

# Configuración de Boot para servidores

<b>Antes de empezar</b>	<b>1</b>
¿Qué realizaremos como ejemplo práctico?	1
<b>Configuración inicial y esquema</b>	<b>2</b>
Resumen de los Raids:	4
<b>Creación del RAID1</b>	<b>5</b>
<b>Encriptamiento y montaje</b>	<b>7</b>
Encriptamiento	7
Montaje	8
Configuración de montaje automático	10
Desencriptado automático:	11
<b>¿Montajes adicionales?</b>	<b>12</b>

## Antes de empezar

En este documento se resumirán mediante un ejemplo práctico, las bases de configuración del boot de un computador, que normalmente se aplican a servidores linux.

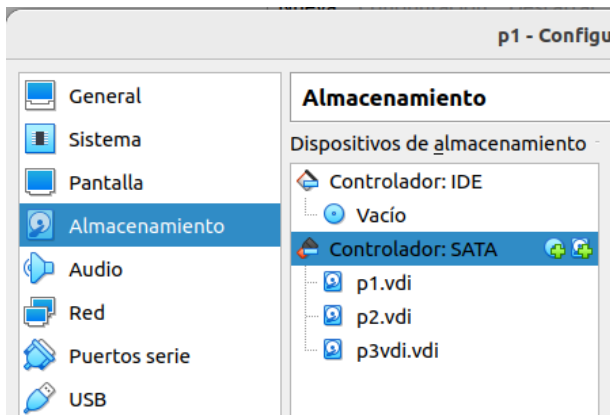
No utilizaremos un dispositivo real para ello, si no un hipervisor (VirtualBox), pero todo lo explicado aquí es extrapolable a una situación real sustituyendo los pasos realizados en el hipervisor por la correspondiente configuración en los equipos físicos.

Esta información proviene de lo aprendido en las prácticas de la asignatura Ingeniería de Servidores, impartida en el grado en Ingeniería informática en la Universidad de Granada.

## ¿Qué realizaremos como ejemplo práctico?

Vamos a crear un raid1, mediante dos dispositivos adicionales al que ya teníamos por defecto, con el objetivo de montar la carpeta /var en este raid, asegurando así el contenido de esta carpeta que será la que nuestro servidor use para su supuesta misión. Además lo encriptaremos para darle un extra de seguridad.

## Configuración inicial y esquema



En este caso utilizaremos el sistema operativo **Rocky Linux**. Puedes utilizar cualquier otro sistema operativo de distribución linux teniendo en cuenta los posibles pequeños cambios que pueda haber entre distribuciones.

Comenzaremos con la instalación del sistema operativo ya hecha por defecto, para proceder a insertar los discos duros mediante el hipervisor. En esta ocasión hemos añadido dos más (P2 y P3). Esto se corresponde con insertar los discos de forma física en una instalación real.

Procederemos a:

1. Crear un usuario.
2. Asignarle una contraseña.
3. Añadir al usuario al grupo wheel para tener acceso al SU.

Todo esto lo haremos con el fin de no tener que operar desde el usuario root.

```
[root@localhost ~]# useradd albertopm
[root@localhost ~]# passwd albertopm
Changing password for user albertopm.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]# usermod -aG wheel albertopm
[root@localhost ~]# _
```

A continuación nos añadiremos al grupo de usuarios que pueden usar sudo:

accederemos al archivo que contiene la lista de usuarios que pueden usar sudo con:

**visudo /etc/sudoers**

y bajaremos hasta casi el final del archivo, hasta encontrar una línea como esta:

“root ALL=(ALL) ALL”. debajo de esa línea añadimos una exactamente igual, pero cambiando root por el nombre de usuario que nos hayamos puesto. El archivo quedaría tal que así:

```
## Allow root to run any commands anywhere
root    ALL=(ALL)        ALL
albertopm    ALL=(ALL)        ALL

##
## Defaults    env_keep += "HOME"
Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##     user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)        ALL
albertopm    ALL=(ALL)        ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)        ALL

## Same thing without a password
# %wheel    ALL=(ALL)        NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users    ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users    localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
```

**Recordatorio sobre el editor de texto vi:** si no sabes como funciona el editor vi, puedes moverte por el archivo con las flechas, para comenzar a escribir pulsa la tecla **i**, para salir del modo escritura **esc**, y para salir y guardar los cambios, en la pequeña terminal de una línea que hay abajo del todo (siempre está activa, solo tienes que comenzar a escribir mientras no estés en el modo insertar) escribe **:wq** (write and quit).

