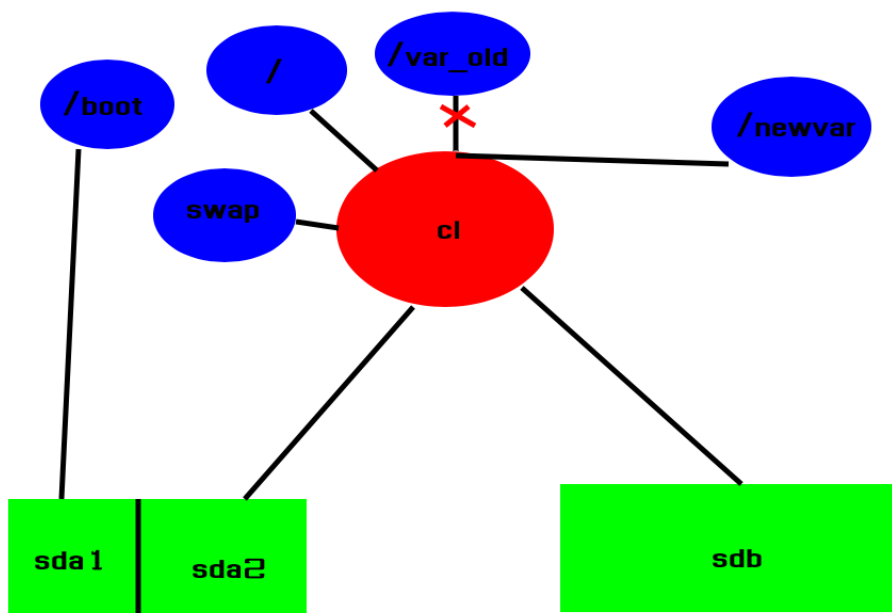


Guión práctica 1-2 y 1-3 de ISE

Vamos a crear un nuevo `/var` para almacenar archivos de gran tamaño. Para ello vamos a empezar creando un nuevo volumen lógica y asignarle un grupo de volúmenes y un volumen físico. En este caso no se necesitará establecer ningún tipo de cifrado. Antes de empezar con la configuración de CentOS (el sistema operativo que vamos a usar) vamos a ver con un esquema que queremos conseguir:



En la imagen podemos ver los physical volumes (en verde) que vamos a usar, los volumes groups (en rojo) y los logical volumes (en azul). Ahora sí, vamos a ver como configurar los discos:

1. Vamos a empezar insertando un nuevo disco en nuestro sistema. Para ellos nos vamos a VirtualBox y seleccionamos nuestra máquina virtual. Ahora le damos a Configuración -> Discos y seleccionamos el controlador: SATA y añadimos un nuevo disco. Una vez hecho esto ya podemos arrancar la máquina.
2. Ahora una vez dentro (se recomienda usar el usuario root) vamos a empezar por listar los volúmenes que tenemos instalados para ver sobre cuál queremos trabajar. Para ello usamos el comando `lsblk` y vemos como el nuevo volumen que hemos incluido es `/dev/sdb`.
3. Vamos a crear un volumen físico a partir de este utilizando la orden `pvcreate /dev/sdb`. Podemos comprobar su correcta creación con el comando `pvdisplay` o su forma

reducida `pvs` .

4. Ahora vamos a añadir el nuevo volumen que tenemos al grupo de volúmenes que ya existe, el `cl`. Para ello usamos la orden `vgextend cl /dev/sdb` , y comprobamos ahora con `pvs` que en la categoría de VG (volume group) aparece la etiqueta `cl`.
5. El siguiente paso será ya crear un nuevo volumen lógico donde diremos que irá almacenado el `/var` . Esto se hace utilizando la orden `lvcreate -L 1G -n newvar cl` La opción `-L` sirve para especificar el tamaño y `-n` para el nombre. Vamos a comprobar que está todo correcto utilizando `lsblk` .
6. Ya tenemos el volumen creado y correctamente ubicado, pero sin embargo todavía no dispone de un formato correcto. Para darle el formato más utilizado en Linux (el `ext4`) hacemos `mkfs -t ext4 /dev/cl/newvar` . En este caso hemos utilizado la opción `-t` para especificar el tipo de formato.
 1. También podríamos haber usado `mkfs -t ext4 /dev/mapper/cl-newvar` y en este caso funcionaría usar el tabulador para autocompletar, pero el resultado es el mismo. A partir de ahora el mapper se podrá usar siempre que se desee en lugar de `/dev/cl/newvar` .
7. Ahora lo que toca sería crear el lugar donde vamos a montar nuestro nuevo `/var` y posteriormente montarlo. Entonces hacemos `mkdir /mnt/newvar` seguido de `mount /dev/cl/newvar /mnt/newvar` . A partir de ahora si estuviéramos trabajando en un servidor habría que entrar en modo mantenimiento haciendo `systemctl isolate rescue` obligatoriamente como usuario `root`. Comprobar que está en mantenimiento con `systemctl status` .
8. Antes que nada vamos a añadir el cambio anterior al fichero `/etc/fstab` utilizando el editor de shell `nano` . Añadimos una línea como la siguiente:

```
/dev/mapper/cl-newvar    /var                    ext4    defaults    0 0
```

