

LVM y RAID5 para el Almacenamiento de Cache y Registros del Squid Proxy

Mostramos los discos de 200 Gigabytes

```
sysadmin@proxy: ~  
sysadmin@proxy:~$ lsblk  
NAME                                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT  
sda                                  8:0      0  120G  0 disk  
├─sda1                               8:1      0   976M  0 part /boot  
├─sda2                               8:2      0     1K  0 part  
├─sda5                               8:5      0  119G  0 part  
│   └─vg_proxy-vl_swap              254:0    0   3.8G  0 lvm [SWAP]  
│   └─vg_proxy-vl_system            254:1    0 115.2G  0 lvm /  
sdb                                  8:16     0  200G  0 disk  
sdc                                  8:32     0  200G  0 disk  
sdd                                  8:48     0  200G  0 disk  
sde                                  8:64     0  200G  0 disk  
sr0                                  11:0     1 1024M  0 rom  
sysadmin@proxy:~$
```

Empezamos a crear las particiones con fdisk para cada disco de 200 Gigabytes. Solamente se creará una sola partición para cada disco. Empezamos con el disco sdb y debemos repetir los mismos pasos para los demás discos sdc, sdd, y sde.

```
sysadmin@proxy:~$ sudo fdisk /dev/sdb  
Bienvenido a fdisk (util-linux 2.36.1).  
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.  
Tenga cuidado antes de utilizar la orden de escritura.  
  
El dispositivo no contiene una tabla de particiones reconocida.  
Se ha creado una nueva etiqueta de disco DOS con el identificador de disco 0xcbbdeb51.  
  
Orden (m para obtener ayuda): n  
Tipo de partición  
p primaria (0 primaria(s), 0 extendida(s), 4 libre(s))  
e extendida (contenedor para particiones lógicas)  
Seleccionar (valor predeterminado p): p  
Número de partición (1-4, valor predeterminado 1):  
Primer sector (2048-419430399, valor predeterminado 2048):  
Último sector, +/-sectores o +/-tamano{K,M,G,T,P} (2048-419430399, valor predeterminado 419430399):  
  
Crea una nueva partición 1 de tipo 'Linux' y de tamaño 200 GiB.  
  
Orden (m para obtener ayuda): w  
Se ha modificado la tabla de particiones.  
Llamando a ioctl() para volver a leer la tabla de particiones.  
Se están sincronizando los discos.
```

Demostramos la partición del primer disco sdb. De la misma forma se puede validar las particiones de los demás discos.

```

sysadmin@proxy:~$ sudo fdisk -l /dev/sdb
Disco /dev/sdb: 200 GiB, 214748364800 bytes, 419430400 sectores
Modelo de disco: VMware Virtual S
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0xcbbdeb51

Disposit.  Inicio Comienzo      Final  Sectores  Tamaño  Id  Tipo
/dev/sdb1          2048 419430399 419428352    200G   83  Linux

```

Otra manera para demostrar las particiones de los discos de 200 Gigabytes.

```

sysadmin@proxy:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0      0  120G  0 disk
├─sda1                               8:1      0   976M  0 part /boot
├─sda2                               8:2      0     1K  0 part
├─sda5                               8:5      0  119G  0 part
│   └─vg_proxy-vl_swap              254:0    0   3.8G  0 lvm  [SWAP]
│   └─vg_proxy-vl_system            254:1    0 115.2G  0 lvm  /
sdb                                  8:16     0  200G  0 disk
├─sdb1                               8:17     0  200G  0 part
sdc                                  8:32     0  200G  0 disk
├─sdc1                               8:33     0  200G  0 part
sdd                                  8:48     0  200G  0 disk
├─sdd1                               8:49     0  200G  0 part
sde                                  8:64     0  200G  0 disk
├─sde1                               8:65     0  200G  0 part
sr0                                  11:0     1 1024M  0 rom

