

ISE Practica 1

ISE: oresti@ugr.es - D2.26- M, J 10h-11:30h (enviar email)

Ubuntu

Términos

Servidor: Sistema que recibe peticiones y proporciona servicios

- Dedicados: Problemas físicos, mejor rendimiento
- VPS (Virtual Private Server): Más baratos, añaden versatilidad

Máquina Virtual: Emula el comportamiento de una máquina real

Hipervisor: Software que permite crear una Máquina Virtual (ej: VirtualBox)

- Native: Montan sobre el hardware
- Hosted: Existe un SO operativo por debajo que controla (VirtualBox y Windows/Linux sobre el que se instala)

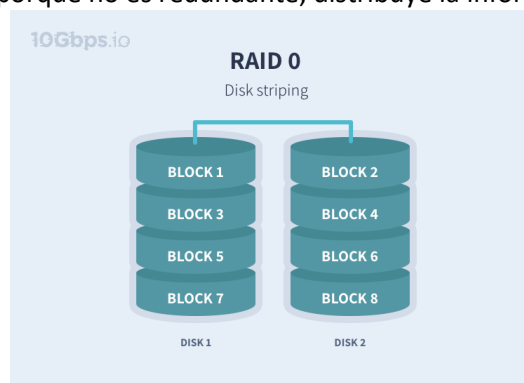
CentOS se utiliza para servidores de gama media/alta

Es importante configurar (especificar) el servidor al uso que se le va a dar

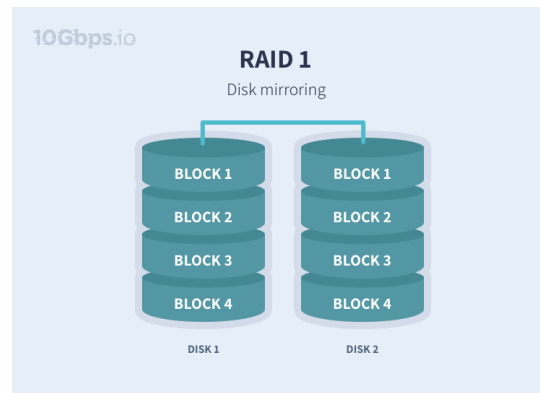
RAID (Redundant Array Inexpensive Disks): Estructura de almacenamiento de datos que combina múltiples discos en una o más unidades lógicas con el propósito de generar redundancia, aumentar rendimiento, o ambas cosas. Esta redundancia también nos permite la realización de copias de seguridad.

En vez de 1TB -> 10 de 100GB, lo que resulta en un acceso más rápido (pues permite paralelismo)

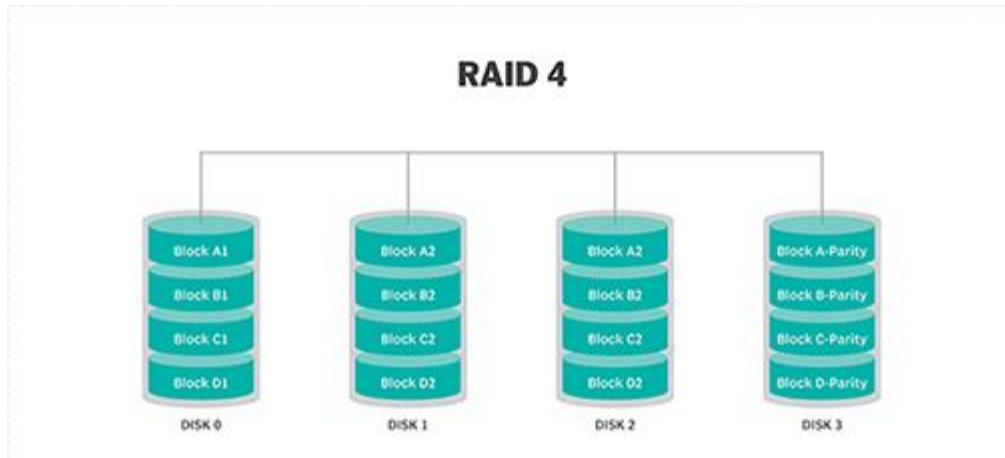
- Tipo 0: No es realmente un RAID porque no es redundante, distribuye la información en los discos