Una vez hayamos hecho esto, podemos pasar a nuestro usuario con el comando **su <tuusuario>**. Y después haremos un **cd** para salir de la carpeta root y evitar así posibles despistes en los direccionamientos.

Ahora instalaremos las herramientas que utilizaremos más adelante:

**sudo dnf update**

**sudo dnf install mdadm** (creación de raids)

**sudo dnf install cryptsetup** (encriptamiento)

Respecto al esquema, tenemos que decidir qué sistema de raid utilizaremos. Esta decisión tenemos que tomarla nosotros en función de las necesidades y limitaciones que tenga nuestro servidor.

## Resumen de los Raids:

- **Raid 0:** mínimo 2 discos. Reparte datos entre los discos, lo que implica mayor rendimiento pero menor seguridad. Sería óptimo si quisiéramos velocidad y no nos importase el riesgo de perder la información (ya que si se rompe uno de los discos, como la información está compartida entre los dos, esta se perderá por completo).
- **Raid 1:** mínimo 2 discos. Hace un efecto “espejo”, duplicando los datos. Aporta menos rendimiento que el raid 0, ya que tiene que copiar los datos en los dos discos, pero aporta más seguridad, ya que la rotura de uno de los discos no implicaría la pérdida de la información.
- **Raid 5:** mínimo 3 discos. Usa paridad para recuperar archivos. Esto le dota de mayor rendimiento y seguridad que sus dos anteriores versiones, pero implica más costo ya que necesitas un disco más. Además la reconstrucción de la información mediante paridad es larga y costosa.

Hay más dispositivos raid, pero estos son los principales. Si quieres más información sobre los raid puedes consultarla [aquí](#).

En nuestro ejemplo queremos un servidor cuya información esté siempre disponible (lo que descarta automáticamente el raid 0, ya que, a diferencia del 1 y el 5, la rotura de un disco inhabilitaría la información). Por tanto lo ideal sería un Raid 5, ya que es el que nos aporta más seguridad y rendimiento, sin embargo sólo disponemos de dos discos (como vimos en la configuración anteriormente, añadimos los discos p2 y p3, el p1 no cuenta, ya que es el de la configuración inicial). Como nuestro presupuesto no nos permite añadir un tercer disco p4, realizaremos un raid 1 para ajustarnos a las exigencias del servidor.

Si hacemos un lsblk podemos consultar la estructura actual de nuestro servidor:

```
[albertopm@localhost ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   8G  0 disk
├─sda1       8:1    0   1G  0 part /boot
└─sda2       8:2    0   7G  0 part
   └─rl-root 253:0    0  6.2G  0 lvm  /
      └─rl-swap 253:1    0  820M  0 lvm  [SWAP]
sdb          8:16    0   8G  0 disk
sdc          8:32    0   8G  0 disk
sr0         11:0    1 1024M  0 rom
```

Analicemos un poco la estructura:

Respecto a la configuración inicial, podemos ver lo que hace nuestro sistema operativo por defecto. El disco sda (correspondiente al disco inicial, P1) lo parte en dos (no de igual tamaño), sda1 en el que hace el montaje de /boot, para poder arrancar el computador, y sda2, en el que configura root (la carpeta raíz /) y la partición del swap.

Respecto a sdb y sdc (los discos que nosotros hemos añadido) podemos ver cómo están conectados al computador, pero no tienen ningún tipo de asignación hecha.

## Creación del RAID1

Con la herramienta mdadm que instalamos antes, realizaremos la instalación del Raid que queremos.

```
sudo mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb /dev/sdc
```

/dev/md0 es el nombre del raid. Es convención llamarlos md<num de raid>

--level=<num> es el tipo de raid que se va a configurar

--raid-devices=<num> es el número de dispositivos que va a tener el raid, a continuación se añaden los discos que se van a añadir, en nuestro caso sdb y sdc

Todo va precedido de /dev/... porque este es el directorio que utiliza linux para referirse a los dispositivos de hardware.

Te saldrá un aviso sobre /boot que tendrás que confirmar, y comenzará la sincronización. te dejará usar la terminal mientras la realiza, hasta que te salga un mensaje que confirme la finalización.

```
[albertopm@localhost ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   8G  0 disk
├─sda1       8:1    0   1G  0 part /boot
└─sda2       8:2    0   7G  0 part
   └─rl-root 253:0    0  6.2G  0 lvm  /
     └─rl-swap 253:1    0  820M  0 lvm  [SWAP]
sdb          8:16    0   8G  0 disk
└─md0        9:0    0   8G  0 raid1
sdc          8:32    0   8G  0 disk
└─md0        9:0    0   8G  0 raid1
sr0         11:0    1 1024M  0 rom
[albertopm@localhost ~]$
```

En el lsblk ahora podemos ver como tanto sdb como sdc conforman el dispositivo raid md0.

