrollo de la actividad, surgían algunas problemas con NetworkManager, polkiy y DBus, pero se solucionaban al reiniciarlos o bien reinstalarlos

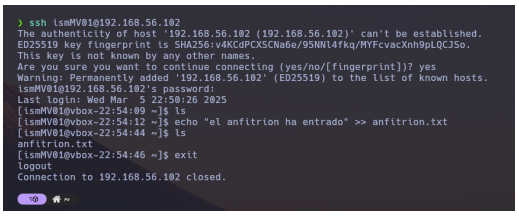


Figura 2.12: Ssh en la máquina anfitriona y creación de un archivo en la MV

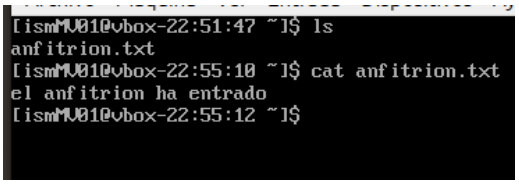


Figura 2.13: Ver el contenido del archivo creado en la MV desde el anfitrión

2.2. Servidor con LVM + RAID

2.2.1. Aspectos clave de LVM

Para gestionar eficazmente el *Logical Volume Manager* (LVM), es fundamental comprender los siguientes componentes y conceptos:

Componentes de la arquitectura de almacenamiento

- **Physical Volume (PV)**: Representa los dispositivos de almacenamiento físico, como discos duros o particiones, que se incorporan al sistema LVM.
- **Volume Group (VG)**: Es una agrupación de uno o más PVs que forman un pool de almacenamiento, del cual se pueden asignar espacios para crear volúmenes lógicos.
- **Logical Volume (LV)**: Son volúmenes virtuales creados dentro de un VG. Los LVs se utilizan como si fueran particiones de disco tradicionales, permitiendo la creación de sistemas de archivos o la asignación directa a aplicaciones.

Gestión de almacenamiento con diferentes características físicas

LVM ofrece flexibilidad para manejar dispositivos de almacenamiento con diversas características físicas:

- **HDD y SSD**: Se pueden combinar en un mismo VG, permitiendo equilibrar rendimiento y capacidad según las necesidades.
- **RAID**: LVM puede trabajar sobre dispositivos RAID, proporcionando una capa adicional de gestión y flexibilidad sobre la configuración RAID existente.

Etiquetado y correspondencia con los archivos de dispositivo

Cada componente en LVM tiene una nomenclatura específica y se asocia a archivos de dispositivo en el sistema:

- **Physical Volumes**: Corresponden a dispositivos físicos, como `/dev/sda1`, `/dev/sdb1`, etc.
- **Volume Groups**: Se nombran según la convención establecida por el administrador, por ejemplo, `vg_datos`.

- **Logical Volumes**: Se nombran dentro de su VG correspondiente, como */dev/vg_datos/lv_backup*, donde *lv_backup* es el nombre del LV.

Comandos de LVM para la gestión de componentes

LVM proporciona una serie de comandos para administrar sus componentes:

- `pvcreate`: Inicializa un dispositivo físico como PV.
- `vgcreate`: Crea un VG a partir de uno o más PVs.
- `lvcreate`: Crea un LV dentro de un VG.
- `pvs`, `vgs`, `lvs`: Muestran información sobre PVs, VGs y LVs respectivamente.
- `pvremove`, `vgremove`, `lvremove`: Eliminan PVs, VGs y LVs respectivamente.

Estos comandos permiten una gestión eficiente y flexible del almacenamiento en sistemas que utilizan LVM.

2.2.2. Niveles de RAID: 0, 1 y 5

Redundant Array of Independent Disks (RAID) es una tecnología que permite combinar múltiples dispositivos de almacenamiento en una unidad lógica para mejorar el rendimiento, la redundancia o ambos. A continuación, se detallan los niveles de RAID 0, 1 y 5, sus ventajas, desventajas y su administración en sistemas Linux utilizando la herramienta de línea de comandos `mdadm`.

RAID 0

RAID 0, conocido como *striping*, distribuye los datos de manera equitativa entre dos o más discos sin información de paridad ni redundancia.