```

Ahora debemos instalar mdadm para administrar los arreglos de los discos en nivel 5. El “**md**” significa “**múltiples dispositivos**”.

```

sysadmin@proxy:~$ sudo apt search mdadm
[sudo] password for sysadmin:
Ordenando... Hecho
Buscar en todo el texto... Hecho
backupninja/stable 1.2.1-1 all
  lightweight, extensible meta-backup system

growlight/stable 1.2.31-1 amd64
  Disk manipulation and system preparation tool

mdadm/stable 4.1-11 amd64
  Tool to administer Linux MD arrays (software RAID)

sysstray-mdstat/stable 1.2.0-2 all
  Notifies about Linux Software RAID changes in system tray

sysadmin@proxy:~$ sudo apt install mdadm -y

```

Validamos que mdadm fue instalado correctamente.

```
sysadmin@proxy:~$ sudo dpkg -l mdadm
Desestado=desconocido(U)/Instalar/eliminar/Purgar/retener(H)
| Estado=No/Inst/ficheros-Conf/desempaquetado/medio-conf/medio-inst(H)/espera-disparo(W)/pendiente-disparo
| / Err?=(ninguno)/requiere-Reinst (Estado,Err: mayúsc.=malo)
|/ Nombre Versión Arquitectura Descripción
+++-----+-----+-----+-----+
ii mdadm 4.1-11 amd64 Tool to administer Linux MD arrays (software RAID)
```

Comenzamos con crear el arreglo de nivel 5 agregando cada partición de los discos en un solo dispositivo **“md0”**. Recuerden que **“md”** significa **“múltiples dispositivos”**.

```
sysadmin@proxy:~$ sudo mdadm --create /dev/md0 --level=5 --raid-devices=4 /dev/sd[b-e]1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

Una vez que finalizamos de crear y reconstruir las unidades en el arreglo de nivel 5, podemos revisar y validar la estadística del dispositivo **“md0”**.

```
sysadmin@proxy:~$ sudo mdadm --detail /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Fri Sep 16 16:23:45 2022
  Raid Level : raid5
  Array Size : 628746240 (599.62 GiB 643.84 GB)
  Used Dev Size : 209582080 (199.87 GiB 214.61 GB)
  Raid Devices : 4
  Total Devices : 4
  Persistence : Superblock is persistent

  Intent Bitmap : Internal

  Update Time : Fri Sep 16 16:41:19 2022
  State : clean
  Active Devices : 4
  Working Devices : 4
  Failed Devices : 0
  Spare Devices : 0

  Layout : left-symmetric
  Chunk Size : 512K

Consistency Policy : bitmap

  Name : proxy:0 (local to host proxy)
  UUID : bcd3f9bf:4aca1508:888f928d:b389cbe7
  Events : 212

  Number Major Minor RaidDevice State
    0      8     17        0   active sync  /dev/sdb1
    1      8     33        1   active sync  /dev/sdc1
    2      8     49        2   active sync  /dev/sdd1
    4      8     65        3   active sync  /dev/sde1
```

Podemos observar que en RAID5 tenemos 4 dispositivos. Cada uno dispone una capacidad total de 214.61 GB y en cada uno tiene disponible una capacidad de 199.87 GB. De los 4 dispositivos, solamente se suman las capacidades totales de 3 dispositivos. Esto quiere decir que un cuarto dispositivo no se adiciona en la sumatoria con los demás dispositivos porque es el dispositivo con la paridad para la tolerancia a fallos de un disco. Entonces, si multiplicamos 214.61×3 , obtendremos una capacidad total de 643.84 GB, el cual es la sumatoria de 3 dispositivos con una capacidad disponible de $599.62 \text{ GB} (199.87 \times 3)$.

Si revisamos las particiones con el RAID5, obtendremos los siguientes resultados.

