

# 1 Instalación sobre RAID e LVM

## 1.1 Sumario

- 1 Prerequisitos
- 2 Instalación sobre RAID 1 por software
  - ◆ 2.1 1.- Creación del dispositivo RAID software
  - ◆ 2.2 2.- Particionado del dispositivo RAID software
  - ◆ 2.3 3.- Finalizar el particionado y escribir los cambios en el disco
  - ◆ 2.4 4.- Chequeos
- 3 Instalación sobre RAID1 con LVM
- 4 Duplicación del GRUB para poder arrancar desde cualquiera de los 2 discos
  - ◆ 4.1 Primer procedimiento. Utilizando el comando dd
  - ◆ 4.2 Segundo procedimiento. Utilizando grub-install
- 5 Simulación de un fallo en el RAID
- 6 Ampliación del Grupo de Volúmenes con otro dispositivo RAID1
  - ◆ 6.1 Esquema de la configuración
  - ◆ 6.2 Conexión de discos virtuales para el RAID1
  - ◆ 6.3 Particionado de los discos
  - ◆ 6.4 Creación del nuevo dispositivo RAID1
  - ◆ 6.5 Creación del Volumen Físico LVM sobre el dispositivo RAID1
  - ◆ 6.6 Extensión del Grupo de Volúmenes VG1 con el Volumen Físico
  - ◆ 6.7 Extensión del Volumen Lógico VL1 con el nuevo Volumen Físico
- 7 Referencias

## 1.2 Prerequisitos

- Conocer el concepto **RAID**, y concretamente el nivel RAID 1 (Mirroring)
- Conocer el concepto **LVM** (Logical Volumen Manager)
- Disponer de una imagen .iso de Debian9
- Sistema de virtualización de escritorio: VirtualBox, KVM, HyperV, etc

## 1.3 Instalación sobre RAID 1 por software

Creamos una máquina virtual con **dos discos virtuales del mismo tamaño**. Conectamos un medio de instalación a la unidad de CDROM e iniciamos el proceso de instalación.

### 1.3.1 1.- Creación del dispositivo RAID software

Desde el instalador, **en el apartado de Particionado**, seguimos los siguientes pasos

- Particionado en modo Manual
- Crear una tabla de particiones en cada disco
- Definir cada partición como volumen físico para RAID
- Configurar RAID por software
- Crear un dispositivo MD
- RAID1
- Seleccionar dispositivos para RAID (cada una de las 2 particiones)
- Terminar

A la conclusión de estos pasos dispondremos de un dispositivo **/dev/md0** de tipo RAID1

### 1.3.2 2.- Particionado del dispositivo RAID software

Seguimos en el **instalador**

- Definir la partición #1 del dispositivo RAID1 /dev/md0
- Utilizar como ext4

- Montar en /
- Se ha terminado de definir la partición

### 1.3.3 3.- Finalizar el particionado y escribir los cambios en el disco

Tras lo cual terminamos la instalación del sistema.

### 1.3.4 4.- Chequeos

Comprobación de los dispositivos:

```
mdadm --detail --scan
cat /proc/mdstat
```

Ver los datos del dispositivo (para un dispositivo dado):

```
mdadm --detail /dev/md0
```

## 1.4 Instalación sobre RAID1 con LVM

Desde el **instalador** de Debian, en el **apartado de Particionado**, editar el dispositivo creado en el punto 2 del procedimiento anterior, /dev/md0, para que la partición que alberga sea utilizada como volumen físico para LVM. A continuación ejecutar los pasos:

- Configurar el Gestor de Volúmenes Lógicos LVM
- Mantener distribución
- Crear grupo de volúmenes con nombre **VG1**
- Añadir el dispositivo **/dev/md0**
- Crear un volumen lógico en **VG1** de nombre **VL1**
- Terminar
- Volvemos al punto 3 del apartado anterior para definir la partición de instalación en el volumen lógico creado **VL1**

Una vez completados los pasos anteriores la distribución de particionado debería de ser la siguiente

## Particionado de discos

Éste es un resumen de las particiones y puntos de montaje que tiene configurados actualmente. Seleccione una partición para modificar sus valores (sistema de ficheros, puntos de montaje, etc.), el espacio libre para añadir una partición nueva o un dispositivo para inicializar la tabla de particiones.