- **Ventajas**:
  - Mayor rendimiento en lectura y escritura debido a la distribución de datos entre los discos.
  - Uso completo de la capacidad de almacenamiento total, ya que no se reserva espacio para paridad o duplicación.
- **Desventajas**:
  - Ausencia de redundancia; la falla de un solo disco resulta en la pérdida total de los datos.

RAID 1

RAID 1, o *mirroring*, duplica los datos en dos o más discos, creando copias idénticas en cada uno.

- **Ventajas**:
  - Alta redundancia; los datos permanecen intactos incluso si uno de los discos falla.
  - Mejora en la velocidad de lectura, ya que los datos pueden leerse desde cualquiera de los discos.
- **Desventajas**:
  - Capacidad de almacenamiento efectiva reducida al 50 % del total, ya que los datos se duplican.

RAID 5

RAID 5 combina rendimiento y redundancia distribuyendo los datos y la paridad entre tres o más discos.

- **Ventajas**:
  - Proporciona tolerancia a fallos; si un disco falla, los datos pueden recuperarse con la información de paridad.
  - Mejor aprovechamiento del almacenamiento comparado con RAID 1, ya que solo se utiliza una fracción del espacio para la paridad.
- **Desventajas**:
  - Rendimiento de escritura inferior al de RAID 0 debido al cálculo de la paridad.
  - En caso de falla de un disco, la reconstrucción puede ser lenta y afectar el rendimiento.

Administración de RAID en Linux con mdadm

La herramienta `mdadm` permite gestionar arreglos RAID en Linux. A continuación, se presentan comandos esenciales:

- Crear un RAID 0 con dos discos:

```
1     mdadm --create --verbose /dev/md0 --level=0 --raid-devices
    =2 /dev/sdX /dev/sdY
2
```

- Crear un RAID 1 con dos discos:

```
1     mdadm --create --verbose /dev/md0 --level=1 --raid-devices
    =2 /dev/sdX /dev/sdY
2
```

- Crear un RAID 5 con tres discos:

```
mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3 /dev/sdX /dev/sdY /dev/sdZ
```

- Verificar el estado del RAID:

```
cat /proc/mdstat
```

- Detener un RAID:

```
mdadm --stop /dev/md0
```

2.2.3. Aspectos clave para la administración de servidores Linux

Para implementar eficazmente soluciones en servidores Linux, es esencial comprender y manejar los siguientes aspectos:

Modos de ejecución y modo de mantenimiento en un servidor Linux

Los sistemas Linux operan en diferentes niveles de ejecución o *runlevels*, que determinan los servicios y procesos que se ejecutan. Con la adopción de *systemd*, estos niveles se denominan *targets*. El modo de mantenimiento, conocido como *rescue.target* o *emergency.target*, es crucial para tareas de recuperación y administración del sistema. Para cambiar al modo de mantenimiento, se puede utilizar el siguiente comando:

```
sudo systemctl isolate rescue.target
```

Para volver al modo multiusuario estándar:

```
sudo systemctl isolate multi-user.target
```

Estructura estándar del sistema de archivos en Linux

La estructura de directorios en Linux sigue el estándar de jerarquía de sistemas de archivos (*Filesystem Hierarchy Standard - FHS*). Algunos directorios principales incluyen:

- /: Directorio raíz que contiene todos los demás directorios.
- /**bin**: Ejecutables esenciales para todos los usuarios.
- /**etc**: Archivos de configuración del sistema.
- /**home**: Directorios personales de los usuarios.
- /**var**: Datos variables como registros y colas de impresión.

Sistemas de archivos comunes en Linux

Linux soporta diversos sistemas de archivos. Algunos de los más comunes son:

- **ext4**: Sistema de archivos por defecto en muchas distribuciones, conocido por su estabilidad y rendimiento.
- **XFS**: Adecuado para manejar grandes volúmenes de datos y archivos de gran tamaño.
- **Btrfs**: Ofrece características avanzadas como instantáneas (*snapshots*) y compresión.

Montaje y desmontaje de volúmenes

El montaje de sistemas de archivos permite acceder a dispositivos de almacenamiento. Para montar un dispositivo:

```
sudo mount /dev/sdX1 /mnt/punto_de_montaje
```

Para desmontarlo:

```
sudo umount /mnt/punto_de_montaje
```

Las configuraciones de montaje persistentes se definen en el archivo `/etc/fstab`.