```

sysadmin@proxy: ~$ lsblk -fm
NAME                                FSTYPE     FSVER  LABEL  UUID                                  FSAVAIL  FSUSE%  MOUNTPOINT  SIZE OWNER GROUP MODE
sda
├─sda1                               ext4        1.0    791ec00e-0602-4b77-80ec-344b50da953d  830.9M   9%      /boot        976M root disk brw-rw----
├─sda2                               ext4        1.0    0f9e6388-4c1d-45b8-ac3e-94c1f95c49c0  119G    root disk brw-rw----
├─sda5                               LVM2_member LVM2 001   shwVP6-Xwr3-B6d6-uhBL-qovP-aIA4-VCqgh4  119G    root disk brw-rw----
│   └─vg_proxy-vl_swap                swap        1      0fbde638-fc1d-45b8-ac3e-94c1f95c49c0  3.8G    root disk brw-rw----
│   └─vg_proxy-vl_system              ext4        1.0    387756be-640e-4a6d-b136-e06658fe2b90  105.7G  1%      /            115.2G root disk brw-rw----
sdb
├─sdb1                               linux_raid_member 1.2   proxy:0 bcd3f9bf-4aca-1508-888f-928db389cbe7  200G    root disk brw-rw----
│   └─md0                             599.6G   root disk brw-rw----
sdc
├─sdc1                               linux_raid_member 1.2   proxy:0 bcd3f9bf-4aca-1508-888f-928db389cbe7  200G    root disk brw-rw----
│   └─md0                             599.6G   root disk brw-rw----
sdd
├─sdd1                               linux_raid_member 1.2   proxy:0 bcd3f9bf-4aca-1508-888f-928db389cbe7  200G    root disk brw-rw----
│   └─md0                             599.6G   root disk brw-rw----
sde
├─sde1                               linux_raid_member 1.2   proxy:0 bcd3f9bf-4aca-1508-888f-928db389cbe7  200G    root disk brw-rw----
│   └─md0                             599.6G   root disk brw-rw----

```

Ahora, ya que tenemos creado el RAID5 con 4 dispositivos, el nuevo objetivo es crear 2 particiones con el dispositivo del arreglo “md0”. ¿Por qué dos particiones? Una partición para almacenar los datos del cache del servidor proxy y la otra partición para almacenar los registros del mismo servicio. Para cumplir con éste objetivo, vamos a necesitar de configurar **LVM** en el dispositivo “md0”.

Creamos el **Volumen Físico (Physical Volume)** para el dispositivo “md0”.

```

sysadmin@proxy: ~$ sudo pvcreate /dev/md0
[sudo] password for sysadmin:
Physical volume "/dev/md0" successfully created.

```

Luego podemos validar el PV del dispositivo “md0”.

```

sysadmin@proxy: ~$ sudo pvscan
PV /dev/sda5   VG vg_proxy   lvm2 [119.04 GiB / 0   free]
PV /dev/md0    lvm2 [<599.62 GiB]
Total: 2 [718.66 GiB] / in use: 1 [119.04 GiB] / in no VG: 1 [<599.62 GiB]
sysadmin@proxy: ~$ sudo pvs
PV          VG          Fmt Attr PSize  PFree
/dev/md0    lvm2 ---  <599.62g <599.62g
/dev/sda5   vg_proxy lvm2 a--  119.04g  0
sysadmin@proxy: ~$ sudo pvdisplay /dev/md0
"/dev/md0" is a new physical volume of "<599.62 GiB"
--- NEW Physical volume ---
PV Name     /dev/md0
VG Name
PV Size    <599.62 GiB
Allocatable NO
PE Size    0
Total PE   0
Free PE    0
Allocated PE 0
PV UUID    51GZFI-lyej-905N-nQDf-qR14-ljMz-2Wn322

```

Ahora vamos a crear un **Grupo de Volumen (Volume Group)** con el nombre “*vg_cache_logs*” para el dispositivo “*md0*”.

```
sysadmin@proxy:~$ sudo vgcreate vg_cache_logs /dev/md0
Volume group "vg_cache_logs" successfully created
```

Luego podemos validar el VG del dispositivo “*md0*”.

