Universidad de Valladolid

Cuaderno de bitácora

Ingeniería Informática
Tecnologías de la Información

Daniel Sanabria Salamanqués

13 de octubre de 2025

Índice

1. Instalación

1.1 Clúster de Máquinas Virtuales

Para comenzar con la instalación, me dirijo a la página matrix.inf.uva.es e inicio sesión con mi cuenta de laboratorio de la escuela. Una vez hecho, observo que en el `Datacenter` se encuentra mi máquina virtual `vm3803.virtual.lab.inf.uva.es`. Al hacer doble clic, compruebo en la sección de `Hardware` si está en el apartado `CD/DVD` la imagen de `Ubuntu Server`. Como no aparece, hago clic sobre ese apartado y con la opción `Edit` que aparece en la parte superior, agrego la imagen a ese disco de la máquina.

1.2 Configuración de instalación

Tras esto, voy a la sección `Console` para iniciar la máquina virtual y comenzar con la instalación de `Ubuntu Server`. Lo primero es seleccionar el idioma para el sistema; en mi caso, escojo inglés. Después, indico que no quiero realizar la actualización para obtener `Ubuntu Server 25.04`. Luego, para la configuración del teclado, selecciono el teclado español, debido a que mi teclado necesita esa configuración. En la siguiente pantalla, escojo que la instalación base será `Ubuntu Server` por defecto y sin opciones adicionales. En la configuración de red, no modifico ningún valor ni agrego ningún `proxy`. En cuanto al almacenamiento, indico que para la instalación use todo el disco y que no lo monte como un grupo `LVM`. Después de confirmar la configuración del almacenamiento, relleno en la siguiente pantalla los datos de mi perfil:

- **Nombre**: Daniel
- **Nombre de servidor**: vm3803
- **Username**: dansana

Para la configuración de la conexión SSH, selecciono la opción de que se instale `OpenSSH`. Para terminar, no agrego ninguna `snap` al sistema y después de seleccionar `Done`, dejo que se termine la instalación con la configuración seleccionada. Tras unos minutos, la instalación termina y reinicio el sistema.

Una vez que ha arrancado, inicio sesión con el usuario y la contraseña que he creado y, acto seguido, procedo a purgar ciertas aplicaciones que no son necesarias.

1.3 Reconocimiento del entorno

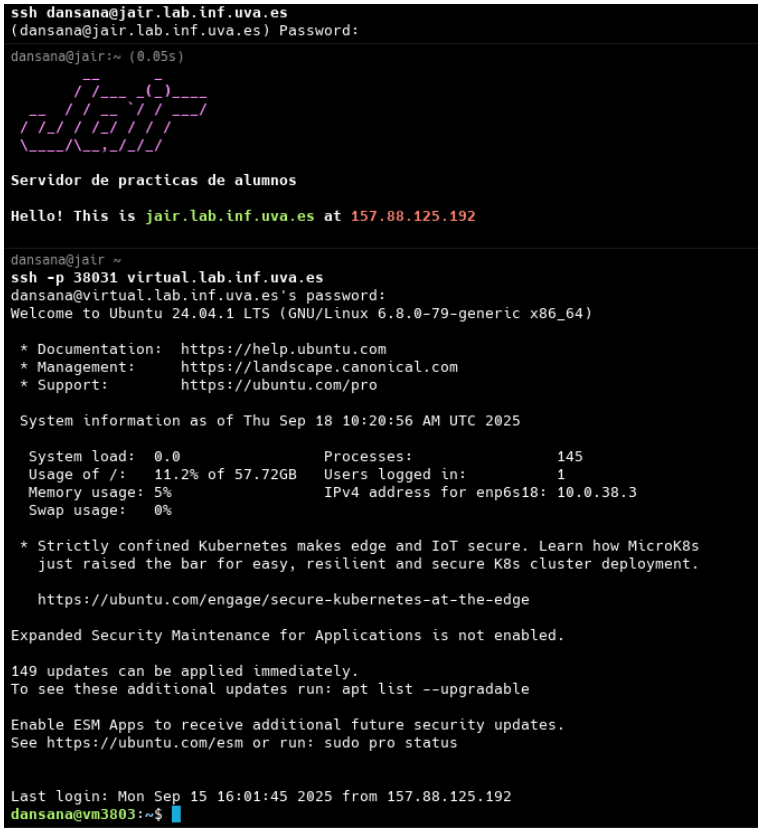
Nos piden realizar un reconocimiento del entorno para conocer acerca del sistema que hemos instalado, además de saber cómo funciona la máquina virtual en la página matrix.inf.uva.es:

- **Version Kernel Linux**: El comando `cat /proc/version`, nos devuelve la información acerca del Linux instalado. En este caso, se trata de un Linux con la versión el kernel 6.8.0-79-generic. El funcionamiento del comando es mostrar lo que contiene el archivo `version` dentro de `proc`, que se trata del sistema de ficheros. Otra opción, es con el comando `uname` que muestra información del sistema dependiendo de la opción que se le pase como argumento.
- **Particiones**: Con el comando `df -h`, se obtiene las particiones montadas. En este caso, tenemos las siguientes particiones:
  - */dev/sda1*: Montada en el directorio `/boot/efi` y es la encargada de el arranque del sistema.
  - */dev/sda2*: Montada en el directorio raíz `/`, dedicada al resto de sistema.
- **Espacio libre**: Con el mismo comando que el punto anterior, se puede ver que hay varias columnas dedicadas al almacenamiento de cada partición:
  - */dev/sda1*: Con `1.1G` en total, solo se ha usado el 1 %, es decir, `6.2M` se ha utilizado y se encuentran disponibles `1.1G` para usar.
  - */dev/sda2*: Con `58G` en total, solo se ha usado el 12 %, es decir, `6.5G` se ha utilizado y se encuentran disponibles `49G` para usar.

- **Cerrar sesión**: Cuando se ha iniciado sesión y queremos cerrar esa misma sesión, simplemente tenemos que escribir el comando `logout` y el sistema cerrará la sesión.
- **Apagar la máquina**: Desde la consola del sistema, mediante el comando `shutdown -h` se le enviará una señal al sistema para apagar la máquina, deteniendo todos los procesos y sincronizando los discos. Si queremos hacerlo inmediato, añadimos `now` al lado de `shutdown`.
- **Reiniciar la máquina**: Para el reinicio inmediato, se emplea el comando `reboot`, o, para un reinicio programado, se emplea `shutdown -r`.
- **Controles de la consola de la máquina virtual**: Se pide usar los controles que aparecen en la parte superior:
  1. Cuando la máquina esté encendida, nos indican apagar la máquina con `Stop`. Esto obligará a la máquina a hacer un apagado forzado.
  2. Después de volver a encender, nos piden restear la máquina mediante la opción `Reset`. Funciona igual que escribir el comando `reboot`.
  3. Por último, será apagar de nuevo la máquina pero con la opción `Shutdown` que será lo mismo que escribir el comando `shutdown -h`.

1.4 Acceso remoto vía ssh

Se nos indica que el sistema ya tiene instalado y activado el servicio de conexión segura `sshd` (que previamente hemos configurado en la configuración de la instalación) y para comprobar que funciona correctamente, me conectaré desde `Jair` a esta máquina, usando la red de la UVa. Aquí se muestra una captura del proceso:

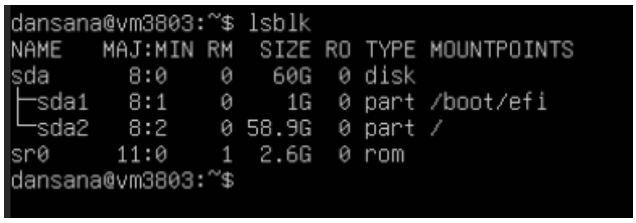


1.5 Activar cuenta root

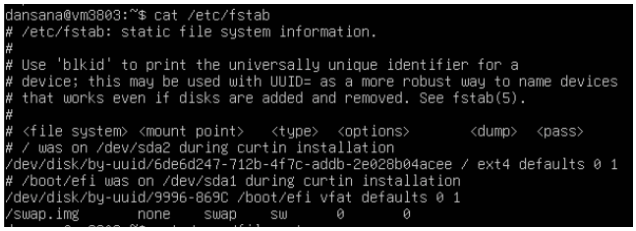
Lo siguiente que se indica es activar la cuenta `root` cambiando su contraseña mediante `sudo passwd root` e indicando una clave para ese usuario y así poder acceder a la consola directamente como `root`, ya que por defecto no trae ninguna contraseña y puede ser una brecha de seguridad.

1.6 Administración del disco

Se pide obtener información sobre las particiones lógicas y física de nuestra máquina virtual, con ayuda de los comandos que se explican en las transparencias. Y para saber el sistema de ficheros que se está utilizando, tendremos que hacer un `cat` al fichero `/etc/fstab`, que contiene las informaciones que conciernen al montaje de las particiones que hay en el sistema.



- **Dispositivos**: Tal y como se muestra en la imagen, solo tenemos un dispositivo de almacenamiento `sda` con una capacidad de 60G. El otro dispositivo que existe es el CD de instalación de `Ubuntu Server` que ocupa 2.6G.



- **Particiones**: Existen dos particiones en el disco `sda`:
  - *sda1*: Con un tamaño de 1G y montado en el directorio `/boot/efi`, es la encargada de almacenar las herramientas de arranque del sistema que serán lanzadas por el firmware UEFI. Emplea el sistema de ficheros `vfat`.
  - *sda2*: Partición principal, anclado en el directorio `/`, con el tamaño restante del disco para almacenar todas las aplicaciones y ficheros del sistema operativo y del usuario. Emplea el sistema de ficheros `ext4`.