Comandos básicos para la gestión de archivos

La administración de archivos en Linux se realiza mediante comandos de línea.

Estos se deben de haber estudiado en asignaturas anteriores como Sistemas Operativos.

2.2.4. Ejercicio Opcional

Partiendo de un servidor básico configurado de acuerdo al apartado 2, el alumno/a deberá afrontar el caso práctico descrito a continuación:

Se desea instalar un servicio de gestión documental en el servidor. Se espera que este servicio precise de una cantidad espacio de almacenamiento creciente con el tiempo, pudiendo llegar a ser considerable.

Por otro lado, el contenido será crítico, por lo que se desea proporcionar algún mecanismo de respaldo ante fallos en el dispositivo de almacenamiento.

El alumno/a debe diseñar los cambios en el sistema de almacenamiento e implementarlo empleando prácticas adecuadas de administración que garanticen la conservación de la información en el sistema y procuren la máxima disponibilidad del servicio.

Solución

Para la resolución de este ejercicio vamos a seguir lo realizado en clase. (Ver apuntes de clase correspondientes a la sección de LVM+RAID).

En primer lugar creamos los discos, para ello entramos en la configuración de la máquina virtual y añadimos dos discos duros de 1GB cada uno. Una vez añadidos los discos duros, iniciamos la máquina virtual y ejecutamos el comando `lsblk` para ver los discos duros que tenemos disponibles. (Ver Figura 14).

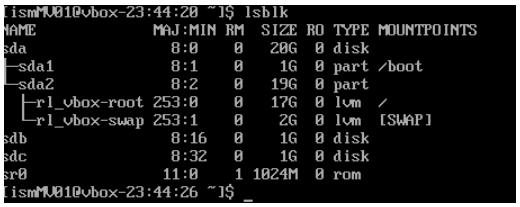


Figura 2.14: Resultado del comando lsblk

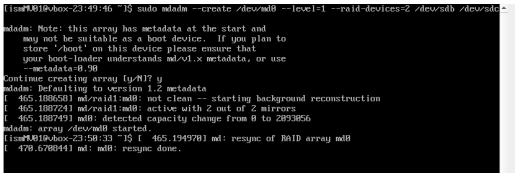


Figura 2.15: Comando mdadm

A continuación, instalamos *mdadm* con el comando `sudo dnf install -y mdadm` y creamos un RAID 1 con los dos discos duros que hemos añadido. Para ello ejecutamos el comando `sudo mdadm -create -verbose /dev/md0 -level=1 -raid-devices=2 /dev/sdb /dev/sdc` y vemos que se ha creado correctamente. (Ver Figura 15).

Acto seguido debemos de crear un PV, VG y LV. Para ello ejecutamos los siguientes comandos (Ver Figura 16):

- `sudo pvcreate /dev/md0`
- `sudo vgcreate vg_datos /dev/md0`
- `sudo lvcreate -L 900M -n lv_datos vg_datos`

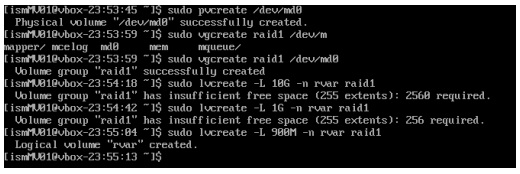


Figura 2.16: Resultado de crear los volúmenes

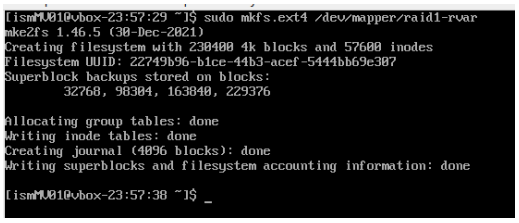


Figura 2.17: Comando mkfs

Ahora debemos de formatear el volumen lógico en formato ext4. (Ver Figura 17).

A continuación, montamos (Ver Figura 18) y trasladamos la carpeta *var* al nuevo volumen lógico. Para ello ejecutamos los comando que figuran en los apuntes de clase.