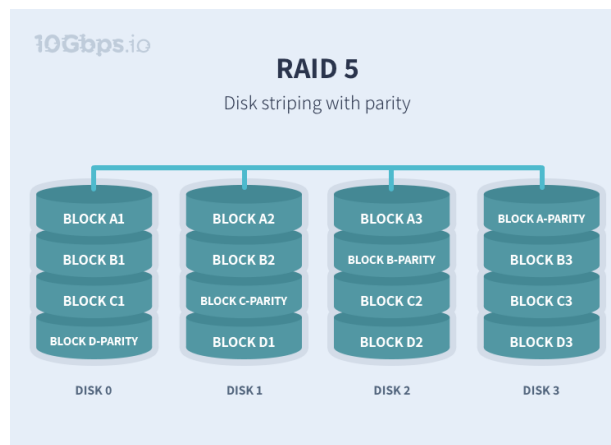
- Tipo 1: Se duplica la información



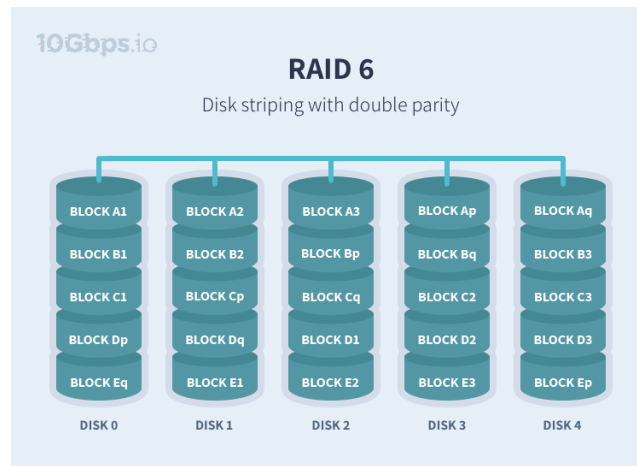
- Tipo 4: Se distribuye la información en todos los discos menos en uno, que se utiliza para almacenar únicamente datos de paridad.



- Tipo 5: Distribuye la información en discos, pero también guarda datos de paridad, usados para recuperar información en caso de errores (en el hardware). Ofrece mejor velocidad pues la información está distribuida en diferentes discos, y si alguno de ellos falla la información se puede recuperar en base al resto de datos y los bloques de paridad.

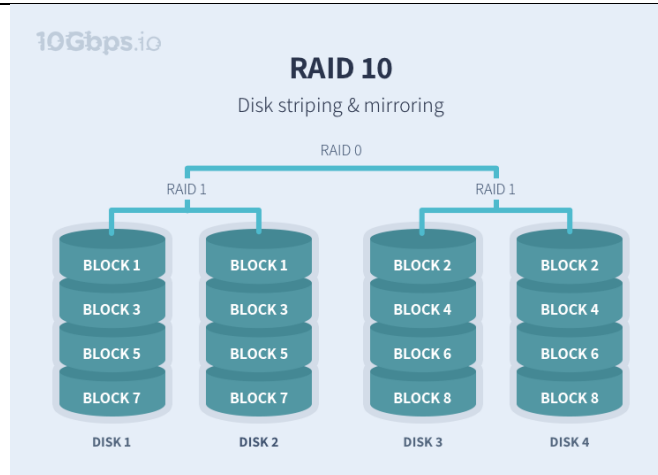


- Tipo 6: Similar a RAID5, solo que almacena un bloque de paridad extra (o sea, existen una copia adicional de los bloques de paridad), permitiendo que dos discos fallen al mismo tiempo.



Cuando se combinan, el primero indica el nivel más bajo

- O+1: Dos segmentos distribuidos y duplicados 1-2 – 1-2
- 1+0: Distribuidos en dos segmentos duplicados 1-1 – 2-2



LVM (Logical Volume Manager): Permite crear abstracciones y redimensionar las particiones del disco duro. Facilita la creación de grupos de volúmenes formados por distintos volúmenes físicos.

El volumen físico tiene particiones (en sí está completo), varios volúmenes físicos se pueden unir en una VG (volumen group). El volumen lógico coge un número de sectores del VG (no tienen por qué ser del mismo disco).

MD (Multiple Device): Es un array de dispositivos

Diferencias extended filesystems

- Ext2: Sistema de archivos de altas prestaciones indicado para discos fijos y sistemas extraíbles.
- Ext3: Versión de ext2 que incluye el Journaling
- Ext4: Mejoras sobre ext3 que aumentan el rendimiento y la fiabilidad, incluyendo además incrementos en los límites de tamaño para volúmenes, archivos y directorios.