### Configurar RAID por software

#### Configurar el Gestor de Volúmenes Lógicos (LVM)

#### Configurar los volúmenes cifrados

#### Configurar los volúmenes iSCSI

#### ▼ LVM VG VG1, LV VL1 - 21.5 GB Linux device-mapper (linear)

> #1 21.5 GB f ext4 /

#### ▼ Dispositivo RAID1 #0 - 21.5 GB Dispositivo RAID por software

> #1 21.5 GB K lvm

#### ▼ SCSI3 (0,0,0) (sda) - 21.5 GB ATA VBox Harddisk

> #1 primaria 21.5 GB B K raid

#### ▼ SCSI4 (0,0,0) (sdb) - 21.5 GB ATA VBox Harddisk

> #1 primaria 21.5 GB K raid

### Deshacer los cambios realizados a las particiones

### Finalizar el particionado y escribir los cambios en el disco

Capturar la pantalla

Ayuda

Retroceder

Continuar



Tras terminar la instalación y reiniciar podremos comprobar como, efectivamente, el Sistema de Archivos ahora subyace sobre el Volumen Lógico VL1

```
mount -l | grep ext4
```

muestra

```
/dev/mapper/VG1-VL1 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
```

## 1.5 Duplicación del GRUB para poder arrancar desde cualquiera de los 2 discos

Al acabar la instalación anterior nos había pedido instalar el gestor de arranque GRUB en uno de los discos. GRUB necesita un soporte de dispositivo de bajo nivel, y no puede ser instalado en el dispositivo RAID1 /dev/md0.

El dispositivo /dev/md0 está compuesto por las dos particiones:

- **sda1**: primera partición del primer disco
- **sdb1**: primera partición del segundo disco

Este dispositivo no tiene nada que ver con GRUB, ya que solo conoce particiones en los discos, es decir, el arranque queda relegado a los discos. Por tanto, para poder arrancar el sistema, en un contexto de alta disponibilidad, desde cualquiera de los discos, deberíamos de duplicar GRUB en el disco en el que éste no se ha incluido en el proceso de instalación.

Puede ocurrir que el disco en el que se ha instalado GRUB falle y por tanto, ya que estamos usando un RAID1 en el sistema de archivos, sería deseable también poder arrancar desde el otro dispositivo. El problema es que si no tenemos GRUB instalado en el segundo disco no podremos arrancar desde él.

Vamos a duplicar GRUB en el segundo disco para que, en caso de que falle el primero, poder arrancar desde él.

### 1.5.1 Primer procedimiento. Utilizando el comando dd

Supongamos que el dispositivo en el que instalamos GRUB es /dev/sda, es decir, el primer disco duro SATA. Ejecutamos

```
dd if=/dev/sda of=/dev/sdb count=100
```

Con el comando anterior copiamos los primeros bloques, incluyendo al MBR, del primer disco duro /dev/sda en el segundo, /dev/sdb. Copiando un único bloque, con count=1, no funcionaría porque los metadatos que necesita GRUB se extienden más allá del primer bloque físico del disco.

A continuación desconectamos el primer disco duro, /dev/sda, de la máquina virtual y arrancamos. El resultado debería de ser el arranque normal del sistema.

Tras el arranque el dispositivo /dev/md0 debe de encontrarse en modo fallo, pues uno de los discos que alberga una de las particiones que le sirven de soporte no está presente.

Podemos ver ese estado ejecutando

```
mdadm --detail /dev/md0
```

Observando la salida de este comando podremos ver como, efectivamente, el RAID se encuentra en estado degradado.

Para solucionarlo, apagamos la máquina y volvemos a conectar el primer disco a la máquina virtual. Arrancamos y veremos que al ejecutar el comando anterior

```
mdadm --detail /dev/md0
```

la salida del comando indica que el array sigue estando degradado. Esto se debe a que mdadm ha detectado anteriormente la ausencia del disco /dev/sda y ha eliminado su partición del dispositivo /dev/md0, de modo que muestra el dispositivo como ?removed?.