Figura 2.18: Resultado del comando mount

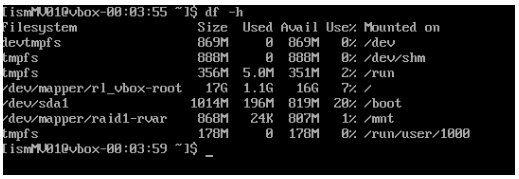


Figura 2.19: Comando df -h

Ejecutamos el comando `df -h` para ver que efectivamente se ha montado correctamente el volumen lógico. (Ver Figura 19).

Ejecutamos el isolate del systemctl para entrar en modo de mantenimiento y poder trasladar la carpeta *var* al nuevo volumen lógico. (Ver Figura 20).

Ahora debemos de trabajar con la serie de comandos que se especifican para poder mover la carpeta y que se haga de forma correcta. (Ver Figura 21).

Una vez trasladad la carpeta, debemos de hacer permanentes los cambios, para ello debemos de editar el fichero `/etc/fstab` y añadir la siguiente línea (Ver Figura 22):

Figura 2.20: Resultado del comando systemctl isolate

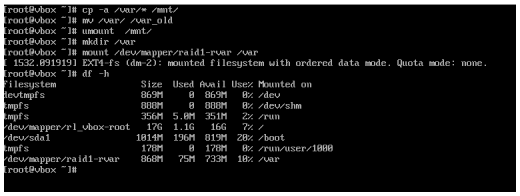


Figura 2.21: Serie de operaciones para trasladar la carpeta var

```
/dev/mapper/raid1-rvar /var ext4 defaults 0 0
```

Recargamos la configuraciones con el deamon de systemd y vemos que aparece la parte que estabamos montando (Ver Figura 23).

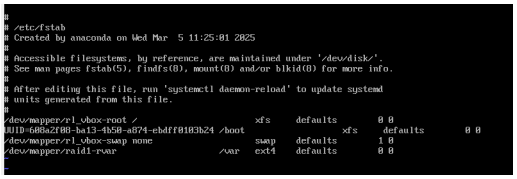


Figura 2.22: Resultado de editar /etc/fstab como se indica

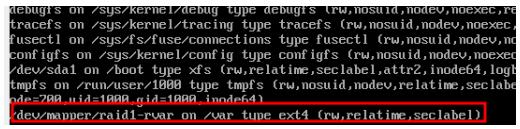


Figura 2.23: Resultado del comando systemctl daemon-reload y mount

Y como podemos comprobar en la Figura 24, hemos conseguido trasladar la carpeta *var* al nuevo volumen lógico y todo funciona correctamente.

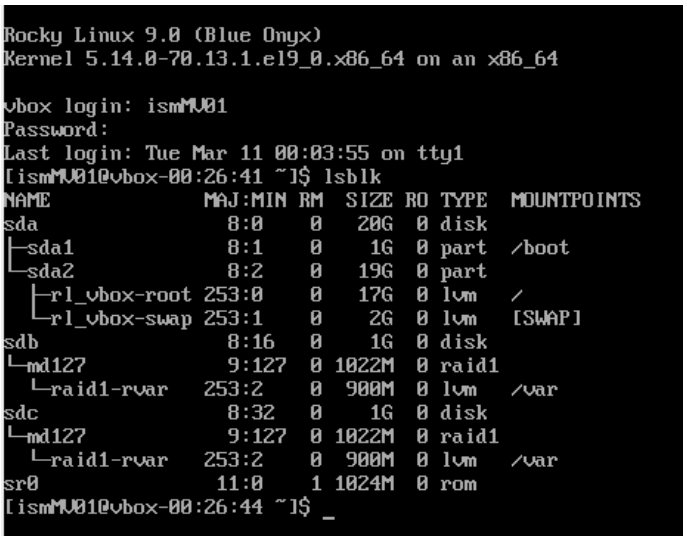


Figura 2.24: Resultado final

2.3. Acceso seguro al servidor: Firewall + SSHD

2.3.1. Iptables

`iptables` es una herramienta de línea de comandos utilizada para configurar el cortafuegos en el núcleo de Linux, implementado dentro del proyecto Netfilter. Aunque `iptables` es un marco heredado, `nftables` se presenta como su reemplazo moderno, ofreciendo una capa de compatibilidad.

