



Universidad de Valladolid

Cuaderno de bitácora

Ingeniería Informática
Tecnologías de la Información

Daniel Sanabria Salamanqués

6 de octubre de 2025

Índice

1. Instalación	2
1.1. Clúster de Máquinas Virtuales	2
1.2. Configuración de instalación	2
1.3. Reconocimiento del entorno	2
1.4. Acceso remoto vía ssh	3
1.5. Activar cuenta root	3
1.6. Administración del disco	4
1.6.1. Loop	5
1.6.2. LVM	6
1.7. Administración de almacenamiento	7
1.7.1. RAID 5	7
1.8. Nueva instalación personalizada	8
1.9. Trabajo No Presencial	9

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 **snapp** 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.
- **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.
- **Reiniciar la máquina:** Para el reinicio, se emplea el comando `reboot`.
- **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:

```
ssh dansana@jair.lab.inf.uva.es
(dansana@jair.lab.inf.uva.es) Password:
dansana@jair:~ (0.05s)

  _____
 /  _  _  _  \
/_  /  _  _  \
 \  \  _  _  /
  \  \  _  _ /
   \  \  _  /
    \  \  _/
     \  \_
      \___

Servidor de practicas de alumnos

Hello! This is jair.lab.inf.uva.es at 157.88.125.192

dansana@jair ~
ssh -p 38031 virtual.lab.inf.uva.es
dansana@virtual.lab.inf.uva.es's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-79-generic x86_64)

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

System information as of Thu Sep 18 10:20:56 AM UTC 2025

System load:  0.0               Processes:            145
Usage of /:   11.2% of 57.72GB   Users logged in:     1
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

Expanded Security Maintenance for Applications is not enabled.

149 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Mon Sep 15 16:01:45 2025 from 157.88.125.192
dansana@vm3803:~$
```

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.

```
dansana@vm3803:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
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
```

- **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.

```
dansana@vm3803:~$ cat /etc/fstab
# /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>
# / was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/6de6d247-712b-4f7c-addb-2e028b04acee / ext4 defaults 0 1
# /boot/efi was on /dev/sda1 during curtin installation
/dev/disk/by-uuid/9996-869C /boot/efi vfat defaults 0 1
/swap.img none swap sw 0 0
```

- **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
nodev    ext3
nodev    ext2
nodev    ext4
nodev    squashfs
nodev    vfat
nodev    ecryptfs
nodev    fuseblk
nodev    fuse
nodev    fusectl
nodev    efivarfs
nodev    mqueue
nodev    pstore
nodev    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 ñas 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                 au1Q83-5hWC-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/efivars
/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:

```
dansana@vm3803:~$ sudo dd if=/dev/zero of=fich1 bs=1 count=4096
[sudo] password for dansana:
4096+0 records in
4096+0 records out
4096 bytes (4.1 kB, 4.0 KiB) copied, 0.0066625 s, 615 kB/s
dansana@vm3803:~$ sudo dd if=/dev/zero of=fich2 bs=1 count=512
512+0 records in
512+0 records out
512 bytes copied, 0.000625003 s, 819 kB/s
dansana@vm3803:~$ sudo dd if=/dev/zero of=fich3 bs=1 count=512
512+0 records in
512+0 records out
512 bytes copied, 0.00153401 s, 334 kB/s
dansana@vm3803:~$ sudo dd if=/dev/zero of=fich3 bs=1 count=512
512+0 records in
512+0 records out
512 bytes copied, 0.00118201 s, 433 kB/s
dansana@vm3803:~$ ls
fich1  fich2  fich3  fichero
dansana@vm3803:~$ ls -l
total 103148
-rw-r--r-- 1 root root 512 Oct 5 17:31 fich1
-rw-r--r-- 1 root root 512 Oct 5 17:31 fich2
-rw-r--r-- 1 root root 512 Oct 5 17:31 fich3
-rw-rw-r-- 1 dansana dansana 4294967296 Oct 5 16:26 fichero
```