A continuación ejecutamos:

```
mdadm --manage /dev/md0 --add /dev/sda1
```

Este comando añade de nuevo la partición del dispositivo en conflicto al RAID. El proceso tarda un tiempo, pues mdadm vuelve a copiar los datos de la partición ?sana?, en la nueva partición. Al terminar el estado del RAID vuelve a ser consistente.

### 1.5.2 Segundo procedimiento. Utilizando grub-install

Otro modo de proceder para poder arrancar con GRUB desde el segundo disco, /dev/sdb, es instalarlo directamente en ese disco, para ello ejecutamos

```
grub-install /dev/sdb
update-grub /dev/sdb
```

El resto del procedimiento es análogo al apartado anterior.

## 1.6 Simulación de un fallo en el RAID

Para probar el comportamiento de mdadm ante fallo de uno de los dispositivos dentro del array podemos utilizar el propio comando mdadm.

A continuación vamos a poner como faulty, ¿en fallo?, el primer dispositivo, correspondiente a la primera partición del primer disco duro del array

```
mdadm --manage /dev/md0 --fail /dev/sda1
```

Con el comando:

```
mdadm --detail /dev/md0
```

veremos las propiedades del volumen RAID1 y como, efectivamente, uno de los dispositivos aparece en estado de fallo y solo tenemos activo al dispositivo /dev/sdb1

A continuación reiniciamos la máquina

```
reboot
```

En el siguiente arranque usará el dispositivo activo, es decir el segundo disco duro. Si éste dispone de GRUB, podremos arrancar sin problema.

Tras el reinicio, podremos añadir de nuevo el dispositivo que pusimos faulty al array

```
mdadm --manage /dev/md0 --add /dev/sda1
```

Tardará un tiempo en completar el proceso de regeneración del array, el cual podremos monitorizar con el comando anterior

```
mdadm --detail /dev/md0
```

## 1.7 Ampliación del Grupo de Volúmenes con otro dispositivo RAID1

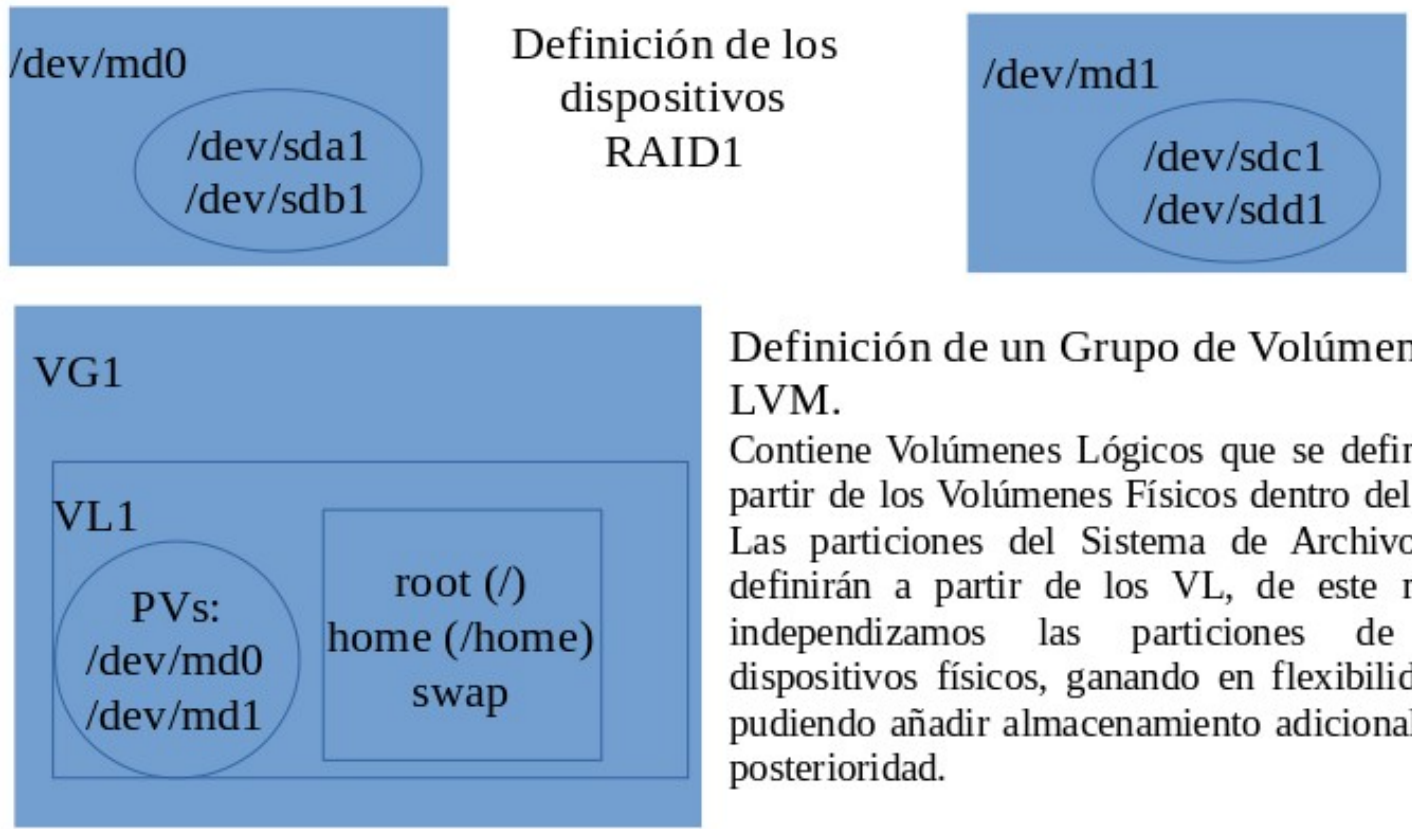
Vamos a hacer uso de las características de LVM para aumentar el espacio de almacenamiento disponible para el sistema de archivos. Para ello realizaremos los siguientes pasos:

- Creación de un nuevo dispositivo RAID1 sobre dos discos duros virtuales
- Particionado de los discos y creación del dispositivo RAID1 con mdadm
- Creación del PV (Volumen Físico) LVM para el dispositivo anterior
- Asignación del PV al VG (Grupo de Volúmenes) LVM utilizado en la instalación
- Ampliación del VL (Volumen Lógico) LVM utilizado en la instalación con el nuevo PV

### 1.7.1 Esquema de la configuración

En el siguiente gráfico vemos la distribución de los componentes:

- **/dev/md0 y /dev/md1** son los dispositivos RAID por software que serán utilizados como volúmenes físicos, es decir, como recursos de almacenamiento. Estos dispositivos, a su vez, contienen dos dispositivos físicos en modo ¿espejo?, RAID1, para disponer de seguridad física basada en redundancia de datos.
- **VG1** es el **Grupo de Volúmenes** LVM, dentro del cual se asignarán los **PV (Volúmenes Físicos)** LVM. Por último, dentro del VG también se define el **VL1, Volumen Lógico**, que conlleva asociados PV. Mediante el concepto de VL independizamos los recursos de almacenamiento, en este caso los dispositivos RAID1, del Sistema de Archivos desplegado sobre el VL. Los términos Grupo de Volúmenes (VG), Volumen Físico (PV) y Volumen Lógico (LV) son términos asociados a LVM.



### 1.7.2 Conexión de discos virtuales para el RAID1

En el siguiente procedimiento vamos a añadir otro dispositivo RAID1 por software como Volumen Físico asociado al Volumen Lógico VL1, dentro del grupo de volúmenes VG1 LVM configurado en nuestra instalación.

Añadimos a la máquina virtual **otros 2 discos virtuales** del mismo tamaño.

Al arrancar la máquina ejecutamos `lsblk` para ver los nuevos dispositivos conectados

```
lsblk
```

Como vemos en la salida hay dos nuevos dispositivos `sdc` y `sdd`, correspondientes a los dos nuevos discos virtuales conectados.