```
sysadmin@proxy:~$ sudo vgs
VG                #PV #LV #SN Attr   VSize   VFree
vg_cache_logs     1  0  0 wz--n- <599.62g <599.62g
vg_proxy          1  2  0 wz--n- 119.04g  0
sysadmin@proxy:~$ sudo vgscan
Found volume group "vg proxy" using metadata type lvm2
Found volume group "vg_cache_logs" using metadata type lvm2
sysadmin@proxy:~$ sudo vgsdisplay vg_cache_logs
--- Volume group ---
VG Name           vg_cache_logs
System ID
Format            lvm2
Metadata Areas    1
Metadata Sequence No 1
VG Access         read/write
VG Status         resizable
MAX LV            0
Cur LV           0
Open LV           0
Max PV            0
Cur PV           1
Act PV            1
VG Size           <599.62 GiB
PE Size           4.00 MiB
Total PE          153502
Alloc PE / Size   0 / 0
Free PE / Size    153502 / <599.62 GiB
VG UUID           n4JpTH-R6AG-Aad0-aXcy-1Aum-81Ep-YC1zv2
```

Ahora, para crear las 2 particiones, debemos crear dos Volúmenes Lógicos (Logical Volumes), un VL para cada partición. Para esto, debemos dividir la capacidad de almacenamiento disponible del arreglo nivel 5 de 599.62 GB entre dos. Para el primer volumen lógico lo nombraremos “*vl_cache*” con una capacidad de 299.81 GB y para el segundo volumen lógico lo nombraremos “*vl_logs*” con la mitad restante de 299.80 GB.

Crear el primer volumen lógico para el Cache del Squid Proxy.

```
sysadmin@proxy:~$ sudo lvcreate -L 299.81G -n vl_cache vg_cache_logs
Rounding up size to full physical extent 299.81 GiB
Logical volume "vl_cache" created.
```

Crear el segundo volumen lógico para los Registros del Squid Proxy.

```
sysadmin@proxy:~$ sudo lvcreate -L 299.80G -n vl_logs vg_cache_logs
Rounding up size to full physical extent 299.80 GiB
Logical volume "vl_logs" created.
```

Ahora podemos validar el estado de ambos volúmenes lógicos.

```
sysadmin@proxy:~$ sudo lvs
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
vl_cache vg_cache_logs -wi-a----- 299.81g
vl_logs vg_cache_logs -wi-a----- 299.80g
vl_swap vg_proxy -wi-ao---- 3.81g
vl_system vg_proxy -wi-ao---- 115.23g
sysadmin@proxy:~$ sudo lvscan
ACTIVE '/dev/vg_proxy/vl_swap' [3.81 GiB] inherit
ACTIVE '/dev/vg_proxy/vl_system' [115.23 GiB] inherit
ACTIVE '/dev/vg_cache_logs/vl_cache' [299.81 GiB] inherit
ACTIVE '/dev/vg_cache_logs/vl_logs' [299.80 GiB] inherit
```

```
sysadmin@proxy:~$ sudo lvdisplay /dev/vg_cache_logs/vl_cache
--- Logical volume ---
LV Path                /dev/vg_cache_logs/vl_cache
LV Name                 vl_cache
VG Name                 vg_cache_logs
LV UUID                 cP3Qc1-3ev6-qDDT-9nap-zAmB-YjLY-9BQceT
LV Write Access         read/write
LV Creation host, time proxy, 2022-09-16 18:16:42 -0600
LV Status                available
# open                  0
LV Size                 299.81 GiB
Current LE              76752
Segments                1
Allocation              inherit
Read ahead sectors      auto
- currently set to     6144
Block device            254:2

sysadmin@proxy:~$ sudo lvdisplay /dev/vg_cache_logs/vl_logs
--- Logical volume ---
LV Path                /dev/vg_cache_logs/vl_logs
LV Name                 vl_logs
VG Name                 vg_cache_logs
LV UUID                 htpJxJ-Ik7Z-LdBf-0nOz-yzNf-pKK7-3uloJu
LV Write Access         read/write
LV Creation host, time proxy, 2022-09-16 18:17:03 -0600
LV Status                available
# open                  0
LV Size                 299.80 GiB
Current LE              76749
Segments                1
Allocation              inherit
Read ahead sectors      auto
- currently set to     6144
Block device            254:3
```

Ahora lo que debemos hacer es formatear las particiones de cada volumen lógico y para esto vamos a utilizar el sistema de archivo **XFS** para ambas particiones con la finalidad de ofrecer mayor rendimiento y velocidad de cache a la hora de navegar a través del Servidor Proxy, y también a la hora de registrar los filtros de toda navegación en la Internet. Otra ventaja del XFS es que éste soporta tamaños de discos arriba de 16 Terabytes para sistemas de 64 bits.

Para poder utilizar el sistema de archivo XFS debemos instalar la herramienta para el manejo de dicho sistema de archivo. **Si ya tienen instalado XFS, pueden saltarse todo esto y empezar a formatear los volúmenes lógicos con dicho sistema de archivo.**