Y después asociarlo a 3 dispositivos loop:

```
dansana@vm3803:~$ sudo losetup /dev/loop1 fich1
dansana@vm3803:~$ sudo losetup /dev/loop2 fich2
dansana@vm3803:~$ sudo losetup /dev/loop3 fich3
dansana@vm3803:~$ sudo losetup -a
/dev/loop1: [2050]:1966094 (/home/dansana/fich1)
/dev/loop2: [2050]:1966095 (/home/dansana/fich2)
/dev/loop0: [2050]:1966092 (/home/dansana/fichero)
/dev/loop3: [2050]:1966096 (/home/dansana/fich3)
```

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.

```
dansana@vm3803:~$ sudo mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/loop1 /dev/loop2 /dev/loop3
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
dansana@vm3803:~$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid5 loop3[3] loop2[1] loop1[0]
      1044480 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
```

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

```
dansana@vm3803:~$ sudo mkfs.ext4 /dev/md0
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 261120 4k blocks and 65280 inodes
Filesystem UUID: 8a34336f-0feb-43d5-aa1e-4040d30a2160
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

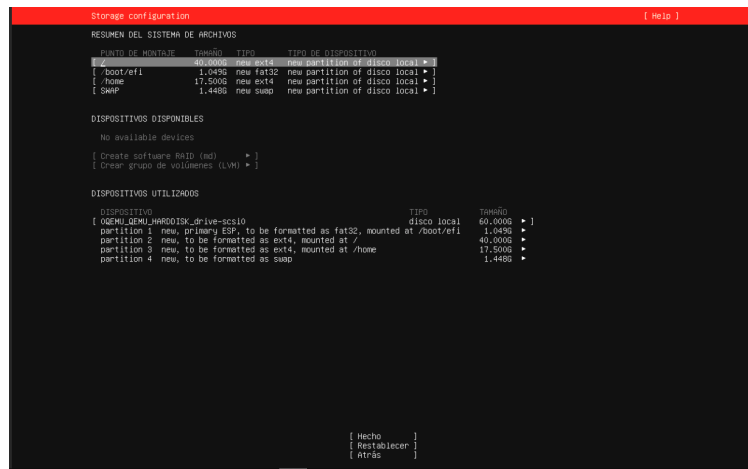
dansana@vm3803:~$ sudo mkdir /mnt/raid5
dansana@vm3803:~$ sudo mount /dev/md0 /mnt/raid5
dansana@vm3803:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           534M  1.2M  533M   1% /run
efivarfs        56K   40K   12K  78% /sys/firmware/efi/efivars
/dev/sda2       58G   8.6G   47G  16% /
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           594M   8.0K  594M   1% /run/user/1000
/dev/md0        986M   24K  919M   1% /mnt/raid5
```

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 fichero no es el disco como tal, por lo que no pasa nada.

1.8 Nueva instalación personalizada

Hasta ahora hemos trabajado 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 **40G** para que no haya problemas a la hora de agregar elementos al sistema.
- **swap**: Con un espacio de **1.448G**, solo actúa en caso de que la memoria RAM se quede sin espacio.
- **/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*.^[4]
 - Una partición sirven organizan y almacenan el sistema operativo, las aplicaciones y los archivos personales. Existen diferentes esquemas de particiones 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:**
 - 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.^[1]
 - 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.
- **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.^[2]
 - 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.).^[4]
 - 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. ^[3]
- 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.

Referencias

- [1] Ubuntu. Guía de escritorio de kubuntu. <https://help.ubuntu.com/kubuntu/desktopguide/es/directories-file-systems.html>, 2006.
- [2] Ubuntu. Guía de escritorio de kubuntu. <https://help.ubuntu.com/kubuntu/desktopguide/es/apt-get.html>, 2006.
- [3] Wikipedia. Raid. <https://es.wikipedia.org/wiki/RAID#>, 2011.
- [4] Wikipedia. Gnu/linux. https://es.wikipedia.org/wiki/GNU/Linux#Discos,_particiones_y_sistemas_de_archivos, 2025.