Instalación

El núcleo estándar de Arch Linux viene compilado con soporte para `iptables`. Para instalar las utilidades de usuario, se debe instalar el paquete `iptables`, que es una dependencia indirecta del metapaquete `base`, por lo que debería estar instalado por defecto en el sistema.

Conceptos Básicos

- **Tablas**: Colecciones de reglas con propósitos específicos.
- **Cadenas**: Listas de reglas dentro de una tabla que se recorren en orden.
- **Reglas**: Definiciones que consisten en un criterio de coincidencia y una acción a ejecutar si se cumple dicho criterio.

Configuración y Uso

`iptables` se puede configurar directamente desde la línea de comandos o mediante diversas interfaces frontales, tanto de consola como gráficas. Para mostrar las reglas actuales, se utiliza:

```
iptables -L
```

Para restablecer las reglas:

```
iptables -F
```

Registro de Actividad

`iptables` permite registrar paquetes para monitorear actividad o depurar reglas. Es posible limitar la tasa de registro para evitar la saturación de los registros del sistema.

Alternativas y Herramientas Relacionadas

- `nftables`: Proyecto que busca reemplazar a `iptables`, proporcionando un nuevo marco de filtrado de paquetes y una utilidad de espacio de usuario llamada `nft`.
- `ipset`: Aplicación complementaria que permite crear conjuntos de direcciones IP para ser utilizados en reglas de `iptables`, facilitando el bloqueo o la aceptación de múltiples direcciones de manera eficiente.

- `ufw` **(Uncomplicated Firewall)**: Programa para gestionar el cortafuegos de manera sencilla, proporcionando una interfaz de línea de comandos fácil de usar.

Recursos Adicionales

Para configuraciones más avanzadas, como la creación de un cortafuegos con estado o compartir la conexión a Internet, se pueden consultar las siguientes guías:

- **Cortafuegos con Estado Sencillo**: Explica cómo configurar un cortafuegos con estado utilizando `iptables`, detallando las reglas necesarias y su propósito.
- **Compartir Conexión a Internet**: Describe cómo compartir la conexión a Internet desde una máquina a otras, incluyendo los requisitos y pasos necesarios.

2.3.2. firewalld y firewall-cmd en Rocky Linux

`firewalld` es un servicio de cortafuegos dinámico que permite gestionar reglas de firewall sin necesidad de reiniciar el servicio. Su herramienta de línea de comandos, `firewall-cmd`, proporciona una interfaz para la administración del firewall en sistemas como Rocky Linux.

Gestión del Firewall con firewall-cmd

`firewall-cmd` permite configurar reglas de firewall de manera flexible y en tiempo real. Algunas de las operaciones más comunes incluyen:

- **Verificar el estado del firewall**:

```
firewall-cmd --state
```

- **Listar las reglas activas**:

```
firewall-cmd --list-all
```

- **Abrir un puerto específico** (ejemplo: puerto 80 TCP):

```
firewall-cmd --add-port=80/tcp --permanent
```

- **Cerrar un puerto específico**:

```
firewall-cmd --remove-port=80/tcp --permanent
```

- **Recargar la configuración del firewall** para aplicar cambios:

```
firewall-cmd --reload
```

Administración del Servicio firewalld con systemctl

`systemctl` es una herramienta utilizada para gestionar servicios en sistemas basados en `systemd`, como Rocky Linux. Se puede utilizar para controlar el servicio `firewalld` de la siguiente manera:

- **Verificar si el servicio está activo**:

```
systemctl status firewalld
```

- **Iniciar el servicio** si está detenido:

```
systemctl start firewalld
```

- **Detener el servicio** si se necesita desactivarlo temporalmente:

```
systemctl stop firewalld
```

- **Habilitar el servicio para que se inicie automáticamente** al arrancar el sistema:

```
systemctl enable firewalld
```

- **Deshabilitar el servicio para evitar su inicio automático**:

```
systemctl disable firewalld
```

Verificación de Configuración con nmap

`nmap` es una herramienta de escaneo de redes que permite verificar los puertos abiertos y configuraciones de firewall. Se puede usar para comprobar la efectividad de las reglas de firewall aplicadas. Algunos ejemplos de uso incluyen:

- **Escanear los puertos abiertos de una dirección IP específica**:

```
nmap <direccion_ip>
```

- **Escanear puertos específicos** (ejemplo: puertos 22 y 80):

```
nmap -p 22,80 <direccion_ip>
```

- **Detectar el sistema operativo del host analizado**:

```
nmap -O <direccion_ip>
```

Estas herramientas permiten una administración avanzada del firewall en Rocky Linux, garantizando la seguridad de la red y el control del tráfico de red en el sistema.