```
sysadmin@proxy:~$ sudo apt install xfsprogs xfsdump -y
```

Una vez instalado XFS, debemos validar que XFS fue cargado correctamente en el módulo del Kernel del sistema.

```
sysadmin@proxy:~$ sudo grep xfs /proc/filesystems
xfs
sysadmin@proxy:~$ sudo lsmod | egrep xfs
xfs                1802240  0
libcrc32c          16384  3 btrfs,xfs,raid456
sysadmin@proxy:~$ sudo modinfo xfs
filename:           /lib/modules/5.10.0-18-amd64/kernel/fs/xfs/xfs.ko
license:            GPL
description:        SGI XFS with ACLs, security attributes, realtime, quota, no debug enabled
author:             Silicon Graphics, Inc.
alias:              fs-xfs
depends:             libcrc32c
retpoline:         Y
intree:            Y
name:              xfs
vermagic:          5.10.0-18-amd64 SMP mod_unload modversions
sig_id:            PKCS#7
signer:            Debian Secure Boot CA
sig_key:           32:A0:28:7F:84:1A:03:6F:A3:93:C1:E0:65:C4:3A:E6:B2:42:26:43
sig_hashalgo:     sha256
signature:         2A:25:AA:B9:56:B5:7C:E0:08:5D:B9:90:74:64:E7:F9:FC:82:0C:C8:
79:09:18:AA:2C:62:16:62:65:6C:F7:32:B3:05:E2:56:B6:4C:EE:ED:
39:0A:CE:26:F6:EC:DD:BC:0F:18:FB:7C:B1:77:08:9A:51:AE:38:19:
F5:C3:04:BD:37:71:74:E3:EC:5C:40:41:0F:BE:A0:63:C9:66:4D:08:
75:C1:08:E6:96:58:DD:CB:54:21:24:27:BA:DC:2C:92:AD:CD:41:D2:
35:0A:47:13:80:8F:0D:98:B7:64:38:3F:67:7C:6E:FD:24:4E:8B:07:
57:57:4A:22:B3:C7:5E:6B:82:8D:27:13:12:3D:8F:0F:A6:14:3C:78:
5A:C7:8A:32:35:E5:20:3B:42:46:AF:11:F1:F9:25:84:F2:89:2B:7B:
51:9A:CD:74:FF:76:7A:0E:1E:68:70:01:D0:3E:70:81:3F:68:72:CC:
46:1C:5A:68:1B:61:08:AF:67:53:93:52:02:72:26:82:62:7F:1F:37:
D4:5B:7F:10:FD:63:6D:F0:7B:37:4A:0F:5A:6B:1C:46:2D:D7:93:0E:
E1:A0:56:91:C0:0D:F8:CA:F2:F1:62:60:97:A1:FB:86:7A:F5:C2:07:
FE:12:B4:A5:1F:5E:E6:11:CF:B3:9D:64:ED:57:2A:27
```

Empezamos a formatear los volúmenes lógicos con XFS.