Journaling: Hace referencia a un sistema de archivos que almacena un archivo especial (llamado *journal*) que se usa para reparar inconsistencias que pueden ocurrir (como al apagar el pc de golpe). Lo que hace es guardar metadatos antes de que algún comando retorne. En caso de crasheo, las actualizaciones se habrán aplicado al sistema de archivos o se habrán marcado como incompletas, en cuyo caso el sistema leerá el *journal* para retornar a un punto de consistencia.

Aunque se necesitan 2 discos mínimos para configurar un RAID, se podría formar con 1, pero carece de sentido.

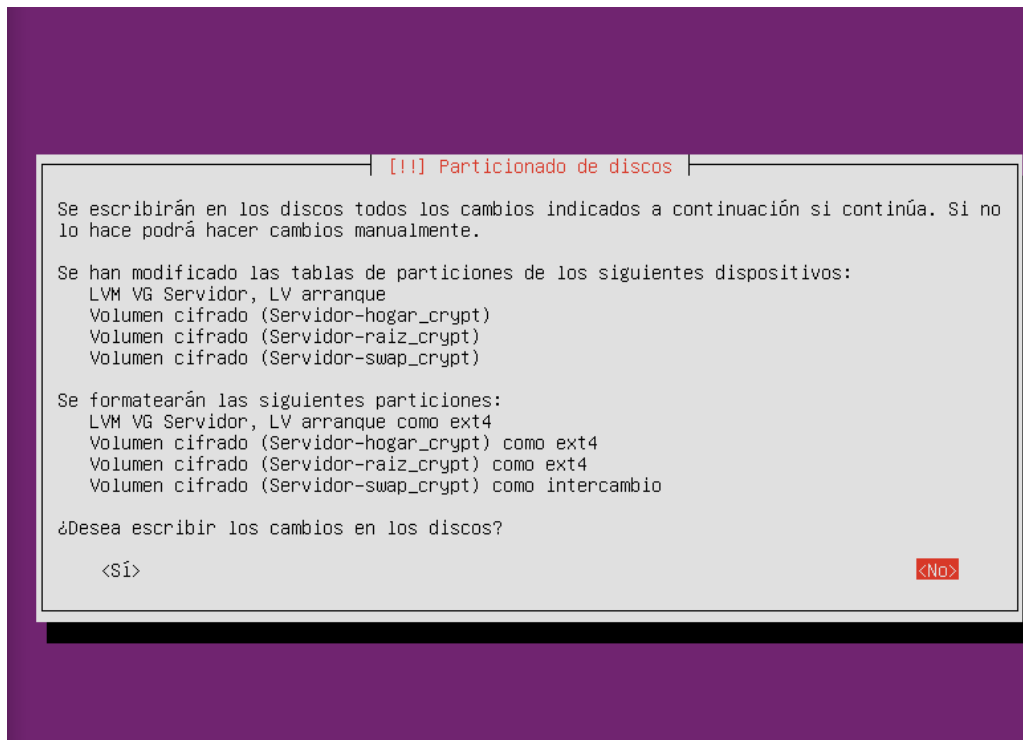
Un RAID puede ser:

- Por software: Necesita un controlador, en este caso el SO, que distribuye la información de acorde al tipo de RAID. Añade por tanto carga adicional al SO, empeorando prestaciones
- Por hardware: No necesita nada (en todo caso una tarjeta especial), siendo más rápido y más eficiente

Procedimiento de instalación de UbuntuServer

- Nombre máquina: ubuntuISEC3
- Usuario: ive Password: practicas,ISE
- No ciframos la carpeta personal, ya que queremos FDE (Full Disk Encrypted)
- Particionado de discos manual
- Raid 1
- 2 dispositivos activos, 0 libres
- Seleccionar las 2 particiones
- Crear grupo de volúmenes: Servidor
- Seleccionar el MD
- Crear volúmenes lógicos: swap (1GB), arranque (200 MB), hogar (800MB) y raiz (lo que sobre)
- Configurar volúmenes cifrados -> **create encrypted volumes** -> todo menos arranque
- Darle a sí y poner la contraseña a cada volumen (practicas,ISE)
- Seleccionar todos y poner punto de montaje (a los cifrados hay que darle a lo de debajo de Volumen cifrado). Utilizar como ext4 a todos menos swap (poner intercambio, no se pone punto de montaje)

```
LVM VG Servidor, LV arranque - 197.1 MB Linux device-mapper (linear)
#1          197.1 MB   f  ext4          /boot
LVM VG Servidor, LV hogar - 796.9 MB Linux device-mapper (linear)
#1          796.9 MB   K  cifrado        (Servidor-hogar_crypt)
Volumen cifrado (Servidor-hogar_crypt) - 794.8 MB Linux device-mapper (crypt)
#1          794.8 MB   f  ext4          /home
LVM VG Servidor, LV raiz - 6.6 GB Linux device-mapper (linear)
#1          6.6 GB    K  cifrado        (Servidor-raiz_crypt)
Volumen cifrado (Servidor-raiz_crypt) - 6.6 GB Linux device-mapper (crypt)
#1          6.6 GB    f  ext4          /
LVM VG Servidor, LV swap - 998.2 MB Linux device-mapper (linear)
#1          998.2 MB   K  cifrado        (Servidor-swap_crypt)
Volumen cifrado (Servidor-swap_crypt) - 996.1 MB Linux device-mapper (crypt)
#1          996.1 MB   f  intercambio    intercambio
Dispositivo RAID1 #0 - 8.6 GB Dispositivo RAID por software
#1          8.6 GB    K  lvm
SCSI3 (0,0,0) (sda) - 8.6 GB ATA VBOX HARDDISK
#1 primaria    8.6 GB    K  raid
SCSI4 (0,0,0) (sdb) - 8.6 GB ATA VBOX HARDDISK
#1 primaria    8.6 GB    K  raid
```