Después, se nos exige investigar el fichero `/proc/filesystems` donde se ubican los sistemas de ficheros que es capaz de entender el sistema.

```
dansana@vm3803:~$ cat /proc/filesystems
nodev   sysfs
nodev   tmpfs
nodev   bdev
nodev   proc
nodev   cgroup
nodev   cgroup2
nodev   cpuset
nodev   devtmpfs
nodev   configfs
nodev   debugfs
nodev   tracefs
nodev   securityfs
nodev   sockfs
nodev   bpf
nodev   pipefs
nodev   ramfs
nodev   hugetlbfs
nodev   devpts
        ext3
        ext2
        ext4
        squashfs
        vfat
nodev   ecryptfs
        fuseblk
nodev   fuse
nodev   fusectl
nodev   efivarfs
nodev   mqueue
nodev   pstore
        btrfs
nodev   autofs
nodev   binfmt_misc
```

- Se muestra dos columnas donde en la izquierda se indica si se requiere un dispositivo de bloque asociado al sistema de fichero que se muestra en la columna de la derecha.
- Por ejemplo, para los sistemas de ficheros `ext2, ext3 o ext4` no se indica el valor `nodev`, por lo que es necesario usar un dispositivo físico para usar ese sistema de fichero. Pero, para `tmpfs o proc`, no es necesario tener un dispositivo físico.

1.6.1 Loop

Después, creamos un sistema de archivos o fichero dentro de un fichero nuevo:

1. Creamos el fichero mediante el comando `dd`, donde se le indica los siguientes parámetros:
   `dd if=/dev/zero of=fichero bs=1 count=4096`
   - **if**: Desde que fichero o directorio se van a leer los datos. Como vamos a crear un fichero vacío, haremos uso de `/dev/zero` que se trata de un fichero especial desde el que se obtiene un flujo de cero, con el propósito de inicializar un fichero.
   - **of**: Indicamos la ruta con el nombre del fichero creado.
   - **bs**: Indicamos el tamaño del bloque que se quiere leer y escribir. Para este caso, se escoge de 1 MB por comodidad.
   - **count**: El número de bloques que se van a crear. En este caso 4096M que corresponden a los 4G.
2. El siguiente paso es crear el dispositivo de bloque sobre el fichero que hemos creado con el que trabajaremos para crear el sistema de ficheros, mediante el comando `losetup`:
   - Antes de crearlo, tenemos que ver los dispositivos `loop` que están disponibles para asociarlo con el fichero. Para ello, lanzamos el comando `losetup -f` y nos devuelve que el único dispositivo disponible es `/dev/loop0`.
   - Ahora lo único que tenemos que hacer es ejecutar este comando `sudo losetup /dev/loop0 fichero`. Es necesario usar permisos de administrador, por lo que se lanzará el comando con `sudo`.
3. Con el dispositivo de tipo bloque, le asignamos un sistema de fichero cualquiera con `mkfs`. En mi caso, le asigno el mismo que el que tiene la partición principal: `sudo mkfs.ext4 /dev/loop0`.
4. Lo último es montar ese sistema de fichero nuevo en un directorio (`/mnt` debido a que está dedicado a montar dispositivos).
5. Para comprobar que lo hemos montado correctamente, usamos el comando `lsblk` para ver todas las particiones montadas.

```
dansana@vm3803:/mnt$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0    7:0    0    4G  0 loop /mnt
sda      8:0    0   60G  0 disk
├─sda1   8:1    0    1G  0 part /boot/efi
└─sda2   8:2    0 58.9G  0 part /
sr0     11:0    1  2.6G  0 rom
```

Cuando tengamos el dispositivo de disco `Loop`, nos piden administrar las particiones en ese dispositivo:

- Para crear una partición, haremos uso de la herramienta de `fdisk`. No tendrá ningún valor específico, por lo que se deja todo por defecto.
- Puede ser que el kernel no pueda actualizar automáticamente la tabla de particiones al terminar, por lo que habrá que desanclar y volver anclar el `Loop`.
- Al igual que hemos hecho con el dispositivo, habrá que formatear esa partición y asignarle un sistema de ficheros. En este caso el mismo que tiene el propio dispositivo.
- Después, se monta la partición con el comando `mount` sobre el directorio `/mnt` y se comprueba lanzando un `df -h`. Tras esto, se desmonta con `umount`.
- Ahora nos piden un resumen acerca de la función de los ficheros `/etc/fstab` y `/etc/mtab`:
  - **/etc/fstab**: Es un fichero de configuración estático que define qué sistemas de archivos hay en el sistema y cómo deben montarse. La máquina lo consulta durante el arranque para montar automáticamente discos, particiones o sistemas de ficheros de red.
  - **/etc/mtab**: Es un fichero dinámico, generado por el sistema, que refleja qué sistemas de ficheros están montados en este momento. En sistemas modernos, muchas veces /etc/mtab es un enlace simbólico a /proc/self/mounts, que cumple la misma función.
- Por último, nos indican eliminar la partición existente en `loop0` y crear varias particiones primarias y extendidas o lógicas. Además, cada partición tiene que tener un sistema de ficheros independiente. Este sería el esquema resultante:

```
Command (m for help): p
Disk /dev/loop0: 4 GiB, 4294967296 bytes, 8388608 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x657ff2b2

Device       Boot   Start     End Sectors  Size Id Type
/dev/loop0p1         2048 2099199 2097152    1G 83 Linux
/dev/loop0p2      2099200 4196351 2097152    1G 83 Linux
/dev/loop0p3      4196352 8388607 4192256    2G  5 Extended
/dev/loop0p5      4198400 6295551 2097152    1G 83 Linux
/dev/loop0p6      6297600 8388607 2091008 1021M 83 Linux
```

```
dansana@vm3803:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           574M  1.2M  573M   1% /run
efivarfs         56K   27K   25K  52% /sys/firmware/efi/efivars
/dev/sda2        58G  7.1G   48G  13% /
tmpfs           2.9G     0  2.9G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/sda1       1.1G  6.2M  1.1G   1% /boot/efi
tmpfs           587M  8.0K  587M   1% /run/user/1000
/dev/loop0p1    974M   24K  907M   1% /mnt/p1
/dev/loop0p2    960M   51M  910M   6% /mnt/p2
/dev/loop0p5   1022M  4.0K 1022M   1% /mnt/p5
/dev/loop0p6    988M   44K  937M   1% /mnt/p6
dansana@vm3803:~$
```

1.6.2 LVM

Llegados a este punto, se requiere volver a destruir las particiones existentes para crear y administrar volúmenes lógicos (LVM).

1. Primero, modificamos la etiqueta de cada partición para marcar que es de tipo `Linux LVM`. Desde la herramienta de `fdisk`, seleccionamos cada partición del dispositivo `loop0` y con la opción `t`, escogemos el número 44 (en este caso que usamos una tabla de particiones de tipo `GPT`).
2. Después, creamos el volumen físico en cada partición con `pvcreate`:

```
dansana@vm3803:~$ sudo pvcreate /dev/loop0p1
  Physical volume "/dev/loop0p1" successfully created.
dansana@vm3803:~$ sudo pvcreate /dev/loop0p2
  Physical volume "/dev/loop0p2" successfully created.
dansana@vm3803:~$ sudo pvcreate /dev/loop0p3
  Physical volume "/dev/loop0p3" successfully created.
```

3. Lo siguiente, es crear un grupo de volúmenes físicos, con `vgcreate`, donde agregamos los que hemos creado:

```
dansana@vm3803:~$ sudo vgcreate vg_practica /dev/loop0p1 /dev/loop0p2 /dev/loop0p3
  Volume group "vg_practica" successfully created
dansana@vm3803:~$ sudo vgdisplay
  --- Volume group ---
  VG Name               vg_practica
  System ID
  Format                lvm2
  Metadata Areas        3
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                3
  Act PV                3
  VG Size               <3.99 GiB
  PE Size               4.00 MiB
  Total PE              1021
  Alloc PE / Size       0 / 0
  Free  PE / Size       1021 / <3.99 GiB
  VG UUID               aUlQ83-5hWC-8UJY-dpB1-Q4wb-P7Yw-Ktik5g
```

4. Por último, creamos un par de volúmenes lógicos, con `lvcreate`, sobre ese grupo que hemos creado en el punto anterior para después montarlo en el sistema:

```
dansana@vm3803:~$ sudo lvcreate -n lv_datos -L 500M vg_practica
  Logical volume "lv_datos" created.
dansana@vm3803:~$ sudo lvcreate -n lv_backup -L 300M vg_practica
  Logical volume "lv_backup" created.
dansana@vm3803:~$ ls /dev/vg_practica/
lv_backup  lv_datos
```

5. Una vez ya tenemos los volúmenes lógicos, los formateamos ambos para asignarles una estructura de directorios y los montamos en el directorio `/mnt` para comprobar que se ha creado de forma correcta:

```
dansana@vm3803:~$ sudo mkdir -p /mnt/lv_datos
dansana@vm3803:~$ sudo mkdir -p /mnt/lv_backup
dansana@vm3803:~$ sudo mount /dev/vg_practica/lv_datos /mnt/lv_datos
dansana@vm3803:~$ sudo mount /dev/vg_practica/lv_backup /mnt/lv_backup
dansana@vm3803:~$ df -h
Filesystem                         Size  Used Avail Use% Mounted on
tmpfs                              575M  1.2M  574M   1% /run
efivarfs                            56K   36K   16K  70% /sys/firmware/efi/efiv
/dev/sda2                           58G  7.2G   48G  13% /
tmpfs                              2.5G     0  2.5G   0% /dev/shm
tmpfs                              5.0M     0  5.0M   0% /run/lock
/dev/sda1                          1.1G  6.2M  1.1G   1% /boot/efi
tmpfs                              421M  8.0K  421M   1% /run/user/1000
/dev/mapper/vg_practica-lv_datos   452M   24K  417M   1% /mnt/lv_datos
/dev/mapper/vg_practica-lv_backup  236M   20M  217M   9% /mnt/lv_backup
```