```
sysadmin@proxy:~$ sudo mkfs.xfs /dev/vg_cache_logs/vl_cache
log stripe unit (524288 bytes) is too large (maximum is 256KiB)
log stripe unit adjusted to 32KiB
meta-data=/dev/vg_cache_logs/vl_cache isize=512    agcount=16, agsize=4912000 blks
         =                       sectsz=512    attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=0
data      =                       bsize=4096   blocks=78592000, imaxpct=25
         =                       sunit=128    swidth=384 blks
naming    =version 2              bsize=4096   ascii-ci=0, ftype=1
log        =internal log         bsize=4096   blocks=38376, version=2
         =                       sectsz=512    sunit=8 blks, lazy-count=1
realtime  =none                  extsz=4096   blocks=0, rtextents=0
sysadmin@proxy:~$ sudo mkfs.xfs /dev/vg_cache_logs/vl_logs
log stripe unit (524288 bytes) is too large (maximum is 256KiB)
log stripe unit adjusted to 32KiB
meta-data=/dev/vg_cache_logs/vl_logs isize=512    agcount=16, agsize=4912000 blks
         =                       sectsz=512    attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=0
data      =                       bsize=4096   blocks=78590976, imaxpct=25
         =                       sunit=128    swidth=384 blks
naming    =version 2              bsize=4096   ascii-ci=0, ftype=1
log        =internal log         bsize=4096   blocks=38376, version=2
         =                       sectsz=512    sunit=8 blks, lazy-count=1
realtime  =none                  extsz=4096   blocks=0, rtextents=0
```

Ahora debemos crear los Puntos de Montajes para acceder a los sistemas de archivos XFS, un punto de montaje para el Cache y otro punto de montaje para los Registros del Proxy. Para esto debemos crear los directorios para ambos puntos de montajes. El directorio para el cache del Squid Proxy es el **“/var/spool/squid”** y el directorio para los registros del Squid Proxy es el **“/var/log/squid”**. Entonces, antes de crear los puntos de montajes, debemos crear dichos directorios de la siguiente manera.

```
sysadmin@proxy:~$ sudo mkdir /var/spool/squid
sysadmin@proxy:~$ sudo mkdir /var/log/squid
sysadmin@proxy:~$ ls -l /var/spool/ | egrep squid
drwxr-xr-x 2 root      root          4096 sep 16 19:45 squid
sysadmin@proxy:~$ ls -l /var/log/ | egrep squid
drwxr-xr-x 2 root      root          4096 sep 16 19:45 squid
```

Ahora si comenzamos a montar los volúmenes lógicos creados previamente.

```
sysadmin@proxy:~$ sudo mount /dev/mapper/vg_cache_logs-vl_cache /var/spool/squid
sysadmin@proxy:~$ sudo mount /dev/mapper/vg_cache_logs-vl_logs /var/log/squid
```

Si se fijan detalladamente, estamos montando los volúmenes lógicos asociados con su respectivo VG mediante el sub-directorio **“mapper”**.

Validamos los puntos de montajes con “df -h”.

```

sysadmin@proxy: ~$ df -h
S.ficheros          Tamaño Usados  Disp Uso% Montado en
udev                3.9G      0    3.9G  0% /dev
tmpfs               793M      660K    792M  1% /run
/dev/mapper/vg_proxy-vl_system 113G      1.5G   106G  2% /
tmpfs               3.9G      0    3.9G  0% /dev/shm
tmpfs               5.0M      0    5.0M  0% /run/lock
/dev/sda1           943M      86M    831M 10% /boot
tmpfs               793M      0    793M  0% /run/user/1000
/dev/mapper/vg_cache_logs-vl_cache 300G      2.2G   298G  1% /var/spool/squid
/dev/mapper/vg_cache_logs-vl_logs  300G      2.2G   298G  1% /var/log/squid
  
```

Si se fijan detalladamente, cada punto de montaje tiene una capacidad total de 300 GB y una capacidad disponible de 298 GB.

También validamos los puntos de montajes de los volúmenes lógicos con su VG asociados con la unidad “md0” del RAID5 con “lsblk -fm”.

