

# PRÁCTICA 1

---

## Lección 1

---

Estamos en el contexto de una empresa que nos solicita crear un VPS para la implantación un comercio electrónico que utilizará un CMS (control management service). Se nos recomienda la configuración con un RAID1, que nos va a duplicar la información y que estará gestionado por un LVM y se nos pide que se cifre toda la información. También se recomienda crear LVMS para el hogar, el directorio raíz y para el swap, y una partición para el boot.

### Procedimiento para la resolución del problema:

Creamos una máquina virtual de Ubuntu: le damos el nombre que sea, de memoria RAM con 500MB sería suficiente pero lo dejamos por defecto en 1GB. A continuación creamos el disco dur virtual; las máquinas virtuales son esa abstracción que se implementan con archivos y que ejecuta el software de virtualización y cada máquina tiene asociado su disco duro virtual (es en verdad un archivo), dejamos VDI y en cuanto al almacenamiento en unidad del disco duro física, podemos elegir entre reservado dinámico (cuando se necesite espacio, se reserva el tamaño del disco) y el tamaño fijo y creamos.

Configuramos ahora el almacenamiento: insertamos la imagen del sistema operativo, indicando que es un CD/DVD en vivo, para que arranque. Como el enunciado nos dice que hemos de configurar el sistema con un RAID1, serán necesarios 2 discos.

Ahora vamos a insertar dicho disco con los pasos análogos que hemos llevado a cabo al principio.

Arrancamos entonces la máquina y llevamos a cabo la instalación correspondiente. El proxy se edita en la configuración si estamos. A continuación vamos a definir un almacenamiento a medida:

### Almacenamiento a medida

Nos aparecen los dos discos que hemos creado, ambos de igual tamaño. Lo primero que tenemos que hacer es ser conscientes de la instalación que vamos a utilizar. Se está recomendando gestionar el LVM y configurar un RAID1. Tenemos nuestros discos duros, si los particionamos, se crea una primera capa lógica que nos permite distribuir mejor la información.

Tenemos que asegurarnos de que dejamos espacio para cargar el sector de arranque para que si uno falla, el otro disco pueda subsanar el error y poder arrancar. Por ello, le daremos al GRUB 1MB. Entonces, creamos una partición, sin formato de momento, en cada uno de los discos para el arranque, donde mapearemos el /boot, ambas de 400M (el doble de lo que

ocupe la imagen del kernel, porque si vamos a actualizar, queremos tener la anterior en caso de que haya algún problema).

Sobre las particiones nuevas de 400M que hemos creado para el boot en ambos discos, vamos a crear una primera abstracción para nuestro RAID, y se denominará md0. Una vez creado dicho md0, veremos nuestro nuevo dispositivo disponible. A continuación creamos el RAID1 para la partición que va a contener el resto de los datos (resto de espacio libre). Creamos dichas particiones, sin formatear (en el momento en el que especifiquemos un sistema de archivos concreto para la partición, no podremos crear el RAID1) y creamos el RAID1 sobre dichas particiones. Obsérvese que ya no tenemos disponibles los dispositivos sda y sdb (discos originales), si no que ya para continuar con el particionado estamos tratando con la abstracción (RAID) y hacemos referencia a esos dispositivos RAID con md0 y md1 .

Pasamos a asignarle el punto de montaje y el formato a la partición del arranque (a la md0 --> format), eligiendo como SA el ext4 y lo montamos en /boot.

Como en el enunciado se nos dice que configuremos el RAID gestionado con volúmenes lógicos, pasamos a crear nuestro grupo de volúmenes, escogiendo el dispositivo RAID md1, creándolo cifrado porque se nos pide en el enunciado. Así tendríamos nuestro grupo de volúmenes cifrado.

A continuación, creamos el volumen lógico para swap (formato el espacio de intercambio swap), home (ext4 como SA porque no contemplamos que se tengan que almacenar archivos demasiado grandes) y root , con los tamaños que corresponda según la documentación.

Ya tendríamos el esquema requerido montado.

Una vez hecho todo, iniciamos la máquina, nos pedirá la contraseña del disco cifrado y ya nos aparecerá el proceso de logeo. Una vez dentro, comprobamos el estado de la configuración con:

```
lsblk (list block devices)
```

El esquema es el siguiente:

---

## Lección 2

---

El escenario es distinto: vamos a utilizar Rocky. La empresa nos pide relizar una configuración de manera que se puedan almacenar unos videos de alta calidad y de gran tamaño. Para soportar dicha carga, vamos a utilizar el /var que vamos a ubicar dentro de un volumen lógico distinto y para dar fin, tendremos que añadirle un disco extra.

Creamos la máquina nueva y hacemos una instalación parecida a la anterior lección. Una vez esté la imagen del sistema operativo cargada, iniciamos la máquina. La configuración la hacemos modo texto. Reiniciamos la máquina sacando el disco previamente del almacenamiento porque si no vuelve a iniciar el proceso de instalación y configuración.