6. Para acabar con este apartado, retornamos el sistema a su estado anterior desmontando y eliminando los volúmenes:

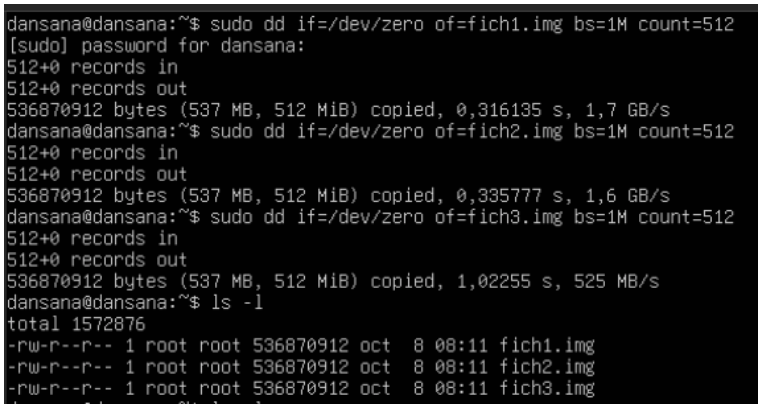
```
dansana@vm3803:~$ sudo lvremove /dev/vg_practica/lv_datos
Do you really want to remove and DISCARD logical volume vg_practica/lv_datos? [y/n]: y
  Logical volume "lv_datos" successfully removed.
dansana@vm3803:~$ sudo lvremove /dev/vg_practica/lv_backup
Do you really want to remove and DISCARD logical volume vg_practica/lv_backup? [y/n]: y
  Logical volume "lv_backup" successfully removed.
dansana@vm3803:~$ sudo vgremove vg_practica
  Volume group "vg_practica" successfully removed
dansana@vm3803:~$ sudo pvremove /dev/loop0p1 /dev/loop0p2 /dev/loop0p3
  Labels on physical volume "/dev/loop0p1" successfully wiped.
  Labels on physical volume "/dev/loop0p2" successfully wiped.
  Labels on physical volume "/dev/loop0p3" successfully wiped.
```

1.7 Administración de almacenamiento

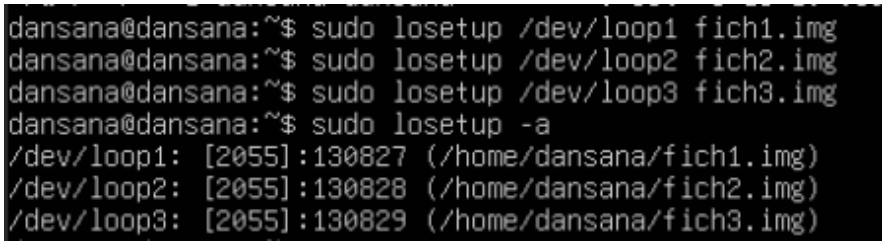
1.7.1 RAID 5

Lo último que vamos a hacer antes de realizar una nueva instalación es crear y administrar un `RAID 5` mediante software:

1. Lo primero es crear 3 nuevos dispositivos `loop` de la misma manera que lo hemos hecho en el apartado Loop:

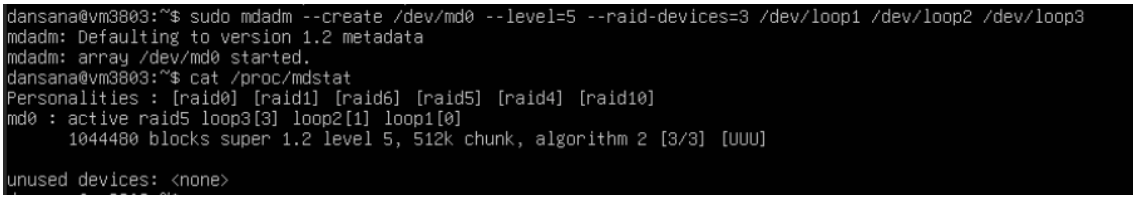


   Y después asociarlo a 3 dispositivos `loop`:

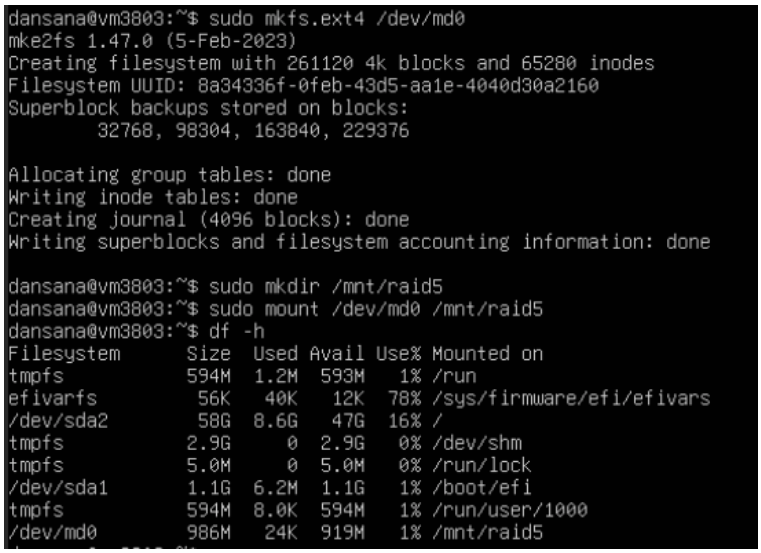


2. Consultando el manual para crear el dispositivo `RAID`, tenemos que seleccionar el modo `Create`, con las opciones de:

   - `level`: Indicando el tipo de `RAID`.
   - `raid-devices`: Número de dispositivos que usaremos.



3. Ahora repetimos el mismo procedimiento que con el resto de dispositivos de almacenamientos para formatearlos y darle un sistema de ficheros y montarlo:

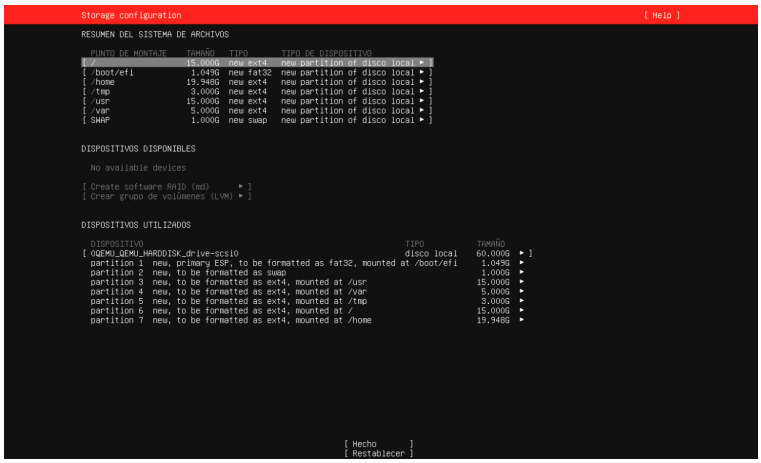


Para comprobar que hemos configurado correctamente el disco, ejecutamos el comando `echo \Voy a destruir el disco!!!!" > fichero.img` donde `fichero.img` es uno de los ficheros que da soporte al `RAID` para hacerlo fallar y ver que sigue funcionando. Al lanzarlo, veo que no se destruye el disco y sigue activo, debido a que el dispositivo `RAID 5` está montado y funcionando. Entonces, lo que hay que hacer es reiniciar la máquina para que deje de estar en funcionamiento y lanzar el comando.

1.8 Nueva instalación personalizada

Hasta ahora hemos trabajo con una configuración "por defecto"sobre la administración del disco de la máquina, teniendo únicamente dos particiones: `/boot/efi` empleada para el arranque del sistema y `/` para el resto de archivos del equipo. En un entorno real, tenemos más particiones para minimizar posibles fallos y errores, por lo que vamos a realizar una nueva instalación con las siguientes particiones:

- `/boot/efi`: Contiene los ficheros para el arranque del sistema y tiene un espacio de `1.049G`, que es lo que ocupaba inicialmente y no se puede modificar. El formato que tiene es `fat32`, bastante antiguo y limitado, pero compatible con muchos dispositivos.
- `/`: Destinado a todos los ficheros para el sistema operativo y con un espacio de `15G` para que no haya problemas a la hora de agregar elementos al sistema.
- `swap`: Con un espacio de `1G`, solo actúa en caso de que la memoria RAM se quede sin espacio. No es necesario añadir nada más ya que el sistema ya cuenta con 6GB de RAM.
- `/usr`: Donde se ubica los programas y librerías instaladas, por lo que es necesario `15G` de almacenamiento.
- `/var`: Lugar donde se ubican los ficheros como logs y colas. No es necesario dar mucho tamaño, por lo que se le asignan `3GB`.
- `/tmp`: Contiene archivos temporales y es necesario separarlo para que en caso de que se ocupe por completo, no bloquee el sistema. Solo es necesario `5G`.
- `/home`: Dedicada al espacio personal del usuario que ocupa el resto del espacio restante del disco.



1.9 Trabajo No Presencial

- **Administración de discos – particiones**:
  - Los discos duros o dispositivos de bloques, se dividen en unidades lógicas llamadas *particiones*.[8]
  - Una partición sirve para organizar y almacenar el sistema operativo, las aplicaciones y los archivos personales. Existen diferentes esquemas para la distribución de particiones en un disco, como MBR o GPT.
  - Cada partición se representa como un archivo en el sistema de archivos de Linux y se encuentra ubicada en el directorio `/dev`.
- **Sistemas de archivos**:
  - Es un elemento que controla cómo se almacenan y recuperan los datos. Sin un sistema de archivos, los datos colocados en un medio de almacenamiento serían un gran cuerpo de datos sin manera de saber dónde termina un dato y comienza el siguiente.
  - Se organizan en una estructura jerárquica, de tipo árbol. El nivel más alto del sistema de ficheros es / o directorio raíz. Todos los demás ficheros y directorios están bajo el directorio raíz.[4]
  - Por debajo del directorio raíz (/) hay un importante grupo de directorios común a la mayoría de las distribuciones de GNU/Linux: `/bin, /boot, /etc/, /opt, etc`.
  - Tipos de sistemas de ficheros más utilizados en la actualidad:
    - `EXT`: Con sus versiones `ext2`, `ext3` y `ext4` (siendo esta la más usada en sistemas Linux), fue creado para sobrepasar las limitaciones de `MINIX` y consiguió implementar `VFS`.
    - `XFS`: Manejo de grandes volúmenes de datos, por lo que es muy usado en servidores y sistemas empresariales.
    - `NTFS`: Sistema principal de `Windows` y con cierta compatibilidad con Linux.
    - `FAT32`: Antiguo y limitado, pero muy compatible con todos los sistemas.
    - `exFAT`: Evolución del anterior sistema y diseñado especialmente para memorias flash.
- **Actualización de un sistema operativo previamente instalado**:
  - En el caso de nuestra máquina virtual, estamos trabajando con `Ubuntu` que pertenece al grupo de distribuciones `Debian`, por lo que para actualizar el sistema operativo una vez instalado se hará uso de la herramienta `apt`.
  - `apt` nos proporciona un sistema de gestión de paquetes donde maneja automáticamente las dependencias para la instalación de esos paquetes. Requiere de privilegios administrativos.[5]
  - Para las actualizaciones será necesario usar los comandos `sudo apt update` y `sudo apt upgrade`.