sda1: el número indica la cantidad de particiones del volumen físico (en nuestro caso no aparece porque no hay más de una)

MD0 (multi device 0): Es la agrupación del RAID, que engloba sda y sdb

Se debe encriptar todo menos el arranque, puesto que si no se necesitaría desencriptar antes de arrancar. El resto sí puesto que pueden contener información del usuario (la seguridad se consigue a costa de rendimiento)

- Instalar grub: Podría ser sda o sdb, (sudo grub-install /dev/sda es la operación que hace)
- En la terminal: Shutdown -h now

CentOS

Términos

Hay dos formas de direccionar volúmenes lógicos (con mapper o con el path absoluto)

- /dev/VG/LV
- /dev/mapper/VG-LV.

Tiene sentido cambiar el tamaño de bloque para ajustarlo a las características del sistema deseadas. En “mk2fs -b” se crea con el tamaño requerido.

No se montarían todos los volúmenes cuando no se quieran usar todas

En /etc/fstab viene la info de todo lo que se arranca junto al sistema

SELinux (contexto) (Security Extend Linux): Etiqueta con información de seguridad para que las aplicaciones no accedan a contextos que no debe. Es como un “policía”, indica las posibilidades que tiene cada proceso. Cuando queramos preservarlo, además de copiar los archivos habrá que hacer una actualización del contexto (porque no es recomendable perderlo)

Comandos: (-h = human-readable)

- **lsblk**: List block devices – Muestra todos los dispositivos de bloques del sistema
- **df -h**: Display the amount of disk space available on the file system
- **du**: Display file space usage of each file, recursively
- **lvmdiskscan**: Logical volumen manage disk scan
- **lvs**: Logical volumen shortlist
- **lvdisplay**: Logical volume display
- **vgs**: Volume group shortlist
- **vgdisplay**: Volume group display
- **pvs**: Physical volume shortlist
- **pvdisplay**: Physical volume display

Por defecto pone 1G de arranque, con 200MB sería más que suficiente

Extensión de LV añadiendo un disco

Añadir un disco

- Estando en VBox, añadir disco en propiedades, new SATA

Añadir un volumen físico

- Abrir máquina -> **pvcreate /dev/sdb** (No hace falta formatearlo aún)

Extender el grupo de volúmenes a sdb (si cl no existe, crearlo con **vgcreate /dev/sdb**)

- **vgextend cl /dev/sdb**

Crear un volumen lógico nuevo

- **lvcreate -L 4G -n newvar cl** (indica tamaño, nombre y grupo de volúmenes)

Crear sistema de archivos sobre el volumen lógico (ext4)

- (Make filesystem) **mkfs -t ext4 /dev/cl/newvar** (o **/dev/mapper/cl-newvar**)

Crear un directorio (media/newvar) y montar el LV (newvar)

- **mkdir /media/newvar** → **mount /dev/cl/newvar /media/newvar** (el volumen y el destino)

Se comprueba con **lsblk**

```
[root@localhost ivel# mkdir /media/newvar
[root@localhost ivel# mount /dev/cl/newvar
mount: can't find /dev/cl/newvar in /etc/fstab
[root@localhost ivel# mount /dev/cl/newvar /media/newvar
[root@localhost ivel# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   8G  0 disk
├─sda1                8:1    0   1G  0 part /boot
├─sda2                8:2    0    7G  0 part
│   └─cl-root         253:0    0  6,2G  0 lvm  /
│       └─cl-swap      253:1    0  820M  0 lvm  [SWAP]
└─sdb                 8:16    0    2G  0 disk
    └─cl-newvar        253:2    0    1G  0 lvm  /media/newvar
sr0                  11:0    1 1024M  0 rom
```