2.3.3. Ejercicio Opcional

Como caso práctico, partiendo de una MV con la configuración base descrita en el apartado 2, el alumno/a deberá ser capaz de instalar un servidor de HTTP, Apache o Nginx, y habilitar/deshabilitar su acceso por Firewall.

Para ello, instalará el servidor web de su elección y modificará la home page para mostrar un mensaje:'Bienvenidos a la web de <Nombre y Apellidos del alumno/a>en Prácticas ISE'.

El servicio web debe estar accesible en la servidor (MV) en el puerto por defecto (80) usando un navegador web convencional corriendo en el anfitrión (Host).

Un escaneo de puertos sobre el servidor solo debe mostrar como accesibles los puerto web y ssh.

Solución

Para la resolución de este ejercicio vamos a seguir los siguientes pasos:

1. Instalamos el servidor web Apache con el comando `sudo dnf install -y httpd`. También tenemos la opción de instalar Nginx con el comando `sudo dnf install nginx- y`. *Tenemos libre elección de servidor web, en mi caso será Apache.*

2. Debemos de activar este servicio con el comando:

   - Apache: `sudo systemctl enable httpd --now`
   - Nginx: `sudo systemctl enable nginx --now`
   - Debemos de verificar que el servicio esta activo con los comandos:
     - `sudo systemctl status httpd # Para Apache`
     - `sudo systemctl status nginx # Para Nginx`

3. Modificación de la página principal.
   Debemos de personalizar la página de inicio de nuestra web para que muestre el mensaje que se nos pide en el enunciado.

   - Apache: `echo "Bienvenidos a la web de <Nombre y Apellidos> en Prácticas ISE" | sudo tee /var/www/html/index.html`
   - Nginx: `echo "Bienvenidos a la web de <Nombre y Apellidos> en Prácticas ISE" | sudo tee /usr/share/nginx/html/index.html`

4. Configuración del Firewall.
   Para permitir el acceso a la web, es necesario abrir el puerto 80 en el firewall. Para ello, ejecutamos los siguientes comandos:

   - `sudo firewall-cmd --add-service=http --permanent`
   - `sudo firewall-cmd --reload`
   - Para verificar que el puerto esta abierto: `sudo firewall-cmd --list-all`

5. Acceder desde el Navegador.
   Si todo está configurado correctamente, se visualizará el mensaje personalizado[3](Ver Figura 2.25).

Figura 2.25: Acceso a la web desde el navegador

6. Restricción de puertos con Firewall.
   Para asegurar que solo los puertos web (80) y SSH (22) estén accesibles, se deben eliminar otras reglas de firewall:

   - `sudo firewall-cmd --list-services --permanent`[4]
   - `sudo firewall-cmd --remove-service=<nombre> --permanent`
   - `sudo firewall-cmd --reload`
   - Extra.
     Como extra podemos ver los puertos que están abiertos con el comando `sudo firewall-cmd --list-ports --permanent` y cerramos el puerto con `sudo firewall-cmd --remove-port=xxxx/tcp --permanent`, donde xxxx es el puerto que queremos cerrar. Otra forma de solo tener dos puertos abiertos es con los comandos:
     - `sudo firewall-cmd --add-service=http --permanent`
     - `sudosudo firewall-cmd --add-service=ssh --permanent`
     - Y de nuevo recargamos y listamos todos los puertos.

[3]Para saber a que ip acceder, podemos usar el comando *ip a* y coger la correspondiente a *enp0s8*.
[4]Con este comando podemos ver los servicios que están activos en el firewall, desactivamos todos, a excepción de http y ssh.