- **Identificación discos duros y particiones**:
  - En Linux, los dispositivos se representan dentro del directorio `/dev` y se identifican como dispositivos de bloques (`sda, sdb, sdc, etc.` o `nvme0n1, nvme0n2, nvme0n3, etc.`).
  - Además, las particiones, tal y como se mencionaba en el primer apartado, son unidades lógicas de estos dispositivos y se identifican numerándose en orden seguido del nombre del dispositivo (`sda1, sda2, sda3, etc.` o `nvme0n1p1, nvme0n1p2, nvme0n1p3, etc.`).[8]
  - También, cada partición puede tener un `UUID` único, que no cambia aunque el disco se conecte en distinto orden.
- **RAID**:
  - `RAID` o Redundant Array of Independent Disks hace referencia a un sistema de almacenamiento de datos que utiliza múltiples discos duros, entre las cuales se distribuyen o replican los datos. [6]
  - Estas son las principales configuraciones de `RAID`:
    - **RAID 0**: Distribuye los datos equitativamente entre dos o más discos sin información de paridad que proporcione redundancia. No tiene tolerancia a fallos, si falla un disco, lo pierdes todo.
    - **RAID 1**: Crea una copia exacta de un conjunto de datos en dos o más discos. Puede fallar solo un disco para no perder todos los datos.
    - **RAID 5**: División de datos a nivel de bloques que distribuye la información de paridad entre todos los discos miembros del conjunto. Esta variante de `RAID` ha logrado popularidad gracias a su bajo coste de redundancia. Puede tolerar 1 disco defectuoso; reconstrucción en curso mientras funciona.
    - **RAID 6**: amplía el nivel RAID 5 añadiendo otro bloque de paridad, por lo que divide los datos a nivel de bloques y distribuye los dos bloques de paridad entre todos los miembros del conjunto. Puede tolerar 2 discos defectuosos; más seguro que RAID 5 en entornos con discos grandes.

2. Administración de usuarios y servicios

2.1 Gestión de usuarios y grupos

Para comprobar si la clave de nuestro usuario y el resto de usuarios está encriptada, tenemos que ver si en el fichero `/etc/passwd`, dedicado a recopilar la información de un usuario, en el segundo campo aparece escrita la clave o se muestra en su lugar el carácter `'x'`. Con solo lanzar un `cat` a ese fichero, descubrimos que cada usuario tiene su clave encriptada:

```
dansana@dansana:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon,,,:/usr/lib/dhcpcd:/bin/false
messagebus:x:101:102::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:992:992:systemd Resolver:/:/usr/sbin/nologin
pollinate:x:102:1::/var/cache/pollinate:/bin/false
polkitd:x:991:991:User for polkitd:/:/usr/sbin/nologin
syslog:x:103:104::/nonexistent:/usr/sbin/nologin
uuidd:x:104:105::/run/uuidd:/usr/sbin/nologin
tcpdump:x:105:107::/nonexistent:/usr/sbin/nologin
tss:x:106:108:TPM software stack,,,:/var/lib/tpm:/bin/false
landscape:x:107:109::/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:989:989:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin
usbmux:x:108:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
dansana:x:1000:1000:Daniel:/home/dansana:/bin/bash
```

Con esto concluimos que se está usando el fichero `/etc/shadow` para contener la clave encriptada.

Creamos dos nuevos grupos de usuario con `addgroup` y dos usuarios para cada grupo con `useradd`, para crearlos, y `adduser`, para agregarlo a uno de los grupos:

```
info: Selecting GID from range 1000 to 59999 ...
info: Adding group `grupo1' (GID 1001) ...
dansana@dansana:~$ sudo addgroup grupo2
info: Selecting GID from range 1000 to 59999 ...
info: Adding group `grupo2' (GID 1002) ...
```

```
dansana@dansana:~$ sudo adduser user1 grupo1
info: Adding user `user1' to group `grupo1' ...
dansana@dansana:~$ sudo adduser user2 grupo2
info: Adding user `user2' to group `grupo2' ...
```

Tras la creación de los nuevos usuarios y grupos, volvemos a ver el contenido del fichero `/etc/passwd` y también de `/etc/group`, que contiene información de cada grupo y los usuarios que hay dentro de él:

```
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
dansana:x:1000:1000:Daniel:/home/dansana:/bin/bas
user1:x:1001:1003::/home/user1:/bin/sh
user2:x:1002:1004::/home/user2:/bin/sh
```

```
grupo1:x:1001:user1
grupo2:x:1002:user2
user1:x:1003:
user2:x:1004:
```

Y para el `password` de cada usuario, lo recomendable es seguir estos consejos:

- **Longitud de la contraseña**: Debería haber entre 12 y 16 caracteres.

- **Uso de distintos tipos de caracteres**: Mezclar entre símbolos y letras mayúsculas y minúsculas, incluso números.
- **No utilizar secuencias conocidas**.

La siguiente tarea es tener una estructura de directorios/ficheros tipo en el directorio `/etc/skel`, encargado de otorgar esa jerarquía al directorio `/home/<user>`. Decidí dar una estructura parecida a la que otorga `Ubuntu Desktop` en su instalación, usando `mkdir`:

- Downloads
- Documents
- Images
- Desktop

Al crear otro par de usuarios y agregarlos cada uno a los grupos nuevos, la estructura de ficheros de sus directorios `/home` tendrá esa jerarquía:

```
dansana@dansana:~$ sudo useradd -m user3
dansana@dansana:~$ sudo useradd -m user4
dansana@dansana:~$ sudo passwd user3
New password:
Retype new password:
passwd: password updated successfully
dansana@dansana:~$ sudo passwd user4
New password:
Retype new password:
passwd: password updated successfully
dansana@dansana:~$ sudo adduser user3 grupo1
info: Adding user `user3' to group `grupo1' ...
dansana@dansana:~$ sudo adduser user4 grupo2
info: Adding user `user4' to group `grupo2' ...
dansana@dansana:~$ ls /home/user3
ls: cannot open directory '/home/user3': Permission denied
dansana@dansana:~$ sudo ls /home/user3
Desktop  Documents  Downloads  Images
dansana@dansana:~$ sudo ls -la /home/user3
total 36
drwxr-x--- 6 user3 user3 4096 oct  6 17:17 .
drwxr-xr-x 8 root  root  4096 oct  6 17:17 ..
-rw-r--r-- 1 user3 user3  220 mar 31  2024 .bash_logout
-rw-r--r-- 1 user3 user3 3771 mar 31  2024 .bashrc
drwxr-xr-x 2 user3 user3 4096 oct  6 17:12 Desktop
drwxr-xr-x 2 user3 user3 4096 oct  6 17:11 Documents
drwxr-xr-x 2 user3 user3 4096 oct  6 17:11 Downloads
drwxr-xr-x 2 user3 user3 4096 oct  6 17:12 Images
-rw-r--r-- 1 user3 user3  807 mar 31  2024 .profile
dansana@dansana:~$ sudo ls -la /home/user4
total 36
drwxr-x--- 6 user4 user4 4096 oct  6 17:17 .
drwxr-xr-x 8 root  root  4096 oct  6 17:17 ..
-rw-r--r-- 1 user4 user4  220 mar 31  2024 .bash_logout
-rw-r--r-- 1 user4 user4 3771 mar 31  2024 .bashrc
drwxr-xr-x 2 user4 user4 4096 oct  6 17:12 Desktop
drwxr-xr-x 2 user4 user4 4096 oct  6 17:11 Documents
drwxr-xr-x 2 user4 user4 4096 oct  6 17:11 Downloads
drwxr-xr-x 2 user4 user4 4096 oct  6 17:12 Images
-rw-r--r-- 1 user4 user4  807 mar 31  2024 .profile
```

2.2 Gestión de password

Se puede establecer una fecha de caducidad a la contraseña de un usuario y de su cuenta con el comando `chage` que estable dicho atributo. Hay que tener en cuenta que el sistema está 2 horas retrasado de la hora real (se puede comprobar con el comando `date`). Para la contraseña, hay que usar la opción `-M` para indicar el máximo de días que tiene esa contraseña antes de caducar, y la opción `-E` indica la fecha tope de ese usuario:

```
dansana@dansana:~$ sudo chage -M 1 user3
dansana@dansana:~$ sudo chage -E 2025-10-07 user3
dansana@dansana:~$ sudo chage -l user3
Last password change                                    : oct 06, 2025
Password expires                                        : oct 07, 2025
Password inactive                                       : never
Account expires                                         : oct 07, 2025
Minimum number of days between password change          : 0
Maximum number of days between password change          : 1
Number of days of warning before password expires       : 7
```

Para comprobar que funciona, podemos verlo en el fichero `/etc/shadow` ya que al tener la información de las claves de los usuarios, se indica también cuando es su fecha de espiración:

```
user3:$y$j9T$Y38Bi9bIm0Z4zzVyGD6sc.$jTqxpM.Nyr3izjHnBaq8P/NsyftZgeSSxiifbJpxM9B:20367:0:1:7::20368:
```

2.3 Gestión avanzada de grupos

Nos piden que uno de los nuevos usuarios creados forme parte de un de los dos grupos. Compruebo viendo el fichero `/etc/group` que cada usuario está asignado a un grupo, por lo que únicamente habrá usar `usermod` con las opciones `-aG` para añadir a ese usuario a un grupo a mayores. En mi caso lo pruebo con el usuario `user4`.

```
grupo1:x:1001:user1,user3,user4
grupo2:x:1002:user2,user4
```