- Copiar el contenido del antiguo /var al nuevo LV (/media/newvar)

- Primero se “bloquea” a los usuarios (se les echa fuera) avisando de que hay mantenimiento previamente. Usarlo en situaciones totalmente necesarias
- Se usa systemctl (con el atributo status se ve la situación actual). Lo que hace es cargarse procesos (pero selectivo, no como kill):
systemctl isolate runlevel1.target
- No se puede estar tranquilo, ya que cerrar todos los procesos no impide que se creen uno nuevos. El nivel de seguridad no sube, podría haber tareas esperando a esta situación.

- **A partir de aquí los pasos se detallan más adelante**

Montar el LV en /var

Desmontar el newvar montado con anterioridad

Liberar información del antiguo /var

Entrar en modo de mantenimiento si no estábamos previamente

- **systemctl status**
- **systemctl isolate runlevel1.target**

Copiar el contenido de var a la nueva carpeta

- **cp -a /var/. /media/newvar** ó
- **cp -v -passive=all /var/. /media/newvar** (-a, copia recursiva que conserva los contextos)

```
[root@localhost ~]# cp -a /var/. /media/newvar
[root@localhost ~]# ls /media/newvar
adm  db      gopher  local  lost+found  opt      spool
cache empty  kerberos lock   mail       preserve tmp
crash games  lib     log    nis        run      yp
[root@localhost ~]# ls /var
adm  crash  empty  gopher  lib  lock  mail  opt  run  tmp
cache db     games  kerberos local log  nis  preserve spool yp
[root@localhost ~]# _
```

Se comprueba con **ls -Z /var** (listar con contexto)

- **cp /etc/fstab /etc/fstab.backup** (copia del fichero con la info de volúmenes lógicos, “opcional”, mejor hacer copia)
- **vi /etc/fstab** (más que un editor de textos)
 - i (insertar)
 - a (añadir)
 - ESC -> :q! (salir sin cambios)
 - ESC -> :wq (salir guardando)
 - Ruta de volumen lógico, punto de montaje, filesystem
 - **/dev/mapper/cl-newvar /var ext4 defaults 0 0**
 - Es interesante ver que root y swap no están montados persistentemente
 - Se montó antes para copiar los datos, lo añadimos a fstab para que se monte automáticamente con el arranque

```
#
# /etc/fstab
# Created by anaconda on Fri Oct  5 18:57:39 2018
#
# 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
#
/dev/mapper/cl-root    /                    xfs      defaults    0 0
UUID=0bfd2fae-bb0e-41ad-b9dd-4fa54d7e9712 /boot                xfs      defaults    0 0
ts                    0 0
/dev/mapper/cl-swap    swap                swap      defaults    0 0
/dev/mapper/cl-newvar  /var                ext4      defaults    0 0
```

- **#mount -a** (fuerza a que lea fstab y monte todo)
- **umount /media/newvar**
- **#umount /var**
- **#mount | grep var** -> No debe devolver nada
- Sería recomendable primero copiar el antiguo /var, y borrarlo con el tiempo: **cp -a /var/. /varold**
- **mv /var/ /varold**
- **mkdir /var**

Restaurar contexto recursivamente

- **restorecon -RV /var**

Forzar y volver a montar todo

- **mount -a**

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

[root@localhost ~]# ls /
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr  varold

[root@localhost ~]# ls /var
adm  db  gopher  local  lost+found  opt  spool
cache  empty  kerberos  lock  mail  preserve  tmp
crash  games  lib  log  nis  run  yp

[root@localhost ~]# ls /varold
adm  crash  empty  gopher  lib  lock  mail  opt  run  tmp
cache  db  games  kerberos  local  log  nis  preserve  spool  yp
```

Sesión 2 – Creación de RAID

Empezando desde 0

Levantar red e instalar mdadm (administración de multidispositivos, lo vamos a usar para crear un array)

- **ifup enp0s3** (equivalente a dhclient UNBOOT=no)
- **yum install mdadm**

Crear el RAID1 con los nuevos discos

(nivel-RAID1, numero de discos, dispositivos)

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

lsblk para comprobar configuración, debe salir md0 saliendo de sdb y de sdc, sda2 con cl-root y cl-swap

