

# LVM Práctico para Proxmox y OVH

## Introducción

Es un tip que uso para **mis necesidades**. Entre otras, no uso **ZSF** por la carga que supone para el sistema, y porque suplo sus bondades por decisión propia con otros mecanismo de salvaguarda.

“ Cada modelo de máquina de OVH y cada instalación deja diferencias sutiles que pueden hacer que uno tenga que modificar los datos de particiones, discos, interfaces de red.

## LVM

En una instalación sin uso de RAID por software del instalador tenemos en la versión del instalador de OVH que instala 7.2-14 de Proxmox un **volumen** llamado **vg** en el disco primario

```
pvs
PV          VG Fmt Attr PSize PFree
/dev/nvme0n1p5 vg lvm2 a-- 1.72t  0
lvscan
ACTIVE      '/dev/vg/data' [1.72 TiB] inherit
```

Sobre el cual ha creado un **volume** llamado **vg** y un **volumen lógico** llamado **data** con el tamaño máximo

“ Mal por todo, porque asigna todo el espacio y no permite hacer snapshots.

Este modelo no me sirve pues yo nunca uso virtualización por **OpenVZ** así que hay que eliminar y reconstruir

# Copia de la partición

## Parada de proxmox

Versiones anteriores a la 7/8

```
systemctl stop pve-cluster pvedaemon pveproxy pvestatsd
```

Versiones 7/8

```
systemctl stop pve-cluster pvedaemon pveproxy
```

## Copia de /var/lib/vz

```
mkdir /old_vz  
rsync -avv --progress /var/lib/vz/ /old_vz/  
umount /var/lib/vz
```

Debemos por seguridad editar `/etc/fstab` para eliminar o comentar el punto de montaje de `/var/lib/vz/` por si ocurriera algún reinicio por despiste

## Eliminación del volumen

```
lvchange -an /dev/vg/data  
lvremove /dev/vg/data  
Logical volume "data" successfully removed
```

## Creación de un volumen menor

Por si acaso, un día queremos una instancia tipo **OpenVZ** crearemos un volumen lógico pequeño pero suficiente para que esté ahí, por si un día lo queremos usar.

```
lvcreate -n data --size 20GB vg  
WARNING: ext4 signature detected on /dev/vg/data at offset 1080. Wipe it? [y/n]: y  
Wiping ext4 signature on /dev/vg/data.  
Logical volume "data" created.  
  
pvscan
```

```
PV /dev/nvme0n1p5 VG vg          lvm2 [1.72 TiB / <1.71 TiB free]
```

```
Total: 1 [1.72 TiB] / in use: 1 [1.72 TiB] / in no VG: 0 [0  ]
```

```
mkfs.xfs /dev/vg/data
```

```
meta-data=/dev/vg/data          isize=512  agcount=16, agsize=327680 blks
```

```
      =                sectsz=4096  attr=2, projid32bit=1
```

```
      =                crc=1      finobt=1, sparse=1, rmapbt=0
```

```
      =                reflink=1   bigtime=0
```

```
data      =                bsize=4096  blocks=5242880, imaxpct=25
```

```
      =                sunit=32   swidth=32 blks
```

```
naming    =version 2          bsize=4096  ascii-ci=0, ftype=1
```

```
log       =internal log      bsize=4096  blocks=2560, version=2
```

```
      =                sectsz=4096  sunit=1 blks, lazy-count=1
```

```
realtime  =none              extsz=4096  blocks=0, rtextents=0
```

```
Discarding blocks...Done.
```

## Obtención del UUID

```
blkid
```

```
...
```

```
/dev/mapper/vg-data: UUID="c0ede30d-f8a6-4897-a392-39c9dfc542a6" BLOCK_SIZE="4096" TYPE="xfs"
```

## Edición de fstab

```
UUID=c0ede30d-f8a6-4897-a392-39c9dfc542a6 /var/lib/vz xfs defaults 0 0
```

## Restauración de ficheros del directorio

```
rsync -avv --progress /old_vz/ /var/lib/vz/
```

```
systemctl restart pve-cluster pvedaemon pveproxy pvestatsd
```

# Creación de volúmenes LVM

En mi caso, y en este en particular, no deseo ampliar el volumen, sino que prefiero tener 3 volúmenes independientes, por el tipo de máquina que uso y el uso que se va a dar, ya que la velocidad y cargas en el proceso, me son más interesante con 3 volúmenes, que con uno, a la par que la posibilidad de una rotura de LVM es mayor y un riesgo que llevo años sin querer sufrirlo (que alguna vez lo he sufrido) y en los últimos 8 años, no he tenido ningún percance con varias

máquinas con este aprovechamiento.