2.4. SSH

2.4.1. Administración remota con SSH y criptografía asimétrica

La administración remota es una tarea fundamental en la gestión de servidores. Una de las herramientas más utilizadas para este propósito es `SSH` (Secure Shell), que permite acceder y administrar un sistema de manera segura a través de una red. Es importante destacar que `SSH` puede referirse tanto a un cliente como a un servicio.

Instalación y configuración de SSH

Para instalar y habilitar el servicio SSH en un sistema basado en Rocky Linux, se utilizan los siguientes comandos:

```
sudo dnf install -y openssh-server
sudo systemctl enable --now sshd
```

El servicio `sshd` es el demonio que permite las conexiones SSH entrantes. Su configuración se encuentra en el archivo:

```
/etc/ssh/sshd_config
```

Al editar este archivo, es posible modificar diversas opciones de seguridad, como:

- **Deshabilitar el acceso como root por contraseña:** Para mejorar la seguridad, se recomienda restringir el acceso directo de root:

  ```
      PermitRootLogin no
  ```

- **Cambio de puerto predeterminado:** SSH por defecto usa el puerto 22, pero puede cambiarse para reducir ataques automatizados:

  ```
      Port 2222
  ```

Después de modificar la configuración, es necesario reiniciar el servicio:

```
sudo systemctl restart sshd
```

Configuración del firewall para SSH

Si se cambia el puerto de `SSH`, es necesario actualizar la configuración del firewall:

```
sudo firewall-cmd --add-port=2222/tcp --permanent
sudo firewall-cmd --reload
```

Criptografía simétrica y asimétrica en SSH

Criptografía simétrica es un método en el que la misma clave se usa tanto para cifrar como para descifrar los datos. Es rápida pero menos segura para la comunicación remota, ya que ambas partes deben compartir la clave de forma segura.

Criptografía asimétrica, utilizada en SSH, emplea un par de claves: una clave pública y una clave privada. El cliente genera un par de claves y comparte la clave

pública con el servidor, lo que permite autenticarse sin necesidad de enviar una contraseña.

Para generar un par de claves en SSH, se usa:

```
ssh-keygen -t rsa -b 4096
```

Y para copiar la clave pública al servidor:

```
ssh-copy-id usuario@servidor
```

Esto permite autenticarse sin necesidad de contraseña, mejorando la seguridad y facilitando la automatización de tareas en servidores remotos.

2.4.2. Ejercicio Opcional

Partiendo de un servidor base configurado siguiendo las indicaciones del apartado 2, el alumno/a modificará servicio SSHD para que, en lugar del puerto por defecto (22), se ejecute en un puerto alternativo de un valor mayor a 1024. Se recomienda que consulte la lista de puertos reconocidos por el sistema en /etc/ports para evitar emplear un puerto que ya tenga una aplicación predefinida.

Se concederá acceso remoto por llave pública a un usuario de su elección. El ejercicio se validará ejecutando un comando de forma remota sobre el servidor SSH con la nueva configuración. El comando presentará el contenido completo (incluido ficheros y directorios ocultos) con del directorio home del usuario remoto empleado en la conexión. Para ello, desde el ordenador anfitrión (o una MV distinta a la que se va a acceder) se empleará ssh sin terminal remoto y sin contraseña, pasando como único como parámetro el comando a ejecutar.

Solución

Modificación del servicio SSHD y acceso remoto por llave pública

En este ejercicio, se modificará la configuración del servicio SSH para que utilice un puerto alternativo y se habilitará el acceso remoto mediante autenticación con clave pública.

Paso 1: Verificar puertos disponibles

Antes de modificar la configuración de SSH, es recomendable consultar la lista de puertos reconocidos por el sistema para evitar conflictos con otros servicios. Para ello, se puede inspeccionar el archivo:

```
cat /etc/services | less
```

Se debe elegir un puerto mayor a 1024 que no esté en uso.

Paso 2: Modificar la configuración de SSH

Editar el archivo de configuración de SSH con un editor de texto como ‘vim‘ o ‘nano‘:

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

Buscar la línea que especifica el puerto (‘#Port 22‘), descomentarla y cambiarla por el número de puerto elegido, por ejemplo:

```
Port 2222
```

Guardar los cambios y salir del editor.