Una vez logueados, aplicaremos el comando `lsblk` y vemos que automáticamente Rocky nos ha creado una partición de 1G donde tiene `/boot` asignado como punto de montaje y luego tiene una partición `sda-2` de los 11G restantes del disco donde tenemos, gestionado por LVM, un grupo de volúmenes, "rl", que tiene dos volúmenes lógicos: `root` y `swap`. Gráficamente tenemos lo que presentamos a la izquierda, y queremos conseguir llegar al esquema de la derecha:

## Procedimiento:

Lo primer que tenemos que hacer es apagar la máquina y vamos a añadirle el disco (`sdb`). Iniciamos de nuevo la máquina y comprobamos que efectivamente la máquina ha detectado el disco (`lsblk`).

Podemos continuar creando nuestro physical volume utilizando todo el disco o creando una partición; los dos enfoques son posibles pero es recomendable crear una partición para tener espacio para metadatos o si en algún momento determinado se va a instalar GRUB en el `sdb`, se reserva algo de espacio para tal fin.

Como superusuario, vamos a darle formato a nuestro `/dev/sdb` (con `fdisk /dev/sdb`). Creamos una partición nueva (opción `n`), tecleamos el número 1. Cuando tecleemos `p`, veremos la partición y cuando tecleemos `w` pasamos a escribir la partición. Comprobamos que hemos creado bien la partición con `lsblk`.

A continuación vamos a crear a pasar nuestro physical volume. Para interactuar con LVM, podemos interactuar con el command line de LVM, donde veremos todos los posibles comandos con LVM. Utilizamos `pvdisplay` para ver todos los physical volume que tenemos.

El siguiente paso es crear el PV a partir del dispositivo `sdb1` (partición nueva que hemos hecho anteriormente al `sdb`). Hacemos entonces

```
pvcreate /dev/sdb1
```

Para ver que se ha creado exitosamente el PV, hacemos

```
pvdisplay
```

Ahora tenemos que extender nuestro volume group para que coja el nuevo PV que hemos creado. Para monitorizar el estado de nuestro grupo de volúmenes hacemos

```
vgs (lista corta del volume group) o vgdisplay.
```

Para extender el tamaño del grupo de volúmenes, utilizamos el comando

```
vgextend -L r1 /dev/sdb1 (vamos a extender el VG r1 con el dispositivo /dev/sdb1)
```

Observamos que se ha aumentado la capacidad y el espacio que tenía libre el VG.

Ya solo nos queda crear nuestro logical volume:

```
sudo lvcreate -n new_var -L 3G r1
```

```
lvdisplay
```

Y ya tenemos nuestro volumen lógico para new\_var. A nivel de configuración, la parte de gestión de LVM está lista. Lo que nos queda son cuestiones relativas a SOs.

A continuación, para continuar con el procedimiento, en primer lugar tenemos que crear un FS (file system) para el LV, acceder a LV (montarlo para que el SO pueda acceder ahí), copiaremos la información del /var al LV (se realizará de forma atómica), indicamos al sistema operativo dónde va /var y por último liberar espacio.

Montaje del Volumen Lógico:

```
sudo mkdir /new_var/
```

```
sudo mkfs -t ext4 /dev/r1/new_var (creamos el file system adecuado)
```

(hacemos accesible el SA mediante mount y le decimos que monte el dispositivo en /new\_var/)

```
sudo mount /dev/cl/new_var /new_var/
```

Ahora vamos a llevar a cabo la copia de la información de /var en el volumen lógico, de forma atómica:

```
sudo systemctl isolate rescue
```

```
sudo systemctl status (comprobamos que todavía estamos en running)
```

```
sudo systemctl isolate rescue (ya si estamos en modo mantenimiento y tenemos los procesos necesarios para que el sistema funcione, solo los necesarios para que mientras hacemos la copia, no haya problemas)
```

Copiamos recursivamente (-a):

```
cp -a /var/. /new_var/
```

```
ls -laZ /new_var/ para comprobar que se ha copiado todo
```

Una vez que ya hemos copiado toda la información, lo que tenemos que indicarle al SO es dónde tiene que asignar el punto de montaje /var. Para ello, nos vamos al archivo /etc/fstab:

```
/dev/mapper/r1-new_var /var ext4 defaults 0 0
```

Ahora hacemos:

```
umount /new_var
```

Tecleando a continuación

```
mount -a
```

va a montar todos los archivos dentro del filesystem table y ya tendríamos nuestro /var montado dentro de nuestro nuevo volumen lógico creado.

Ya tendríamos la configuración deseada, hasta cierto punto. Ya nos quedaría es liberar el espacio. ¿Qué ha pasado con ese /var que teníamos originalmente?, porque recordemos que hemos hecho una copia. Para liberar ese espacio ocupado, editamos el fstab:

```
# linea que hemos añadido comentada
```

```
mount -a
```

```
umount /dev/r1/new_var
```

Volvemos a editar el fstab para asegurarnos de que cuando lo llamemos esté preparado para funcionar, o en caso de que haya que reiniciar la máquina (descomentar la linea).