```
[root@localhost ivel# mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb /dev/sdc
mdadm: Note: this array has metadata at the start and
may not be suitable as a boot device.  If you plan to
store '/boot' on this device please ensure that
your boot-loader understands md/v1.x metadata, or use
--metadata=0.90
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
[root@localhost ivel# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	8G	0	disk	
sda1	8:1	0	1G	0	part	/boot
sda2	8:2	0	7G	0	part	
cl-root	253:0	0	6,2G	0	lvm	/
cl-swap	253:1	0	820M	0	lvm	[SWAP]
sdb	8:16	0	2G	0	disk	
md0	9:0	0	2G	0	raid1	
sdc	8:32	0	2G	0	disk	
md0	9:0	0	2G	0	raid1	
sr0	11:0	1	1024M	0	rom	

Crear volumen físico sobre md0

- **pvccreate /dev/md0**

Verificar con **pvs** o **pvddisplay**

```
[root@localhost ivel# pvccreate /dev/md0
Physical volume "/dev/md0" successfully created.
[root@localhost ivel# pvs
```

PV	VG	Fmt	Attr	PSize	PFree
/dev/md0		lvm2	---	2,00g	2,00g
/dev/sda2	cl	lvm2	a--	7,00g	0

No hay que extender el grupo de volúmenes, porque algunos datos pueden ir a otro disco. Por tanto se crea un grupo de volúmenes nuevo.

- **vgcreate pmraid1 /dev/md0** (nombre y punto de creación ¿)

vgs para comprobar que se ha creado o con **vgdisplay**

```
[root@localhost ivel# vgcreate pmraid1 /dev/md0
Volume group "pmraid1" successfully created
[root@localhost ivel# vgs
```

VG	#PV	#LV	#SN	Attr	USize	UFree
cl	1	2	0	wz--n-	7,00g	0
pmraid1	1	0	0	wz--n-	2,00g	2,00g

Ahora a crear los volúmenes lógicos (**lvs** para revisar)

- **lvcreate -L 1G -n newvar pmraid1** (tamaño, nombre y grupo de volúmenes)

```

root@localhost ivel# lvs
  LV      VG Attr      LSize   Pool Origin Data%  Meta%   Move Log Cpy%Sync Convert
  root    cl -wi-ao----   6,20g
  swap    cl -wi-ao---- 820,00m
root@localhost ivel# lvcreate -L 1G -n newvar pmraid1
Logical volume "newvar" created.
root@localhost ivel# lvs
  LV      VG Attr      LSize   Pool Origin Data%  Meta%   Move Log Cpy%Sync Convert
  root    cl -wi-ao----   6,20g
  swap    cl -wi-ao---- 820,00m
  newvar  pmraid1 -wi-a-----   1,00g
root@localhost ivel#

```

Crear un sistema de archivos en newvar

- **mkfs -t ext4 /dev/mapper/pmraid1-newvar** (o /dev/pmraid1/newvar) (ruta del volumen lógico)

Echar usuarios (se puede hacer después de crear el directorio y montarlo, pero siempre antes de la copia)

- **systemctl isolate runlevel1.target**

Y realizar copia

- **mkdir /mnt/newvar**
- **mount /dev/mapper/pmraid1-newvar /mnt/newvar** (cuando se le ha puesto un nombre al volumen lógico siempre se accede con un path)

mount | grep var para comprobar

- **cp -r --preserve=all /var/. /mnt/newvar**

Indicar que se monte automáticamente

- **cp /etc/fstab /etc/fstab.orig**
- **echo '/dev/mapper/pmraid1-newvar /var ext4 default 0 0' >> /etc/fstab** (también vale vi)

Con **cat** se comprueba

Liberar información

- **umount /mnt/newvar** (desmontar el temporal, hay dos apuntando al mismo sitio (/var??))
- Verificar que no está con **mount | grep var** (no debe devolver nada)
- (En este momento sería hacer una copia de /var, y eliminarla más adelante si todo ha salido bien)
- **mv /var /varold**
- **mkdir /var**

Con **ls -Z /var** no debe mostrar nada

Restaurar contexto

- **restorecon /var**
- **mount -a**
- **ls -Z /var** si debe mostrar cosas

Salir

- **shutdown -h now** (y hacer snapshot)

Ahora se podría eliminar un disco (sda, sdb) y la configuración se debería mantener (hacer copia/snapshot por si acaso)

Sesión 3 – Encriptación

Términos

LUKS (Linux Unified Key Setup)