# Reconocimiento de discos

disk -l

Disk /dev/nvme1n1: 1.75 TiB, 1920383410176 bytes, 3750748848 sectors

Disk model: SAMSUNG MZQL21T9HCJR-00A07

Units: sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 4096 bytes

I/O size (minimum/optimal): 131072 bytes / 131072 bytes

Disk /dev/nvme2n1: 1.75 TiB, 1920383410176 bytes, 3750748848 sectors

Disk model: SAMSUNG MZQL21T9HCJR-00A07

Units: sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 4096 bytes

I/O size (minimum/optimal): 131072 bytes / 131072 bytes

Disk /dev/nvme0n1: 1.75 TiB, 1920383410176 bytes, 3750748848 sectors

Disk model: SAMSUNG MZQL21T9HCJR-00A07

Units: sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 4096 bytes

I/O size (minimum/optimal): 131072 bytes / 131072 bytes

Disklabel type: gpt

Disk identifier: 991EADF9-826A-4A35-B37C-5A1C3D255A5D

Device	Start	End	Sectors	Size	Type
/dev/nvme0n1p1	2048	1048575	1046528	511M	EFI System
/dev/nvme0n1p2	1048576	3145727	2097152	1G	Linux filesystem
/dev/nvme0n1p3	3145728	45088767	41943040	20G	Linux filesystem
/dev/nvme0n1p4	45088768	47185919	2097152	1G	Linux filesystem
/dev/nvme0n1p5	47185920	3750739967	3703554048	1.7T	Linux LVM
/dev/nvme0n1p6	3750744752	3750748814	4063	2M	Linux filesystem

Partition 6 does not start on physical sector boundary. // No importante

Disk /dev/mapper/vg-data: 20 GiB, 21474836480 bytes, 41943040 sectors

Units: sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 4096 bytes

I/O size (minimum/optimal): 131072 bytes / 131072 bytes

Tenemos pues los discos `nvme1n1` y `nvme2n1` disponibles

# Creación de los grupos de volúmenes

Como uso una dinámica de nombres para así hacer más fácil las migraciones entre máquinas Proxmox, haré lo mismo.

“ Es importante crear los volúmenes de almacenamiento en **particiones** y no sobre el disco completo.

## Partición

Usare mi [Parted mejor que fdisk](#)

Creamos las etiquetas

```
parted -s /dev/nvme1n1 mklabel gpt
parted -s /dev/nvme2n1 mklabel gpt
```

Creamos las particiones

```
parted -s /dev/nvme1n1 mkpart primary 0% 100%
parted -s /dev/nvme2n1 mkpart primary 0% 100%
```

## Creación de los volúmenes

```
vgcreate lvm /dev/nvme1n1p1
Physical volume "/dev/nvme1n1p1" successfully created.
Volume group "lvm" successfully created
```

```
vgcreate lvm2 /dev/nvme2n1p1
Physical volume "/dev/nvme2n1p1" successfully created.
Volume group "lvm2" successfully created
```

```
pvscan
```

```
PV /dev/nvme1n1p1 VG lvm          lvm2 [<1.75 TiB / <1.75 TiB free]
```

```
PV /dev/nvme2n1p1 VG lvm2         lvm2 [<1.75 TiB / <1.75 TiB free]
```

```
PV /dev/nvme0n1p5 VG vg           lvm2 [1.72 TiB / <1.71 TiB free]
```

```
Total: 3 [<5.22 TiB] / in use: 3 [<5.22 TiB] / in no VG: 0 [0  ]
```

# Final

A partir de aquí, ya tenemos nuestra estructura deseada en LVM para Proxmox.

# Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

---

Revision #4

Created 19 November 2022 08:45:44 by Abkrim

Updated 13 December 2023 13:07:26 by Abkrim