Ahora movemos /var a /var\_old con la finalidad de que si hay algún problemilla, podamos recuperar la información rápidamente a golpe de mover un directorio:

```
mv /var /var_old
```

```
mount -a (nos dirá que el punto de montaje no existe)
```

```
mkdir /var
```

```
sudo restorecon /var (restaura el contexto de los directorios)
```

```
mount -a
```

Y ya tenemos nuestro /var montado.

---

## Lección 3

---

El escenario en el que nos encontramos es un cliente que nos pide cifrar la información y que dicha información siempre esté disponible.

El esquema del que partimos y al que queremos llegar es el que se presenta en la siguiente imagen:

Partimos de una máquina desde cero, aunque podemos hacer un **snapshot** de la máquina anterior de la lección 2. Elegimos como imagen la de **Rocky**.

Una vez que nos hayamos logueado, bien en la nueva máquina o en el snapshot de la anterior, apagamos la máquina y añadimos dos discos (sdb y sdc).

Comprobamos que tenemos los dos discos reconocidos con lsblk.

Vamos a utilizar el comando nuevo:

```
mdadm (multi device administration)
```

Como buena praxis, es bueno formatear los discos añadiéndole una partición:

```
sudo fdisk /dev/sdb  
...  
sudo fdisk /dev/sdc
```

Una vez que hemos formateado dichos discos, podemos pasar a crear nuestro **multidevice**, nuestro RAID1, con el comando **mdadm**:

```
sudo mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb1 /dev/sdb2
```

Ya tendríamos el dispositivo md0 creado (para verlo: ls /dev).

A continuación, vamos a crear el **Physical Volume** a partir de nuestro md0:

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

Comprobamos que está creado:

```
sudo pvs
```

Creamos nuestro **Volume Group**:

```
sudo vgcreate nombre_vg /dev/md0
```

Comprobamos que está creado:

```
sudo vgs
```

Falta crear el **volumen lógico asociado a /var**:

```
sudo lvcreate -a nombre -L 1.8G volume_group
```

Comprobamos:

```
sudo lvs
```

Ahora **ciframos** el volumen lógico. Para ello, usaremos la herramienta de **cryptsetup**. Si no lo tenemos instalado, lo instalamos:

Vemos si tenemos asignada una ip (ver si tenemos acceso a internet):

```
ip addr
```

Comprobamos si tenemos acceso fuera:

```
ping www.google.es
```

Si no, levantamos la interfaz enp0s3:

```
sudo ifup enp0s3
```

Comprobamos que tenemos ip con ip addr

Y ya sí podríamos salir fuera

Instalamos cryptsetup:

```
yum install cryptsetup
```

Ahora vamos a darle formato al volumen lógico; vamos a cifrarlo:

```
sudo cryptsetup luksFormat /dev/nombr_vl/new_var
```

```
yes + password
```

Con esto, ya tendríamos cifrado el volumen lógico. Una vez hecho esto, tenemos que acceder a él:

```
sudo cryptsetup luksOpen /dev/nombre_vl/new_var nombre_del_abierto
```

Como no nos da nada, para ver si se ha abierto, hacemos

```
ls /dev/mapper
```

Ya tendríamos nuestro diseño de la infraestructura. Los pasos siguiente, son repetir /var en un volumen lógico como hicimos en la Lección 2:

Creación del File System:

```
sudo mkfs -t xfs /dev/mapper/nombre_del_abierto
```

Lo ponemos en mantenimiento:

```
sudo systemctl isolate rescue.target
```

```
sudo systemctl status
```

Creamos el punto de montaje:

```
sudo mkdir /new_var
```

Montamos:

```
sudo mount /dev/mapper/nombre_del_abierto /new_var/
```

Copiamos el contenido de /var en /new\_var:

```
sudo cp -a /var/. /new_var/
```

Indicamos al fstab dónde queremos que monte:

```
/dev/mapper/nombre_del_abierto /var xfs defaults 0 0
```

Ahora le tenemos que indicar al S0 que cuando arranque, active ese volumen lógico:

¿Cómo le vamos a especificar el dispositivo cifrado?:

```
blkid | grep crypto (nos interesa el cryptoluks)
```

Cogemos la salida que nos da ese comando y la redirigimos a /etc/crypttab:

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

Editamos el archivo crypttab:

```
Le quitamos el prefijo /dev/mapper y le decimos  
que lo denomine con el nombre del abierto.  
Le quitamos las comillas y le quitamos el Type  
y ponemos none (le indicamos que no utilice  
ningún archivo extra)
```



Liberamos el espacio:

```
mv /var /var1_old  
mkdir /var  
restorecon /var
```

Reiniciamos la máquina:

```
reboot
```

Ya con lsblk vemos que efectivamente tenemos montado /var en un volumen cifrado que está en RAID1 con los dos discos sdb y sdc.