Sin embargo, linux aún no los ha asignado como dispositivos ni físicos ni lógicos, lo cual podemos comprobar con los comandos **sudo pvs** (para los físicos) y **sudo lvs** (para los lógicos)

```
albertopm@localhost ~]$ sudo pvs
PV          VG Fmt Attr PSize PFree
/dev/sda2   rl  lvm2 a--  <7.00g  0
albertopm@localhost ~]$ sudo lvs
LV          VG Attr      LSize   Pool Origin Data%  Meta%   Move Log Cpy%Sync Convert
root        rl  -wi-ao----  <6.20g
swap        rl  -wi-ao----  820.00m
```

De modo que realizaremos las asignaciones correspondientes:

1. Creación del disco duro físico:

```
sudo pvcreate /dev/md0
```

2. Creación del grupo de volúmenes (LVM):

```
sudo vgcreate vgraid1 /dev/md0
```

vgraid1 es el nombre, le asigno este para dejar claro que se trata de un volumen group de raid 1.

3. Creación del volumen lógico:

```
sudo lvcreate -L 1G -n var vgraid1
```

En este caso, le asigno 1 gigabyte al disco es un número arbitrario, tú le asignas en función de las necesidades de tu servidor. También le llamo var al volumen lógico, simplemente para designar que será el que montará la carpeta /var más adelante.

```
albertopm@localhost ~]$ sudo pvs
PV          VG Fmt Attr PSize PFree
/dev/md0     vgraid1 lvm2 a--  <7.99g <6.99g
/dev/sda2    rl  lvm2 a--  <7.00g  0
albertopm@localhost ~]$ sudo lvs
LV          VG Attr      LSize   Pool Origin Data%  Meta%   Move Log Cpy%Sync Convert
root        rl  -wi-ao----  <6.20g
swap        rl  -wi-ao----  820.00m
var          vgraid1 -wi-a-----  1.00g
albertopm@localhost ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   8G  0 disk
├─sda1       8:1    0   1G  0 part /boot
├─sda2       8:2    0   7G  0 part
│ └─rl-root  253:0    0  6.2G  0 lvm  /
│ └─rl-swap  253:1    0  820M  0 lvm  [SWAP]
sdb          8:16   0   8G  0 disk
└─md0        9:0    0   8G  0 raid1
   └─vgraid1-var 253:2    0   1G  0 lvm
sdc          8:32   0   8G  0 disk
└─md0        9:0    0   8G  0 raid1
   └─vgraid1-var 253:2    0   1G  0 lvm
sr0         11:0    1 1024M  0 rom
```

Ahora están las asignaciones correctas. En pvs podemos ver el disco físico md0, en lvs el vgraid1 como var, en estos dos casos abstrayéndose por completo de que sea en realidad en dos discos diferentes sdb y sdc gracias a que es un raid. Finalmente en lsblk podemos verlo, ya si sabiendo que en realidad es en dos discos diferentes, con la nomenclatura columengroup-logicalvolume (vgraid1-var).

## Encriptamiento y montaje

### Encriptamiento

Si en tu raid no vas a llevar a cabo un encriptamiento, puedes realizar todos estos pasos saltándote todos los correspondientes al encriptamiento y sustituyendo los nombres de raid encriptados por los originales.

Puedes utilizar cualquier otra herramienta alternativa a cryptsetup.

Para encriptar nuestro raid:

**sudo cryptsetup luksFormat /dev/vgraid1/var**

Nos pedirá confirmación, y asignarle contraseña al dispositivo raid. Este ahora estará bloqueado, para operar con él, habrá que abrirlo mediante cryptsetup y operar con el dispositivo abierto. Para ello:

**sudo cyptsetup luksOpen /dev/vgraid1/var vgraid1-var\_crypt**

vgraid1-var\_crypt es el nombre, utilizo este formato por convención y comodidad más adelante, pero puedes utilizar el que quieras.