### 1.7.3 Particionado de los discos

Usamos el comando `fdisk`

```
fdisk /dev/sdc
```

Creamos una partición primaria que abarque todo el espacio, y con la opción `t` de `fdisk`, establecemos el tipo a `?fd?`, el cual hace referencia a `?Linux raid autodetect?`

Repetimos los pasos para el otro disco

```
fdisk /dev/sdd
```

### 1.7.4 Creación del nuevo dispositivo RAID1

A continuación vamos a crear el nuevo dispositivo RAID1 que abarcará la partición creada en los dos dispositivos anteriores

```
mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sdc1 /dev/sdd1
```

Este nuevo dispositivo llevará por nombre **md1** y su archivo de dispositivo será `/dev/md1`

## 1.7.5 Creación del Volumen Físico LVM sobre el dispositivo RAID1

Ejecutamos

```
pvcreate /dev/md1
```

Con el comando anterior creamos un volumen físico sobre el dispositivo RAID1 creado anteriormente. De este modo disponemos de un nuevo dispositivo de almacenamiento asignable al Grupo de volúmenes LVM **VG1** que creamos durante la configuración del instalador con LVM, el cual será asignado al Volumen Lógico **VL1**, de modo que extenderemos el tamaño de éste, y por tanto la disponibilidad de almacenamiento del Sistema de Archivos desplegado sobre VL1.

## 1.7.6 Extensión del Grupo de Volúmenes VG1 con el Volumen Físico

Para ello ejecutamos

```
vgextend VG1 /dev/md1
```

con el comando

```
vgdisplay
```

podremos ver los detalles de configuración de VG1.

El comando

```
pvscan
```

también resultará de utilidad para comprobar la disponibilidad de Volúmenes Físicos. En este caso la salida será del tipo

```
PV /dev/md0   VG VG1          lvm2 [7,99 GiB / 0   free]
PV /dev/md1   VG VG1          lvm2 [7,99 GiB / 7,99 GiB free]
Total: 2 [15,98 GiB] / in use: 2 [15,98 GiB] / in no VG: 0 [0   ]
```

## 1.7.7 Extensión del Volumen Lógico VL1 con el nuevo Volumen Físico

Ejecutando

```
lvdisplay
```

Vemos los detalles de configuración de los volúmenes lógicos LVM. En este caso nos mostrará la información relativa al único definido VL1

```
--- Logical volume ---
LV Path                /dev/VG1/VL1
LV Name                 VL1
VG Name                 VG1
LV UUID                 qpZEHB-bn0S-IGAV-9d0D-G06Y-ch5E-qamBxH
LV Write Access        read/write
LV Creation host, time debian, 2017-11-16 16:16:38 +0100
LV Status               available
# open                  1
LV Size                 7,99 GiB
Current LE              2046
Segments                1
Allocation              inherit
Read ahead sectors     auto
 - currently set to    256
Block device            253:0
```

Ahora vamos a añadir a VL1 el Volumen Físico /dev/md1

```
lvextend /dev/VG1/VL1 /dev/md1
```

Si ejecutamos de nuevo:

```
lvdisplay
```

```
--- Logical volume ---
LV Path                /dev/VG1/VL1
LV Name                VL1
VG Name                VG1
LV UUID                qpZEHB-bn0S-IGAV-9d0D-G06Y-ch5E-qamBxH
LV Write Access        read/write
LV Creation host, time debian, 2017-11-16 16:16:38 +0100
LV Status              available
# open                 1
LV Size                '15,98 GiB'
Current LE             4092
Segments              2
Allocation             inherit
Read ahead sectors    auto
- currently set to    256
Block device          253:0
```

Vemos como ahora el espacio LV size el volumen lógico ha aumentado al doble tras haber añadido el Volumen Físico /dev/md1 a VL1

Por último redimensionamos el espacio disponible para el Sistema de Archivos mediante el comando

```
resize2fs /dev/VG1/VL1
```

Comprobamos que efectivamente el espacio disponible para el Sistema de Archivos ha aumentado, para ello ejecutamos

```
df -h
```

## 1.8 Referencias

RAID <https://es.wikipedia.org/wiki/RAID>

LVM <https://blog.inittab.org/administracion-sistemas/lvm-para-torpes-i/>

[Volver](#)

JavierFP 15:18 18 nov 2017 (CET)