Una de las ventajas de tener un usuario en varios grupos puede darse a la hora de asignar permisos sobre un recurso compartido, ya que un solo usuario puede tener acceso y privilegios a recursos de ambos grupos.

```
dansana@dansana:~$ sudo chown :grupo1 ejemploGrupo1 -R
dansana@dansana:~$ sudo chown :grupo2 ejemploGrupo2 -R
dansana@dansana:~$ sudo chmod 770 ejemploGrupo1 ejemploGrupo2
dansana@dansana:~$ ls -la
total 1048644
drwxr-x--- 6 dansana dansana      4096 oct  8 17:43 .
drwxr-xr-x 8 root    root         4096 oct  6 17:17 ..
-rw------- 1 dansana dansana      1737 oct  8 08:24 .bash_history
-rw-r--r-- 1 dansana dansana       220 mar 31  2024 .bash_logout
-rw-r--r-- 1 dansana dansana      3771 mar 31  2024 .bashrc
drwx------ 2 dansana dansana      4096 oct  6 14:46 .cache
drwxrwx--- 2 dansana grupo1       4096 oct  8 17:41 ejemploGrupo1
drwxrwx--- 2 dansana grupo2       4096 oct  8 17:42 ejemploGrupo2
-rw-r--r-- 1 root    root           29 oct  8 16:38 fich1.img
-rw-r--r-- 1 root    root    536870912 oct  8 16:28 fich2.img
-rw-r--r-- 1 root    root    536870912 oct  8 16:28 fich3.img
-rw------- 1 dansana dansana        20 oct  8 17:43 .lesshst
-rw-r--r-- 1 root    root            9 oct  8 08:41 nuevo
-rw-r--r-- 1 dansana dansana       807 mar 31  2024 .profile
drwx------ 2 dansana dansana      4096 oct  6 14:42 .ssh
-rw-rw-r-- 1 dansana dansana        40 oct  8 08:25 sudo
-rw-r--r-- 1 dansana dansana         0 oct  6 16:10 .sudo_as_admin_successful
-rw------- 1 dansana dansana      1334 oct  8 17:42 .viminfo
```

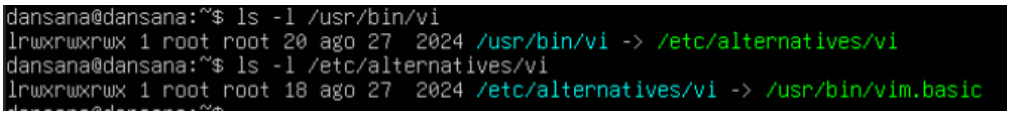
A mayores, tenemos que asignarle una clave a un grupo, con `sudo gpasswd group`. En la captura podemos ver la clave encriptada del `grupo1`:

Esto puede tener varias ventajas, entre ellas el acceso temporal a los recursos que puede tener ese grupo, de forma que el administrador no tiene que estar agregando o excluyendo usuarios del grupo.

2.4 Perfiles de usuario

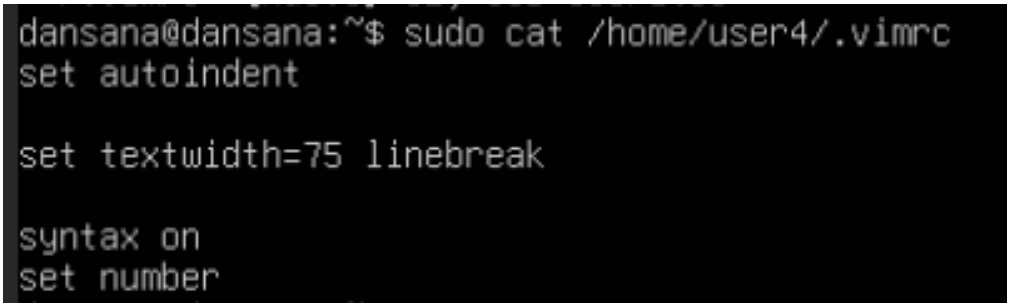
En este apartado, se pide modificar la configuración del editor de texto `vi`, de forma que para uno de los usuarios creados (por ejemplo `user4`) se realicen dos operaciones automáticamente cuando se abra el programa, por lo que será necesario modificar el archivo con la configuración de este editor.

Como ya se ha comentado en la instalación, estamos usando `Ubuntu Server 24` en nuestras máquinas virtuales y, consultando la información en foros [3], en sistemas modernos, utiliza el mismo fichero de configuración que `vim` que es `.vimrc`, porque el comando `vi` es un enlace simbólico al editor `vim`.

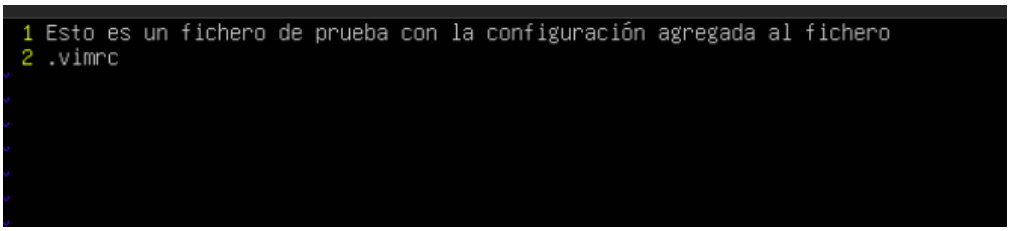


En `/home/user4`, no existe ese fichero de configuración, por lo que habrá que crearlo y agregar las opciones requeridas.

- Para la auto-sangría, tenemos que activar la opción `autoindent`.
- Y para un máximo de 75 caracteres con un salto de línea, la opción `textwidth=75 linebreak`.



Las dos últimas líneas son configuración adicional que he añadido para agregar color al texto y que aparezcan los números de línea por pantalla.



Finalizado esta primera personalización, ahora nos piden que cada vez que el usuario inicie sesión, se muestre el contenido de `$HOME/docs/Agenda.txt`. Haciendo uso de `Shell scripting`, modificamos el fichero `.profile` porque es el que se ejecuta cuando se inicia sesión en un usuario:

```
cat .profile
# ~/.profile: executed by the command interpreter for login shells.
# This file is not read by bash(1), if ~/.bash_profile or ~/.bash_login
# exists.
# see /usr/share/doc/bash/examples/startup-files for examples.
# the files are located in the bash-doc package.

# the default umask is set in /etc/profile; for setting the umask
# for ssh logins, install and configure the libpam-umask package.
#umask 022

# if running bash
if [ -n "$BASH_VERSION" ]; then
    # include .bashrc if it exists
    if [ -f "$HOME/.bashrc" ]; then
        . "$HOME/.bashrc"
    fi
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/.local/bin" ] ; then
    PATH="$HOME/.local/bin:$PATH"
fi

#Aparece el contenido de Agenda.txt
if [ -f "$HOME/Documents/Agenda.txt" ]; then
        cat "$HOME/Documents/Agenda.txt"
fi
```

Reiniciando, veremos lo que hay escrito en `Agenda.txt` ubicado en la última línea de todo el contenido que se muestra tras iniciar sesión.

```
Ubuntu 24.04.1 LTS dansana tty1

dansana login: dansana
Password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-85-generic x86_64)

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

 System information as of jue 09 oct 2025 16:42:50 UTC

  System load:    0.0               Processes:               162
  Usage of /home: 5.1% of 19.47GB   Users logged in:          0
  Memory usage:   5%                IPv4 address for enp6s18: 10.0.38.3
  Swap usage:     0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 129 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»


AGENDA:
dansana@dansana:~$ _
```

También, se requiere que cunado el usuario abre un `shell`, se muestre la fecha y hora del último login. En este caso, el fichero `.bashrc` es el que contiene la configuración de cualquier `shell`. Al final del fichero, agregamos `lastlog -u "$USER"` con `tail -n 1`, donde `lastlog` es el comando que muestra el login más reciente de un usuario y `tail` es para mostrar solo la última línea del registro.

```
109 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
110 # sources /etc/bash.bashrc).
111 if ! shopt -oq posix; then
112   if [ -f /usr/share/bash-completion/bash_completion ]; then
113     . /usr/share/bash-completion/bash_completion
114   elif [ -f /etc/bash_completion ]; then
115     . /etc/bash_completion
116   fi
117 fi
118
119 # Muestra la fecha y hora del último login
120 lastlog -u "$USER" | tail -n 1
```

Al estar una máquina virtual, no estamos interactuando con una `tty` real", por lo que nunca se registrará el inicio de sesión.

```
Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»


dansana                                                  **Never logged in**
AGENDA:
dansana@dansana:~$
```

Y, finalmente, la última personalización es que cada vez que se cierre sesión, se muestre un mensaje de despedida y solo se cierre sesión tras haber respondido al mensaje. Al hacer `logout`, el sistema ejecuta el fichero `.bash_logout` con todas las instrucciones que contiene. Por ello, modificamos dicho archivo para que mediante un `echo` muestre el mensaje y se quede esperando con un `read`:

```
dansana@dansana:~$ cat .bash_logout
# ~/.bash_logout: executed by bash(1) when login shell exits.

echo "Que pase un buen dia $USER"
read -p "Pulsa una tecla para cerrar sesion..."

# when leaving the console clear the screen to increase privacy

if [ "$SHLVL" = 1 ]; then
    [ -x /usr/bin/clear_console ] && /usr/bin/clear_console -q
fi
```

```
dansana@dansana:~$ logout
Que pase un buen dia dansana
Pulsa una tecla para cerrar sesion..._
```

2.5 Servicios del sistema

En este apartado, se trabajará acerca de los servicios que hay en el sistema, destacando el de conexión segura o `ssh`. Para comprobar los servicios activos, lanzamos el comando
`systemctl list-units --type=service --state=running` donde señalamos que en el listado de servicios nos interesa aquellos que están activos:

```
dansana@dansana:~$ systemctl list-units --type=service --state=running
  UNIT                          LOAD   ACTIVE SUB     DESCRIPTION
  cron.service                  loaded active running Regular background program processing daemon
  dbus.service                  loaded active running D-Bus System Message Bus
  getty@tty1.service            loaded active running Getty on tty1
  getty@tty6.service            loaded active running Getty on tty6
  ModemManager.service          loaded active running Modem Manager
  multipathd.service            loaded active running Device-Mapper Multipath Device Controller
  polkit.service                loaded active running Authorization Manager
  rsyslog.service               loaded active running System Logging Service
  systemd-journald.service      loaded active running Journal Service
  systemd-logind.service        loaded active running User Login Management
  systemd-networkd.service      loaded active running Network Configuration
  systemd-resolved.service      loaded active running Network Name Resolution
  systemd-timesyncd.service     loaded active running Network Time Synchronization
  systemd-udevd.service         loaded active running Rule-based Manager for Device Events and Files
  udisks2.service               loaded active running Disk Manager
  unattended-upgrades.service loaded active running Unattended Upgrades Shutdown
  upower.service                loaded active running Daemon for power management
  user@1000.service             loaded active running User Manager for UID 1000

Legend: LOAD   → Reflects whether the unit definition was properly loaded.
        ACTIVE → The high-level unit activation state, i.e. generalization of SUB.
        SUB    → The low-level unit activation state, values depend on unit type.

18 loaded units listed.
```