```

sysadmin@proxy: ~$ lsblk -fm
NAME                                FSTYPE     FSVER  LABEL  UUID                                FSAVAIL  FSUSEX%  MOUNTPOINT  SIZE  OWNER  GROUP  MODE
sda
├─sda1                               ext4        1.0    791ec00e-0602-4b77-80ec-344b50da953d  830.9M   9% /boot      976M  root   disk   brw-rw----
├─sda2
├─sda5                               LVM2_member LVM2 001  shwVP6-Xwr3-B6d6-uhBL-qovP-a1A4-VCqgh4 119G     root     disk   brw-rw----
│   └─vg_proxy-vl_swap                swap        1      [SWAP]                                3.8G    root     disk   brw-rw----
│   └─vg_proxy-vl_system              ext4        1.0    387756be-640e-4a6d-b136-e06658fe2b90  105.7G  1% /          115.2G root     disk   brw-rw----
sdb
├─sdb1                               linux_raid_member 1.2    proxy:0 bcd3f9bf-4aca-1508-888f-928db389cbe7 200G    root     disk   brw-rw----
│   └─md0                             LVM2_member  LVM2 001  51GZFI-1yej-905N-nQDf-qR14-1jMz-2Wn322 599.6G  root     disk   brw-rw----
│       └─vg_cache_logs-vl_cache      xfs        297.5G  1% /var/spool/squid 299.8G  root     disk   brw-rw----
│       └─vg_cache_logs-vl_logs      xfs        297.5G  1% /var/log/squid  299.8G  root     disk   brw-rw----
sdc
├─sdc1                               linux_raid_member 1.2    proxy:0 bcd3f9bf-4aca-1508-888f-928db389cbe7 200G    root     disk   brw-rw----
│   └─md0                             LVM2_member  LVM2 001  51GZFI-1yej-905N-nQDf-qR14-1jMz-2Wn322 599.6G  root     disk   brw-rw----
│       └─vg_cache_logs-vl_cache      xfs        297.5G  1% /var/spool/squid 299.8G  root     disk   brw-rw----
│       └─vg_cache_logs-vl_logs      xfs        297.5G  1% /var/log/squid  299.8G  root     disk   brw-rw----
sdd
├─sdd1                               linux_raid_member 1.2    proxy:0 bcd3f9bf-4aca-1508-888f-928db389cbe7 200G    root     disk   brw-rw----
│   └─md0                             LVM2_member  LVM2 001  51GZFI-1yej-905N-nQDf-qR14-1jMz-2Wn322 599.6G  root     disk   brw-rw----
│       └─vg_cache_logs-vl_cache      xfs        297.5G  1% /var/spool/squid 299.8G  root     disk   brw-rw----
│       └─vg_cache_logs-vl_logs      xfs        297.5G  1% /var/log/squid  299.8G  root     disk   brw-rw----
sde
├─sde1                               linux_raid_member 1.2    proxy:0 bcd3f9bf-4aca-1508-888f-928db389cbe7 200G    root     disk   brw-rw----
│   └─md0                             LVM2_member  LVM2 001  51GZFI-1yej-905N-nQDf-qR14-1jMz-2Wn322 599.6G  root     disk   brw-rw----
│       └─vg_cache_logs-vl_cache      xfs        297.5G  1% /var/spool/squid 299.8G  root     disk   brw-rw----
│       └─vg_cache_logs-vl_logs      xfs        297.5G  1% /var/log/squid  299.8G  root     disk   brw-rw----
  
```

Ahora que tenemos montado **TODO** correctamente, existe un grave problema. El problema es que el punto de montaje es temporal, es decir, se perderán cuando el sistema del servidor se reinicia y esto puede provocar pérdida de datos. Para resolver esto debemos hacer que el punto de montaje sea persistente (permanentemente) y para esto debemos montar los volúmenes lógicos y sus unidades con el archivo “fstab” ubicado en “/etc/fstab” de la siguiente forma.

```

# <file system> <mount point> <type> <options> <dump> <pass>
/dev/mapper/vg_proxy-vl_system / ext4 errors=remount-ro 0 1
# /boot was on /dev/sda1 during installation
UUID=791ec00e-0602-4b77-80ec-344b50da953d /boot ext4 defaults 0 2
/dev/mapper/vg_proxy-vl_swap none swap SW 0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
/dev/mapper/vg_cache_logs-vl_cache /var/spool/squid xfs defaults 0 3
/dev/mapper/vg_cache_logs-vl_logs /var/log/squid xfs defaults 0 4
  
```