- LVM sobre LUKS: Cifrar dispositivos, y aplicar LVM. Solo ciframos el dispositivo que pretendemos cifrar (en este caso sería md0)
- LUKS sobre LVM: Aplicar LVM (crear volúmenes físicos...) y aplicar LUKS. Sobre cada volumen lógico deseado se aplicaría cifrado (lo que se hizo en Ubuntu)

Procedimiento

Al entrar en modo mantenimiento se “carga” la red, hay que levantarla

- **ifup enp0s3**
- **su**
- **yum install --y cryptsetup**

Ahora a encriptar

- **systemctl isolate runlevel1.target**
- **mkdir /varRAID**
- **cp -r --preserve=all /var/. /varRAID/** (el punto es asegura que se copian los ocultos)

Desmontar /var

- **umount /var**
- **mount | grep var** (no debe devolver nada ¿?)

Copiar contenido de /var

- **unalias cp** (para definir el comportamiento de cp)
- **cp -r --preserve=context** (interesado en el contexto) **/varRAID/. /var** (se copia de vuelta porque se había desmontado antes)

Inicializamos una partición LUKS y añadimos su clave

- **cryptsetup luksFormat /dev/mapper/pmraid1-newvar** (la contraseña: practicas,ISE)

Abrimos la partición LUKS y le mapeamos un nombre

- **cryptsetup luksOpen /dev/mapper/pmraid1-newvar pmraid1-newvar_crypt** (o pmraid1_newvar_crypt)

Comprobable con **lblk**

```

Esto sobrescribirá los datos en /dev/pmraid1/newvar de forma irrevocable.
Are you sure? (Type uppercase yes): YES
Introduzca la frase contraseña:
Verifique la frase contraseña:
[root@localhost ~]# cryptsetup luksOpen /dev/pmraid1/newvar pmraid1-newvar_crypt
Introduzca la frase contraseña de /dev/pmraid1/newvar:
[root@localhost ~]# lsblk

```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	8G	0	disk	
└─sda1	8:1	0	1G	0	part	/boot
└─sda2	8:2	0	7G	0	part	
└─┬─cl-root	253:0	0	6,2G	0	lvm	/
└─┬─cl-swap	253:1	0	820M	0	lvm	[SWAP]
sdb	8:16	0	2G	0	disk	
└─md0	9:0	0	2G	0	raid1	
└─┬─pmraid1-newvar	253:2	0	1G	0	lvm	
└─┬─pmraid1-newvar_crypt	253:3	0	1022M	0	crypt	
sdc	8:32	0	2G	0	disk	
└─md0	9:0	0	2G	0	raid1	
└─┬─pmraid1-newvar	253:2	0	1G	0	lvm	
└─┬─pmraid1-newvar_crypt	253:3	0	1022M	0	crypt	
sr0	11:0	1	1024M	0	rom	

Creando un sistema de archivos

- **mkfs -t ext4 /dev/pmraid1/newvar_crypt**
(Puede que no funcione, pero parece que desde el mapper si va: /dev/mapper/pmraid1-newvar_crypt)

```

32768, 98304, 163840, 229376
Allocating group tables: hecho
Escribiendo las tablas de nodos-i: hecho
Creating journal (4096 blocks): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho
o
[root@localhost ~]# lsblk

```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	8G	0	disk	
└─sda1	8:1	0	1G	0	part	/boot
└─sda2	8:2	0	7G	0	part	
└─┬─cl-root	253:0	0	6,2G	0	lvm	/
└─┬─cl-swap	253:1	0	820M	0	lvm	[SWAP]
sdb	8:16	0	2G	0	disk	
└─md0	9:0	0	2G	0	raid1	
└─┬─pmraid1-newvar	253:2	0	1G	0	lvm	
└─┬─pmraid1-newvar_crypt	253:3	0	1022M	0	crypt	
sdc	8:32	0	2G	0	disk	
└─md0	9:0	0	2G	0	raid1	
└─┬─pmraid1-newvar	253:2	0	1G	0	lvm	
└─┬─pmraid1-newvar_crypt	253:3	0	1022M	0	crypt	
sr0	11:0	1	1024M	0	rom	

- **mkdir /mnt/varCifr**
- **mount /dev/mapper/pmraid1-newvar_crypt /mnt/varCifr/**
- **cp -r --preserve=all /varRAID/. /mnt/varCifr**

En /etc/crypttab (aparecen los dispositivos montados que están cifrados) (pero no sale ninguno)

Blkid (da el UUID (Universal Unique Identifier))