En la siguiente captura podemos observar el antes y el después del lsblk tras “abrir” el raid encriptado, apareciendo un nuevo dispositivo “vgraid1-var\_crypt”. Este dispositivo es el que nos permite operar con él pues ya hemos confirmado nuestra identidad superando el encriptamiento.

```
albertopm@localhost ~]$ lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                  8:0      0   8G  0 disk
├─sda1               8:1      0   1G  0 part /boot
├─sda2               8:2      0   7G  0 part
└─rl-root            253:0     0  6.2G  0 lvm /
   └─rl-swap          253:1     0  820M  0 lvm [SWAP]
sdb                  8:16     0   8G  0 disk
├─md0                9:0      0   8G  0 raid1
└─vgraid1-var        253:2     0   1G  0 lvm
sdc                  8:32     0   8G  0 disk
├─md0                9:0      0   8G  0 raid1
└─vgraid1-var        253:2     0   1G  0 lvm
sr0                  11:0     1 1024M  0 rom

albertopm@localhost ~]$ sudo cryptsetup luksOpen /dev/vgraid1/var vgraid1-var_crypt
Enter passphrase for /dev/vgraid1/var:
albertopm@localhost ~]$ lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                  8:0      0   8G  0 disk
├─sda1               8:1      0   1G  0 part /boot
├─sda2               8:2      0   7G  0 part
└─rl-root            253:0     0  6.2G  0 lvm /
   └─rl-swap          253:1     0  820M  0 lvm [SWAP]
sdb                  8:16     0   8G  0 disk
├─md0                9:0      0   8G  0 raid1
└─vgraid1-var        253:2     0   1G  0 lvm
   └─vgraid1-var_crypt 253:3     0 1008M  0 crypt
sdc                  8:32     0   8G  0 disk
├─md0                9:0      0   8G  0 raid1
└─vgraid1-var        253:2     0   1G  0 lvm
   └─vgraid1-var_crypt 253:3     0 1008M  0 crypt
sr0                  11:0     1 1024M  0 rom
```

## Montaje

**Nota sobre /mnt:** La carpeta /mnt es una carpeta que utiliza linux como punto de montajes temporales.

**nota sobre /dev/mapper/:** El mapper es un marco proporcionado por el kernel de linux para mapear dispositivos. Los dispositivos descriptados se encuentran ahí (a no se que nosotros hagamos lo contrario, en nuestro caso, los dejaremos ahí).

1. Damos formato al disco:

```
sudo mkfs -t ext4 /dev/mapper/vgraid1-var_crypt
```

ext4 simplemente porque es lo habitual para el tipo de archivos que se almacenarán en /var.

2. Creamos una carpeta en /mnt para usarla como punto de montaje temporal.

```
sudo mkdir /mnt/new_var
```

3. Montamos el dispositivo descriptado en la carpeta temporal:



```
sudo mount /dev/mapper/vgraid1-var_crypt /mnt/new_var
```

4. Hacemos la copia de /var

```
sudo mv /var /var_old
```

 (simplemente le cambio el nombre a /var)

```
sudo mkdir /var
```

 (creo un nuevo /var)

```
sudo restorecon /var
```

 (le restauro el contexto a la nueva /var)

copio el contenido del antiguo /var al punto de montaje temporal que hicimos antes, es decir, al raid, NO a la nueva carpeta /var que hemos creado:

```
sudo cp -a /var_old/* /mnt/new_var/
```

Es decir ahora mismo tendríamos:

- Una carpeta /var\_old con el contenido de la antigua /var
- Una carpeta /var con su contexto restaurado y que está vacía
- Una carpeta /mnt/new\_var que hemos usado como contenedor temporal, montada sobre el raid1 que hemos creado y que tiene el contenido de la antigua /var.

Con un lsblk podemos ver ese montaje en /mnt/new\_var

```
[albertopm@localhost new_var]$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                                 8:0      0   8G  0 disk
├─sda1                             8:1      0    1G  0 part /boot
├─sda2                             8:2      0    7G  0 part
│   ├─rl-root                       253:0      0   6.2G  0 lvm /
│   └─rl-swap                       253:1      0   820M  0 lvm [SWAP]
sdb                                 8:16     0    8G  0 disk
├─md0                              9:0      0    8G  0 raid1
│   └─vgraid1-var                   253:2      0    1G  0 lvm
│       └─vgraid1-var_crypt         253:3      0  1008M  0 crypt /mnt/new_var
sdc                                 8:32     0    8G  0 disk
├─md0                              9:0      0    8G  0 raid1
│   └─vgraid1-var                   253:2      0    1G  0 lvm
│       └─vgraid1-var_crypt         253:3      0  1008M  0 crypt /mnt/new_var
sr0                                11:0     1  1024M  0 rom
[albertopm@localhost new_var]$
```

y con diferentes ls, podemos ver como /var está vacía, y /var\_old y /mnt/new\_var tienen el contenido de la antigua var y la nueva var respectivamente. El contenido ahora mismo es el mismo, pero podemos crear una carpeta "test" en el contenido del nuevo var (montado aún de forma temporal), es decir, en /mnt/new\_var, para diferenciarlos mejor.

```
albertopm@localhost ~]$ ls /var
albertopm@localhost ~]$ ls /var_old
adm  crash  empty  games  lib  lock  mail  opt  run  tmp
cache db  ftp  kerberos  local  log  nis  preserve  spool  yp
albertopm@localhost ~]$ ls /mnt/new_var
TEST cache db  ftp  kerberos  local  log  mail  opt  run  tmp
adm  crash  empty  games  lib  lock  lost+found  nis  preserve  spool  yp
albertopm@localhost ~]$
```