Solución al error al cambiar el puerto de SSH

Si al cambiar el puerto en la configuración de SSH y reiniciar el servicio se obtiene un error, se deben seguir los siguientes pasos para solucionar el problema:

1. **Verificar errores en la configuración de SSH** Ejecutar el siguiente comando para comprobar si hay errores en el archivo de configuración:

   ```
   sudo sshd -t
   ```

   Si se detectan errores de sintaxis o configuración en el archivo */etc/ssh/sshd_ config*, se deben corregir antes de continuar.

2. **Confirmar que el puerto está permitido en SELinux (si aplica)** Si el sistema usa SELinux, es necesario permitir el nuevo puerto:

   ```
   sudo semanage port -a -t ssh_port_t -p tcp 2222
   ```

   Si el comando `semanage` no está disponible, se puede instalar con:

   ```
   sudo dnf install policycoreutils-python-utils
   ```

   Luego, verificar que el puerto se agregó correctamente con:

   ```
   sudo semanage port -l | grep ssh
   ```

3. **Permitir el nuevo puerto en `firewalld`** Si el sistema usa `firewalld`, se debe agregar el puerto y recargar la configuración:

   ```
   sudo firewall-cmd --permanent --add-port=2222/tcp
   sudo firewall-cmd --reload
   ```

4. **Reiniciar el servicio SSH y comprobar su estado** Una vez realizados los cambios, reiniciar el servicio SSH:

   ```
   sudo systemctl restart sshd
   ```

   Si el servicio sigue fallando, revisar los logs para obtener más detalles sobre el error:

   ```
   sudo journalctl -xeu sshd
   ```

Siguiendo estos pasos, se podrá cambiar el puerto de SSH sin problemas y reiniciar el servicio correctamente.

Paso 3: Ajustar firewalld para permitir conexiones en el nuevo puerto

Si ‘firewalld‘ está en uso, es necesario agregar el nuevo puerto al firewall y eliminar el acceso al puerto 22 si no se desea que permanezca abierto:

```
sudo firewall-cmd --permanent --add-port=2222/tcp
sudo firewall-cmd --permanent --remove-service=ssh
sudo firewall-cmd --reload
```

Paso 4: Reiniciar el servicio SSH

Aplicar los cambios reiniciando el servicio SSH:

```
sudo systemctl restart sshd
```

Paso 5: Configurar autenticación por clave pública

Desde el cliente, generar un par de claves SSH si no se tienen:

```
ssh-keygen -t rsa -b 4096
```

Copiar la clave pública al servidor(Ver Figura 2.26):

```
ssh-copy-id -p 2222 usuario@IP_del_Servidor
```

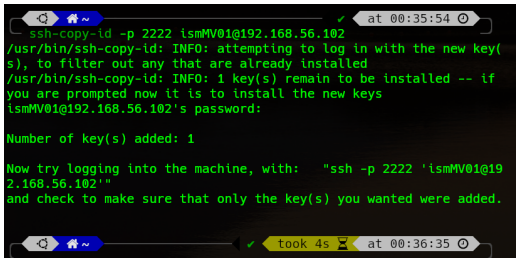


Figura 2.26: SSH en mi máquina

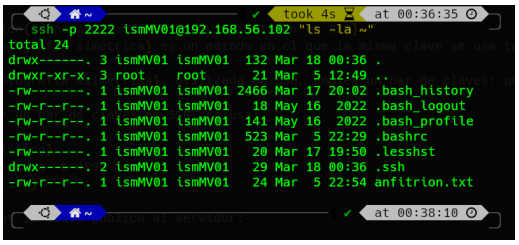


Figura 2.27: SSH final

Paso 6: Validar la conexión sin contraseña

Para comprobar que el acceso es correcto, se ejecutará un comando remoto en la máquina virtual desde otro equipo o una máquina anfitriona. Se listará el contenido completo del directorio home del usuario remoto, incluidos los archivos ocultos (Ver Figura 2.27):

```
ssh -p 2222 usuario@IP_del_Servidor "ls -la ~"
```

Si todo está configurado correctamente, el comando mostrará los archivos del directorio home sin requerir contraseña.

Bibliografía

[1] Ismael Sallami Moreno, **Estudiante del Doble Grado en Ingeniería Informática + ADE**, Universidad de Granada, 2025.