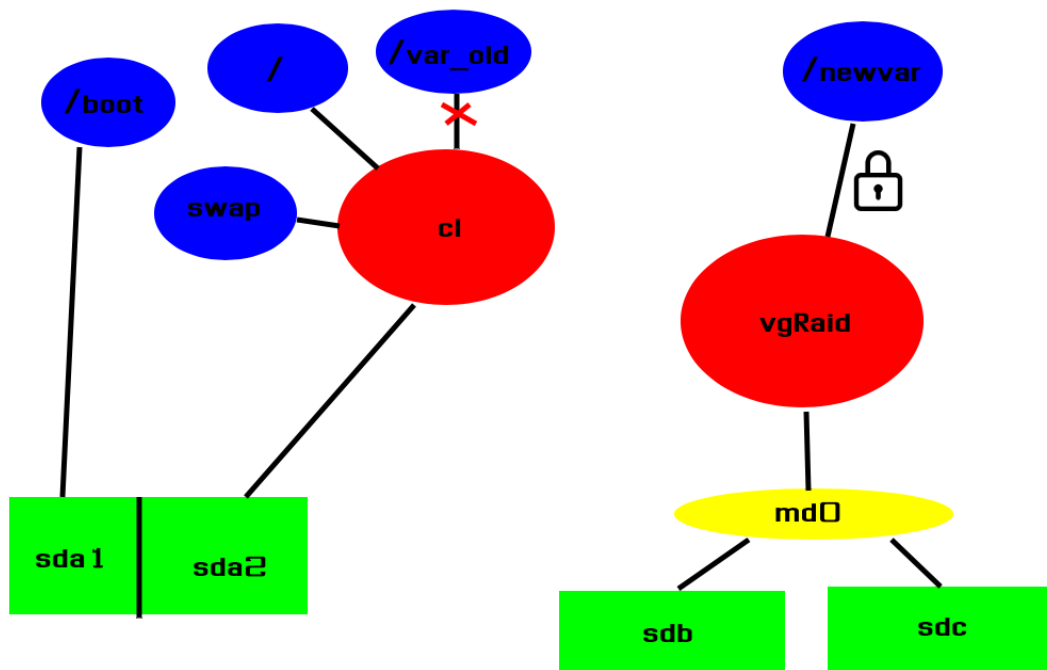
Siendo muy importante que la separación entre los elementos sean en este orden: 1 tab, 3 tabs, 1 tab, 1 tab, espacio.

Además vamos a eliminar (más bien renombrar) nuestro antiguo `/var` utilizando para ello `mv /var /var_OLD` y copiamos su contenido en `/mnt/newvar` con la orden `cp -r /var_OLD/* /mnt/newvar` . La orden `-r` lo que hace es que copia de forma recursiva.

9. Vamos ahora a crear la carpeta `/var` donde vamos a guardar el `newvar` con `mkdir /var` . Lo siguiente será desmontar el `/mnt/newvar` con la orden `umount -l`

`/mnt/newvar` (la orden `-l` solo tiene sentido si trabajamos en un servidor pues se espera a poder desmontarlo cuando nadie está utilizándolo). Ahora vamos a restaurar el contexto de `/var` con `restorecon /var` y posteriormente aplicamos el fstab con el comando `mount -a`.

Ahora vamos a hacerlo de forma que trabajemos sobre un raid y además vamos a encriptar los datos. Igual que en el caso anterior gráficamente tenemos:



Lo que tenemos que configurar ahora en nuestro sistema es lo siguiente:

1. Empezamos añadiendo esta vez dos discos en vez de uno y arrancamos el sistema con normalidad.
2. Ahora vamos a crear el raid de los dos discos con el siguiente comando:

`mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb /dev/sdc`. En este caso estamos usando las opciones `--create` para indicarle que lo queremos crear y posteriormente le damos un nombre, `--level=1` para indicar que tipo de raid queremos (leerse la teoría) y `--raid-devices=2` para indicarle cuántos y cuáles discos queremos que formen el raid.

3. Seguimos con los mismos pasos que el anterior pero ahora usando como disco `/dev/md0` y creando un nuevo volume group que llamaremos `vgRaid`. Paramos al terminar el paso 5.

4. Ahora vamos a cifrar el volumen utilizando el comando `cryptsetup luksFormat /dev/mapper/vgRaid-newvar` y posteriormente descifrarlo utilizando `cryptsetup luksOpen /dev/mapper/vgRaid-newvar vgRaid-newvar-crypt` . Ahora seguimos hasta terminar el punto 7 de la práctica anterior.
5. Ahora vamos a crear un archivo para descifrar automáticamente cuando encendamos el sistema. Además este fichero debe saber la UUID así que hacemos `blkid | grep crypto`
>> `/etc/crypttab` y lo editamos hasta que quede la siguiente línea:

```
vgRaid-newvar-crypt UUID=numero_largo none
```

6. A partir de aquí tan solo seguimos los pasos de la práctica anterior a partir del punto 8.