Yo lo he hecho con la carpeta “TEST” (primera de todas), pero no es necesario, es simplemente para que se vea claro en este documento lo que está sucediendo.

5. Desmontamos el raid1 de la carpeta temporal:

```
sudo umount /mnt/new_var
```

6. Montamos el dispositivo raid definitivamente sobre la carpeta a la que estaba destinado, /var:

```
sudo mount /dev/mapper/vgraid1-var_crypt /var
```

new\_var, estaba montada sobre el raid1, y en esta carpeta habíamos copiado el contenido de var\_old y habíamos creado una carpeta “TEST”. Este contenido estaba, a fin de cuentas, almacenado en el raid1, por lo que al haber montado el raid ahora en /var, este contenido debería aparecer inmediatamente en /var.

```
albertopm@localhost ~]$ ls /var
albertopm@localhost ~]$ sudo mount /dev/mapper/vgraid1-var_crypt /var
[ 6255.766031] EXT4-fs (dm-3): mounted filesystem with ordered data mode. Quota mode: none.
albertopm@localhost ~]$ ls /var
TEST cache db  ftp  kerberos  local  log  mail  opt  run  tmp
adm  crash  empty  games  lib  lock  lost+found  nis  preserve  spool  yp
albertopm@localhost ~]$ _
```

## Configuración de montaje automático

Queremos que siempre que arranquemos nuestro servidor, la carpeta /var se monte automáticamente sobre el raid1 que hemos creado, para hacer esto, tenemos que editar el fichero /etc/fstab.

```
sudo vi /etc/fstab
```

Y en este incluiremos nuestro dispositivo raid descryptado:

```
##
## /etc/fstab
## Created by anaconda on Sun Oct 30 10:58:59 2022
##
## Accessible filesystems, by reference, are maintained under '/dev/disk/'.
## See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
##
## After editing this file, run 'systemctl daemon-reload' to update systemd
## units generated from this file.
##
/dev/mapper/vgraid1--var_crypt /var ext4 defaults 0 0
/dev/mapper/rl-root / xfs defaults 0 0
UUID=55a2b514-dd9f-4577-bee0-9dd8cab99cfc /boot xfs defaults 0 0
/dev/mapper/rl-swap none swap defaults 0 0
```

Desencriptado automático:

Primero tendremos que ponernos en modo root:

**su root**

Ahora buscaremos la clave de criptografía en blkid utilizando un grep y la trasladaremos al fichero /etc/crypttab

```
blkid | grep "crypto" > /etc/crypttab
```

Dejamos el fichero tal que así:

```
vgraid1-var_crypt UUID=cd3446a8-b14a-4996-9d4f-19126916538      _none
```

Finalmente esta es la estructura que nos queda y que arrancará automáticamente cada vez que iniciemos el sistema:

```
[root@localhost ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sda                                8:0    0   8G  0 disk
├─sda1                            8:1    0   1G  0 part  /boot
├─sda2                            8:2    0   7G  0 part
│   └─rl-root                    253:0    0  6.2G  0 lvm    /
│       └─rl-swap                253:1    0  820M  0 lvm    [SWAP]
sdb                                8:16    0   8G  0 disk
├─md0                             9:0    0   8G  0 raid1
│   └─vgraid1-var                253:2    0   1G  0 lvm
│       └─vgraid1-var_crypt      253:3    0 1008M  0 crypt  /var
sdc                                8:32    0   8G  0 disk
├─md0                             9:0    0   8G  0 raid1
│   └─vgraid1-var                253:2    0   1G  0 lvm
│       └─vgraid1-var_crypt      253:3    0 1008M  0 crypt  /var
sr0                                11:0    1 1024M  0 rom
```

## ¿Montajes adicionales?

Si quisiéramos hacer montajes adicionales, por ejemplo hacer esto también con la carpeta /home sobre el mismo raid1 que hemos creado anteriormente, como tenemos espacio de sobra, bastaría con repetir el mismo proceso, creando un nuevo volumen físico, volume group, volumen lógico, etc...