Cada proceso nos indica el nombre (`UNIT`), si está cargado en el sistema (`LOAD`), si está activo (`ACTIVE`), su estado (`SUB`) y una descripción (`DESCRIPTION`). Observando la lista, vemos que no aparece el servicio `ssh` porque está inactivo:

```
dansana@dansana:~$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
     Active: inactive (dead)
TriggeredBy: • ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
```

Vemos que está desactivado, por lo que habrá que activarlo con `sudo systemctl enable ssh`, después se para con el mismo comando pero usando `stop` en vez de `enable`, y se iniciará con `start`.

```
dansana@dansana:~$ systemctl stop ssh
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to stop 'ssh.service'.
Authenticating as: Daniel (dansana)
Password:
==== AUTHENTICATION COMPLETE ====
Stopping 'ssh.service', but its triggering units are still active:
ssh.socket
dansana@dansana:~$ systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ====
Authentication is required to reload the systemd state.
Authenticating as: Daniel (dansana)
Password:
==== AUTHENTICATION COMPLETE ====
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ====
Authentication is required to reload the systemd state.
Authenticating as: Daniel (dansana)
Password:
==== AUTHENTICATION COMPLETE ====
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files ====
Authentication is required to manage system service or unit files.
Authenticating as: Daniel (dansana)
Password:
==== AUTHENTICATION COMPLETE ====
dansana@dansana:~$ systemctl start ssh
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'ssh.service'.
Authenticating as: Daniel (dansana)
Password:
==== AUTHENTICATION COMPLETE ====
```

Es esencial tener un equipo con los servicios esenciales activos y desactivar aquellos que no tienen mucha relevancia, ya que consumen recursos innecesarios y pueden llegar a ser un riesgo en la seguridad. Investi-

gando el listado de servicios que hay activos en mi máquina, hay ciertos servicios que se pueden quitar como `ModemManager.servicie` debido a que solo se emplea para la gestión de módems 3G/4G.

Y para terminar con esta sección, nos piden que el sistema desde el arranque vaya al nivel de ejecución 3 (`runlevel3`), es decir sin entorno gráfico y solo en modo texto con red. Dependiendo de cada `runlevel` entre 1 y 5, Linux se ejecutará de una forma distinta. Para arrancar como ese nivel de ejecución `sudo systemctl set-default multi-user.target`

```
dansana@dansana:~$ sudo systemctl set-default multi-user.target
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/multi-user.target.
```

2.6 Permisos de acceso

Vamos a aprender acerca de cómo funcionan los permisos en sistemas Linux [2]. En los sistemas operativos tipo `POSIX` cada elemento del sistema de archivos tiene la característica de poseer permisos que lo ubican dentro del mismo. Éstos sirven como uno más de los niveles de seguridad del sistema operativo al impedir que cualquier usuario pueda leer, escribir, ejecutar o acceder a dichos archivos y directorios de manera arbitraria. Estos permisos vistos de manera básica son: lectura (r, `read`), escritura (w, `write`) y ejecución (x, `execution`) y se agrupan en bloques (`rwx`) para 3 diferentes clases (usuario, grupo y otros).

Por defecto, cuando se crea un elemento en el sistema de fichero, se le asignará unos permisos dependiendo del objeto creado. `umask` es una función que establece los permisos predeterminados para los nuevos archivos y directorios creados en el sistema. El valor de la máscara de usuario, que se asigna ejecutando `umask`, corresponde a los bits contrarios del permiso predeterminado que se quiera asignar. Es decir, si por ejemplo se quiere asignar una máscara de usuario equivalente a 0775 (`rwxrwxr-x`), el valor de la máscara de usuario corresponderá a 0002 (el resultado de la operación 777 menos 775), que será lo mismo que definir `u=rwx,g=rwx,o=rx`. En este caso, para lograr que cuando se cree un fichero tenga los permisos 550, tenemos que saber que los permisos por defecto son 664 y realizar el siguiente cálculo: `550 = 664 - umask`. El resultado de `umask` es 114 en octal. Para aplicarlo a todos los usuarios, tenemos que agregar al final del fichero `/etc/profile` esta línea `umask 114`.

```
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "${PS1-}" ]; then
  if [ "${BASH-}" ] && [ "$BASH" != "/bin/sh" ]; then
    # The file bash.bashrc already sets the default PS1.
    # PS1='\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
      . /etc/bash.bashrc
    fi
  else
    if [ "$(id -u)" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi

umask 114
~
~
```

```
dansana@dansana:~$ ls -la
total 1048680
drwxr-x--- 10 dansana dansana      4096 oct 10 09:48 .
drwxr-xr-x  8 root    root         4096 oct  6 17:17 ..
-rw-------  1 dansana dansana      4113 oct  9 19:29 .bash_history
-rw-r--r--  1 dansana dansana       303 oct  9 17:20 .bash_logout
-rw-r--r--  1 dansana dansana      3849 oct  9 17:13 .bashrc
drwx------  2 dansana dansana      4096 oct  6 14:46 .cache
drwxrwxr-x  2 dansana dansana      4096 oct  9 16:33 Desktop
drwxrwxr-x  2 dansana dansana      4096 oct  9 16:34 Documents
drwxrwxr-x  2 dansana dansana      4096 oct  9 16:33 Downloads
drwxrwx---  2 dansana grupo1       4096 oct  8 17:41 ejemploGrupo1
drwxrwx---  2 dansana grupo2       4096 oct  8 17:42 ejemploGrupo2
-rw-r--r--  1 root    root          29 oct  8 16:38 fich1.img
-rw-r--r--  1 root    root    536870912 oct  8 16:28 fich2.img
-rw-r--r--  1 root    root    536870912 oct  8 16:28 fich3.img
drwxrwxr-x  2 dansana dansana      4096 oct  9 16:33 Images
-rw-------  1 dansana dansana        20 oct 10 09:42 .lesshst
-rw-r--r--  1 root    root           9 oct  8 08:41 nuevo
-rw-r--r--  1 dansana dansana       926 oct  9 16:21 .profile
drwx------  2 dansana dansana      4096 oct  6 14:42 .ssh
-rw-rw-r--  1 dansana dansana        40 oct  8 08:25 sudo
-rw-r--r--  1 dansana dansana         0 oct  6 16:10 .sudo_as_admin_successful
-rw-rw-r--  1 dansana dansana        28 oct 10 09:48 umaskPrueba.txt
-rw-------  1 dansana dansana      8218 oct 10 09:48 .viminfo
-rw-r--r--  1 root    root          21 oct  9 16:46 .vimrc
```

Acerca del bit `SetUID`, es un permiso especial que hace que cuando se ha establecido ejecución, el proceso resultante asumirá la identidad del usuario dado en la clase de usuario (propietario del elemento). Corresponde al primer dígito del conjunto octal de permisos de los cuatro que hay. Un ejemplo para ver cómo funciona sería creando un programa en C que me indique cuál es el valor del `UID` efectivo (permisos con los que ejecuta el usuario) y real (permisos de ejecución reales de ese usuario).

```c
#include <stdio.h>
#include <unistd.h>

int main() {
        printf("Efectivo: %d\n", geteuid());
        printf("Real: %d\n", getuid());
        return 0;
}
```

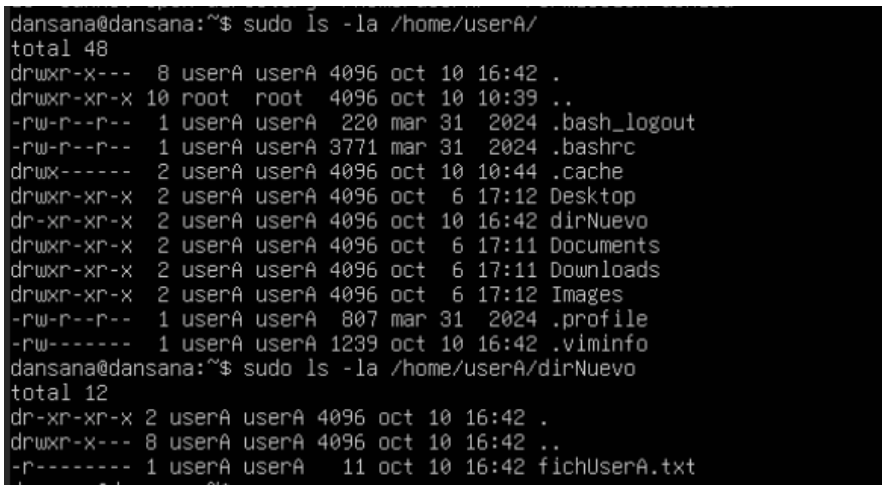
Ahora, le otorgamos permisos a `root` para comprobar cómo varía el valor del `UID`:

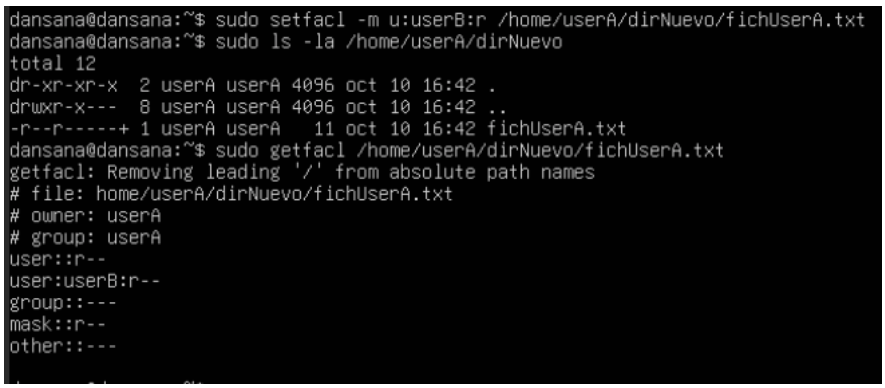
```
dansana@dansana:~$ ./muestraUID
Efectivo: 1000
Real: 1000
dansana@dansana:~$ sudo chown root:root muestraUID
dansana@dansana:~$ sudo chomd u+s muestraUID
sudo: chomd: command not found
dansana@dansana:~$ sudo chmod u+s muestraUID
dansana@dansana:~$ ls -la
total 1048700
drwxr-x--- 10 dansana dansana      4096 oct 10 10:28 .
drwxr-xr-x  8 root    root         4096 oct  6 17:17 ..
-rw-------  1 dansana dansana      4113 oct  9 19:29 .bash_history
-rw-r--r--  1 dansana dansana       303 oct  9 17:20 .bash_logout
-rw-r--r--  1 dansana dansana      3849 oct  9 17:13 .bashrc
drwx------  2 dansana dansana      4096 oct  6 14:46 .cache
drwxrwxr-x  2 dansana dansana      4096 oct  9 16:33 Desktop
drwxrwxr-x  2 dansana dansana      4096 oct  9 16:34 Documents
drwxrwxr-x  2 dansana dansana      4096 oct  9 16:33 Downloads
drwxrwx---  2 dansana grupo1       4096 oct  8 17:41 ejemploGrupo1
drwxrwx---  2 dansana grupo2       4096 oct  8 17:42 ejemploGrupo2
-rw-r--r--  1 root    root           29 oct  8 16:38 fich1.img
-rw-r--r--  1 root    root    536870912 oct  8 16:28 fich2.img
-rw-r--r--  1 root    root    536870912 oct  8 16:28 fich3.img
drwxrwxr-x  2 dansana dansana      4096 oct  9 16:33 Images
-rw-------  1 dansana dansana        20 oct 10 09:42 .lesshst
-rwsrwxr-x  1 root    root        16056 oct 10 10:26 muestraUID
-rw-rw-r--  1 dansana dansana       137 oct 10 10:26 muestraUID.c
-rw-r--r--  1 root    root           9 oct  8 08:41 nuevo
-rw-r--r--  1 dansana dansana       926 oct  9 16:21 .profile
drwx------  2 dansana dansana      4096 oct  6 14:42 .ssh
-rw-rw-r--  1 dansana dansana        40 oct  8 08:25 sudo
-rw-r--r--  1 dansana dansana         0 oct  6 16:10 .sudo_as_admin_suc
-rwSrw-r--  1 dansana dansana        28 oct 10 09:48 umaskPrueba.txt
-rw-------  1 dansana dansana      8228 oct 10 10:28 .viminfo
-rw-r--r--  1 root    root           21 oct  9 16:46 .vimrc
dansana@dansana:~$ ./muestraUID
Efectivo: 0
Real: 1000
```

2.7 Listas de acceso de control

Volvemos a crear dos usuarios nuevos y lo asignamos cada uno a un grupo nuevo distinto, de la misma manera que se ha hecho al comienzo de esta sección. Iniciando sesión en el `userA`, creamos el directorio con solo permisos de lectura y acceso, el cuál va a contener un fichero con solo permiso de lectura para el usuario `root`.

Ahora, mediante el uso de la utilidad `ACL`, damos permisos de lectura únicamente al usuario `userB` con el comando `sudo setfacl -m u:userB:r /home/userA/dirNuevo/fichUserA.txt` donde indicamos con la opción `-m` que se quiere modificar la `ACL` del fichero y señalamos el usuario que obtiene esos permisos:





NOTA: La utilidad no viene instalada en la máquina, hay que instalarla con `sudo apt install acl`.

2.8 Cuota de disco

Se trata de un límite establecido por el administrador del sistema que restringe ciertos aspectos del uso del sistema de archivos en los sistemas operativos modernos. El objetivo de la utilización de las cuotas de disco es limitar la asignación de espacio en el disco duro de una manera razonable [7]. Para activarla, tenemos que seguir estos pasos[1]:

1. Instalamos la herramienta con el comando `sudo apt install quota`.

2. Una vez instalada, tenemos que editar el fichero `/etc/fstab`, para activar las opciones de cuota en la partición `/home`, tanto del usuario como para el grupo.

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
/dev/disk/by-uuid/b7924c12-88fc-4477-ab50-ebc9346eb3f3 none swap sw 0 0
# / was on /dev/sda6 during curtin installation
/dev/disk/by-uuid/bf4ae9ff-8ca6-4d69-ad1a-2eeb46ba55c8 / ext4 defaults 0 1
# /home was on /dev/sda7 during curtin installation
/dev/disk/by-uuid/fc0a346f-23a9-445c-ad91-a521c1067e4a /home ext4 defaults,usrquota,grpquota 0 1
# /boot/efi was on /dev/sda1 during curtin installation
/dev/disk/by-uuid/4024-FA21 /boot/efi vfat defaults 0 1
# /usr was on /dev/sda3 during curtin installation
/dev/disk/by-uuid/2c65be95-ef46-49c3-a504-311ca6f9c8b6 /usr ext4 defaults 0 1
# /var was on /dev/sda4 during curtin installation
/dev/disk/by-uuid/4f9cb4f5-cf8c-4598-83ee-b56381f6c636 /var ext4 defaults 0 1
# /tmp was on /dev/sda5 during curtin installation
/dev/disk/by-uuid/67c8089c-7398-41ff-bc1d-8ed1e4d09fee /tmp ext4 defaults 0 1
~
~
~
```

3. Remontamos la partición con `mount -o remount /home` y reiniciamos los servicios demonios con `sudo systemctl daemon-reload`.

4. Creamos el fichero con el índice de cuota con `quotacheck -cum /home` y lo activamos con `quotaon -v /home`.

```
dansana@dansana:~$ sudo quotacheck -cum /home
dansana@dansana:~$ sudo quotaon -v /home
quotaon: Your kernel probably supports ext4 quota feature but you are using external quota files. Please switch your filesystem to use ext4 quota feature as ext
ernal quota files on ext4 are deprecated.
quotaon: cannot find /home/aquota.group on /dev/sda7 [/home]
/dev/sda7 [/home]: user quotas turned on
```

   Aparece un mensaje de aviso sobre el tipo de cuotas que puede ser ignorado ya que funciona sin problemas.

5. Revisamos la configuración de las cuotas de cada usuario con `edquota`, en el que aparece las columnas `soft limit` o límite blando que se puede sobrepasar temporalmente, y `hard limit` o límite duro que indica que no se puede sobrepasar ese límite:

```
  GNU nano 7.2                                                  /tmp//EdP.aSKdMmx
Disk quotas for user dansana (uid 1000):
  Filesystem                blocks       soft       hard     inodes     soft     hard
  /dev/sda7                     88          0          0         24        0        0
```

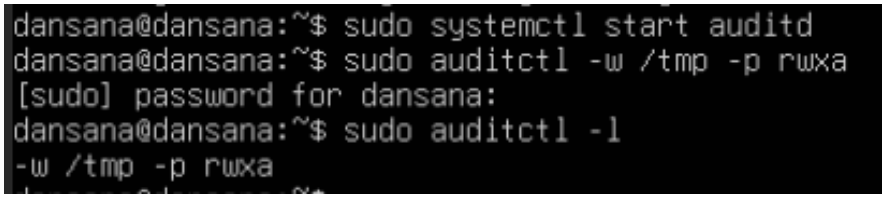
Figura 1: Cuota dansana

Un ejemplo donde se puede ver perfectamente su funcionamiento es modificando uno de los ficheros anteriores de los usuarios y estableciendo un límite de cuota tanto duro como blando, y ocupando el espacio disponible hasta ver el mensaje de `Disk quota exceeded`. Para el límite blando, se nos otorga un periodo de 7 días hasta bloquear la escritura en disco.

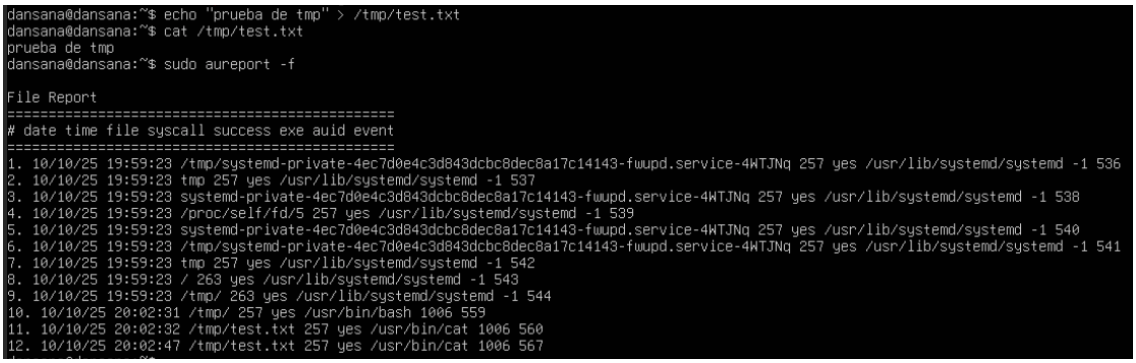
2.9 Autoría /tmp

La auditoría no proporciona seguridad adicional a su sistema; más bien, puede utilizarse para descubrir violaciones de las políticas de seguridad utilizadas en su sistema. Por ello, para realizar este tipo de operaciones, la herramienta `audit` que nos muestra esta información acerca de cualquier evento:

- Instalamos el servicio con `sudo apt install auditd` y lo arrancamos tal y como hemos visto en el apartado de `Servicios del sistema`.
- Después, creamos una nueva regla de auditoría para ese directorio con el comando `sudo auditctl -w /tmp -p rwxa`, donde se indica la carpeta a auditar y las operaciones que se auditan. Comprobamos que se ha agregado correctamente con `sudo auditctl -l`.



- Por último, realizaremos una prueba para ver cuál es contenido del reporte tras crear un fichero en ese directorio y ver su contenido:



  El resultado se puede ver en la última línea, que aparece repetida debido a una comprobación anterior del reporte.

2.10 Copia de seguridad

2.10.1 dump

Consultando el manual, esta herramienta trabaja con sistemas de ficheros `ext2, ext3, ext4` examinado su contenido y determinando qué archivos son necesarios hacer `backup`. Se pueden almacenar tanto en el propio disco como en un medio externo. Después, para recuperar el `backup`, la herramienta que se debe utilizar es `restore` pasando como argumento el fichero `backup`.

Existen diferentes opciones a tener en cuenta para realizar la copia de seguridad:

- **-level**: Nivel de la copia, siendo 0 un copia completo del sistema de ficheros.
- **-f**: El fichero donde queremos tener la copia de seguridad.
- **-u**: Actualiza el historial de copias `/etc/dumpdates`.
- **-z**: Activación del modo compresión de la copia.