- **blkid | grep crypt (o crypto) >>/etc/crypttab** (luego habrá que editarlo con vi)

```
Example 1. /etc/crypttab example

Set up four encrypted block devices. One using LUKS for normal storage,
another one for usage as a swap device and two TrueCrypt volumes.

luks        UUID=2505567a-9e27-4efe-a4d5-15ad146c258b
swap        /dev/sda7          /dev/urandom          swap
truecrypt   /dev/sda2          /etc/container_password tcrypt
hidden      /mnt/tc_hidden    /dev/null             tcrypt-hidden,tcrypt-keyfile

[root@localhost ~]# blkid | grep crypt
/dev/mapper/pmraid1-newvar: UUID="b755d755-5199-49c2-9526-9d192165049f" TYPE="crypto_LUKS"
/dev/mapper/pmraid1-newvar_crypt: UUID="a02de45f-7980-49e2-8da0-77abf6f94f4e" TYPE="ext4"
[root@localhost ~]#
```

- o vi /etc/crypttab
- o Tiene que quedar de la siguiente manera:
- o Pmraid1-newvar_crypt UUID=(lo que haya salido en blkid | grep crypt) none
- o Guardar y salir
 - Con **cryptsetup luksUUID /dev/mapper/pmraid1-newvar** dará el UUID para poner arriba

Poner información para que se monte el volumen lógico (se debería hacer previamente una copia de seguridad)

(Ahora se debería limpiar el contenido /var, para que solo estén los archivos cifrados en el sistema)

- **vi /etc/fstab**
 - o **/dev/mapper/pmraid1-newvar_crypt /var ext4 default 0 0**
- **umount /mnt/varCifr**
- **mount -a**

Sesión 3 – Configuración de red

Seguir los pasos del guión, previamente:

- Recordar configurar VirtualBox, Files, Host ..., Create

Para cada máquina virtual:

- Properties, Network, Adapter 2, Mark box, Host-Only

Ubuntu Server: editar el archivo **/etc/network/interfaces** y añadimos al final:

```
auto enp0s8
iface enp0s8 inet static
address 192.168.56.105
```

```

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
iface enp0s3 inet dhcp

auto enp0s8
iface enp0s8 inet static
address 192.168.56.105

```

CentOS: Añadimos un nuevo archivo (script) a /etc/sysconfig/network-scripts con el nombre ifcfg-enp0s8 (tras saber que nuestra interfaz está ahí resultado de ejecutar `lspci | grep Ether`) con el siguiente contenido:

```

TYPE=Ethernet
BOOTPROTO=none
NAME=enp0s8
DEVICE=enp0s8
ONBOOT=yes
IPADDR=192.168.56.110
NETMASK=255.255.255.0

```

Ubuntu tiene ip: 192.168.56.105

CentOS tiene ip: 192.168.56.110

Tipo de preguntas de examen:

- De las siguientes opciones di cuales conservan el contexto y cuales no:
 - cp -a Copia incluyendo el contexto
- ¿Qué beneficios puede tener usar LVM?
 - Permite instalar un gestor de arranque X
 - Modificación dinámica de particiones O
 - Aumenta las prestaciones del sistema de archivos X
 - Ninguna de las anteriores X
- ¿Qué herramienta de gestión de paquetes está disponible en CentOS?

yum	O
lynx	X
apt	X
wget	X

- Explicar para qué se encripta la partición swap, o si es obligatorio
Encriptando también home y root, conseguimos “seguridad” total en los datos del dispositivo. Si no se encriptase swap, se podría hacer un volcado de los datos no cifrados en swap y se obtendrían descodificados.
- Comando para activar un volumen cifrado
`cryptsetup luksOpen [path del volumen]`
- Cómo se montaría un home igual que se ha hecho con /var
- Diferencias en la instalación de CentOS y Ubuntu
CentOS se instaló en una única partición, mientras que en Ubuntu se creó un RAID1.
El boot de CentOS es sobredimensionado (viene con 1GB por defecto), mientras que en Ubuntu se lo hemos indicado manualmente.
- ¿Qué significa SELinux?
Security Extend Linux: Etiqueta con información de seguridad para que las aplicaciones no accedan a contextos que no debe. Es como un “policía”, indica las posibilidades que tiene cada proceso. Cuando queramos preservarlo, además de copiar los archivos habrá que hacer una actualización del contexto (porque no es recomendable perderlo)
- Fichero que sirve como interfaz de red en CentOS
`/etc/sysconfig/network-scripts/enp0s3`
- Orden para crear un RAID1
`mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb /dev/sdc`