```
dansana@dansana:~$ sudo dump -0uz -f /mnt/backups/home.dump /home
  DUMP: Date of this level 0 dump: Sat Oct 11 15:49:45 2025
  DUMP: Dumping /dev/sda7 (/home) to /mnt/backups/home.dump
  DUMP: Label: none
  DUMP: Writing 10 Kilobyte records
  DUMP: Compressing output at transformation level 2 (zlib)
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 1051319 blocks.
  DUMP: Volume 1 started with block 1 at: Sat Oct 11 15:49:45 2025
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: Closing /mnt/backups/home.dump
  DUMP: Volume 1 completed at: Sat Oct 11 15:49:48 2025
  DUMP: Volume 1 took 0:00:03
  DUMP: Volume 1 transfer rate: 2594 kB/s
  DUMP: Volume 1 1051270kB uncompressed, 7782kB compressed, 135.090:1
  DUMP: 1051270 blocks (1026.63MB) on 1 volume(s)
  DUMP: finished in 3 seconds, throughput 350423 kBytes/sec
  DUMP: Date of this level 0 dump: Sat Oct 11 15:49:45 2025
  DUMP: Date this dump completed:  Sat Oct 11 15:49:48 2025
  DUMP: Average transfer rate: 2594 kB/s
  DUMP: Wrote 1051270kB uncompressed, 7782kB compressed, 135.090:1
  DUMP: DUMP IS DONE
```

2.10.2 tar

Conocida herramienta para empaquetar y comprimir archivos o directorios en un fichero. Añadir que con el mismo comando podemos también desempaquetar y restaurar los archivos en los discos. Con `tar` podemos preservar los metadatos de los archivos originales, permisos, propiedad o enlaces simbólicos. Esto es muy importante a la hora de realizar copias de seguridad o permitir restaurar los contenidos respetando sus propiedades originales.

Las `flags` más utilizadas son las siguientes:

- **-c**: Indica la creación de un archivo.
- **-v**: Muestra todo lo que se está empaquetando.
- **-f**: Especifica el nombre del archivo comprimido.
- **-x**: Indica que se quiere extraer.

```
dansana@dansana:~$ sudo tar -cvf /mnt/backups/home.tar /home
```

2.10.3 rsync

Es una herramienta de sincronización de archivos local y remota, de forma que podemos transferir ficheros de manera eficiente. Utiliza el algoritmo de transferencia delta que minimiza la transferencia de datos al copiar solo las secciones de un archivo que se han actualizado. Admite la copia de enlaces, dispositivos, propietarios, grupos y permisos.

Los parámetros más comunes son:

- **-t**: Indicamos que es de forma recursiva.
- **-a**: Para copiar recursivamente en un directorio, con el añadido de mantener privilegios, permisos y fechas de los ficheros y directorios.
- **-z**: Copiado de forma comprimida.
- **-t**: Preservamos el tiempo de modificación de los archivos.

```
dansana@dansana:~$ sudo rsync -a /home/ /mnt/backups/home_backup/
```

3. Autenticación. Seguridad. Control de Servicios.

3.1 PAM

En Linux, la autenticación de los usuarios en el sistema está estandarizado mediante el uso de `PAM`. Proporciona un mecanismo para añadir autenticación a los programas mediante el uso de llamadas a las bibliotecas de funciones `PAM`.
El uso de diferentes módulos para cada servicio facilita que cada uno de ellos no tenga que implementar el mecanismo de acceso, sino simplemente pasar las credenciales a `PAM` y este se encargue de indicar si el usuario tiene acceso a no.

3.1.1 Claves fuertes

El comando `passwd` es uno de los que sí usan `PAM` a la hora de crear las contraseñas, ya que en vez comunicarse con los ficheros `/etc/passwd` y `/etc/shadow`, hace una llamada al sistema `PAM` y recibe las instrucciones del fichero `/etc/pam.d/password`, donde realmente tiene su configuración en `/etc/pam.d/common-password`. Comprendido esto, modificaremos este fichero para obligar a los nuevos usuarios a que la nueva contraseña que asignen tenga estos criterios:

- Permitir que, en caso de fallo, sólo reintente 2 veces.
- Obligatoriedad de contener al menos 1 carácter numérico, 1 carácter mayúscula y 1 carácter minúscula.
- Que no se utilicen palabras claves ya usadas, o que estén en un diccionario.

Antes de comenzar, **IMPORTANTE** hacer copia del fichero `/etc/pam.d/password` porque si se comenten errores se puede bloquear los cambios de clave y habrá que formatear la máquina. También, es conveniente tener una sesión `root` abierta en caso de problemas.

Este sería el fichero `/etc/pam.d/common-password` actualizado para conseguir cumplir esos criterios:

```
#
# /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=2 minlen=12 ucredit=-1 lcredit=-1 dcredit=-1
password        required                        pam_pwhistory.so remember=5 use_authtok
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
~
~
```

- En el primer módulo, `libpam-pwquality`, indicamos el número de reintentos que tiene el usuario al cambiar la clave y cómo tiene que ser esa clave.
- En el segundo módulo, `libpam-pwhistory`, conseguimos que se almacene las 5 últimas contraseñas.

NOTA: En el caso de perder o dañar dicho archivo y sí tener `backup`, habrá que reiniciar la máquina virtual e ir al CD de `Ubuntu Server` y desde el shell recuperar el fichero.

3.1.2 Control de acceso por hora y por terminal

Siguiendo con el servicio `PAM`, procedemos a cambiar el `login` de nuestra máquina para agregar las siguientes restricciones, a través del fichero `/etc/security/time.conf` que configura las reglas del módulo `pam_time` encargado de la autenticación de un usuario en el sistema:

- Un usuario cualquiera (por ejemplo `userA`) no pueda trabajar de 11:30h a 11:40h.
- Que ese mismo usuario no se pueda conectar a la consola.
- Y además, pueda acceder al resto de consolas que no sea la `tty1`.

En el fichero, cada línea tiene que tener esta estructura: `services;ttys;users;times`.

IMPORTANTE: Al igual que antes, hay que hacer copia de los ficheros que vayamos a cambiar en caso de fallo.

```
# Se bloquea el acceso al sistema del usuario A en el horario de
# 11:30 a 11:40 de la mañana
login;*;userA;!Al1130-1140
sshd;*;userA;!Al1130-1140

# No tiene permitido el usuario A acceder a la consola tty1
login;tty1;userA;!
```

3.2 Crontabd y atd

3.2.1 Crontabd

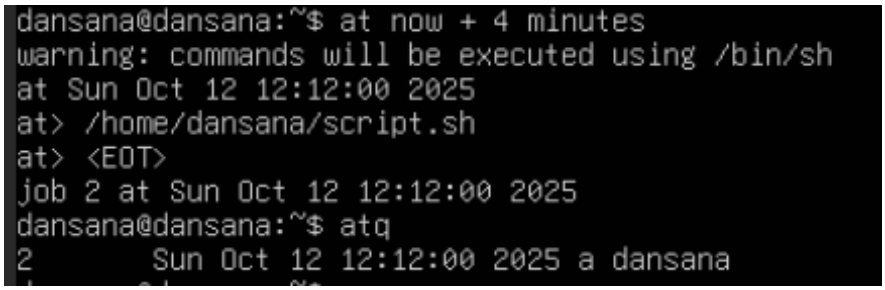
Se trata de un servicio demonio que permite programar tareas automáticas para que se ejecuten de forma periódica. Para asignar un nueva tarea, lanzamos el comando `crontab -e` donde indicamos primero cada cuanto se va a lanzar el script.

```
dansana@dansana:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

* * * * * /home/dansana/script.sh
```

3.2.2 atd

La orden `at` ejecuta un programa en un momento específico en el futuro. Toma el tiempo y la fecha deseados como parámetros de línea de comandos, y el comando a ejecutar en su entrada estándar. Ejecutará el programa como si hubiese ingresado en la consola.



3.3 Seguimiento de la ejecución de servicios

Para ver los resultados el servicio demonio `crond` en el registro `rsyslogd`, hay que activar el registro y editar su configuración añadiendo la regla de que los `logs` del servicio aparezcan en el registro. La línea que hay que añadir es `cron.* /var/log/cron.log` y reiniciar tanto el servicio `cron` y el `rsyslog`.

Referencias

[1] ArchWiki. Disk quota. https://wiki.archlinux.org/title/Disk_quota, 2022.

[2] Joel Barrios Dueñas. Permisos del sistema de archivos en gnu/linux. https://blog.alcancelibre.org/staticpages/index.php/permisos-sistema-de-archivos, 2016.

[3] Unix StackExchange. Where is the rc file for vi editor. https://unix.stackexchange.com/questions/180710/where-is-the-rc-file-for-vi-editor#:~:text=IEEE%20Std%201003.1%2D2001%20does,the%20root%20partition%20is%20present., 2022.

[4] Ubuntu. Guía de escritorio de kubuntu. https://help.ubuntu.com/kubuntu/desktopguide/es/directories-file-systems.html, 2006.

[5] Ubuntu. Guía de escritorio de kubuntu. https://help.ubuntu.com/kubuntu/desktopguide/es/apt-get.html, 2006.

[6] Wikipedia. Raid. https://es.wikipedia.org/wiki/RAID#, 2011.

[7] Wikipedia. Cuota de disco. https://es.wikipedia.org/wiki/Cuota_de_disco, 2024.

[8] Wikipedia. Gnu/linux. https://es.wikipedia.org/wiki/GNU/Linux#Discos,_particiones_y_sistemas_de_archivos, 2025.