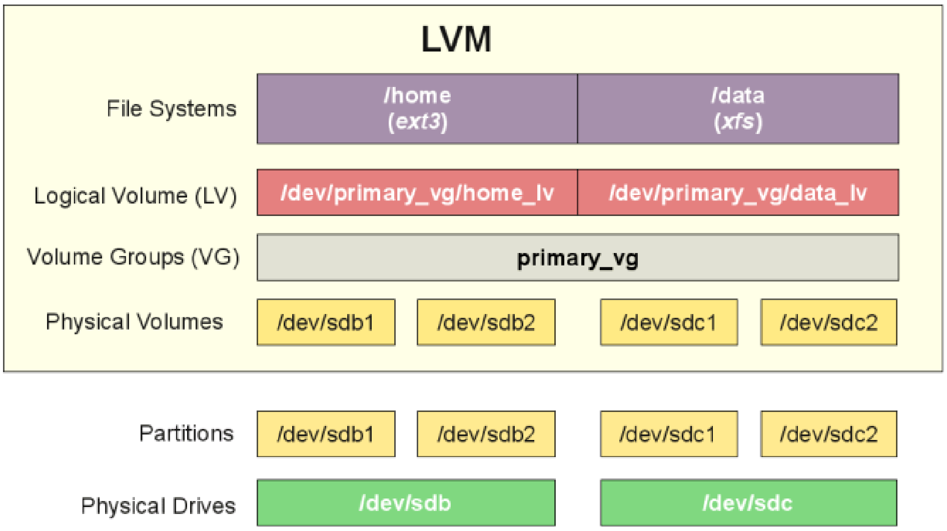
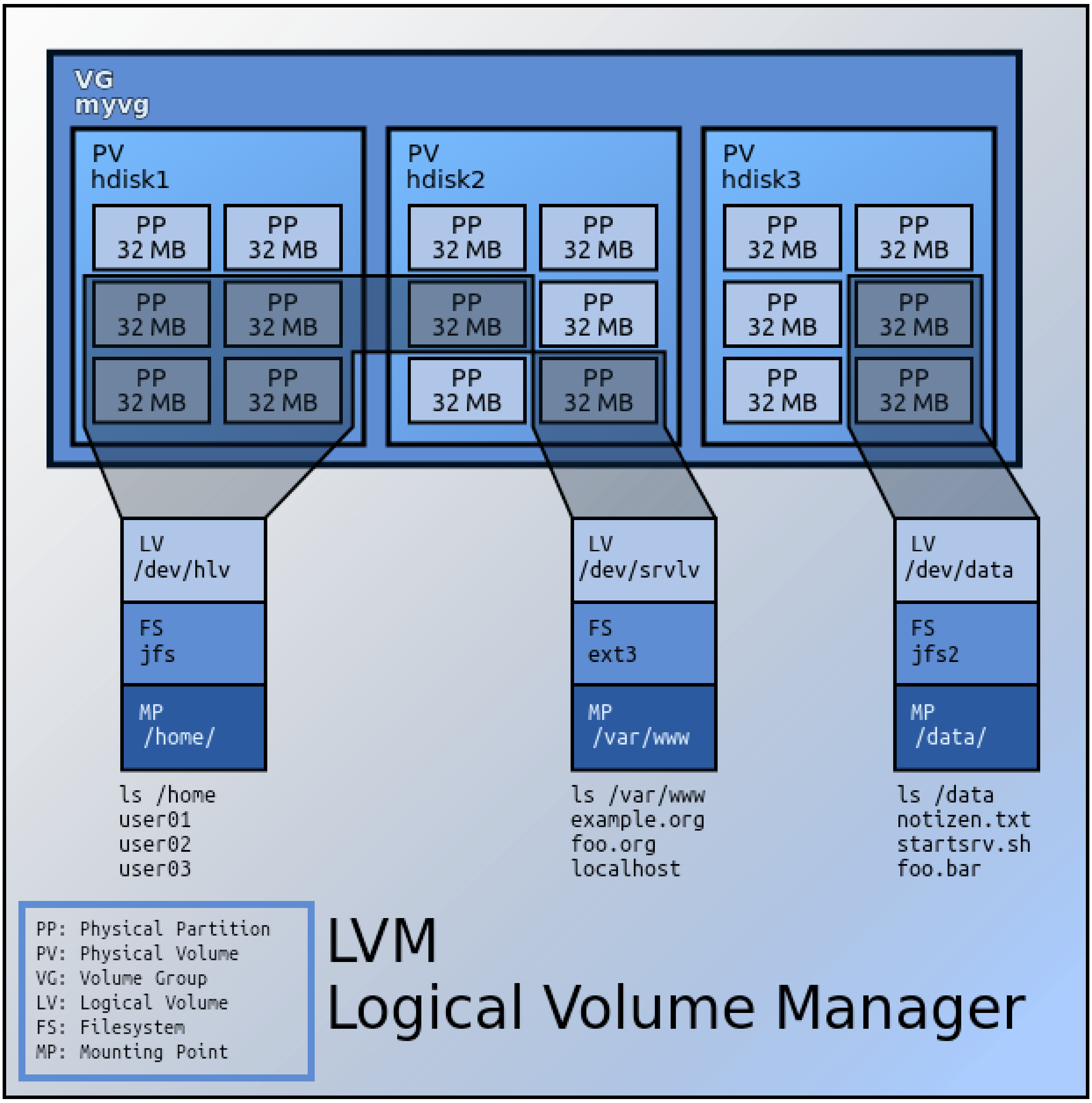
How to mount LVM partition volume on Linux

Author : admin



(**LVM**) = Logical Volume Manager is a device mapper offering **logical volume management** for the Linux kernel. Virtually all modern GNU / Linux distributions has support for it and using LVM is used among almost all **Hosting Providers** on (**dedicated**) backend **physical and Virtual XEN / VMWare etc.** servers because it provides the ability to merge a number of disks into **virtual volumes** (for example you have a number of SSD Hard Drives on a server that are under a separate ***/dev/sda1 /dev/sda2 /dev/sdb3 /dev/sdb4*** etc. and you want all the HDDs to appear as a single file system this is managed by ***Linux*** **LVM**.



Picture sources Wikipedia

The use of **LVM** is somewhat similar to **RAID 0 disk arrays,** where the good about it it allows the removal and addition of hard disks in real time (broken hard disks) on servers to be replaced without service downtime as well as dynamic **HDD volume** resizal is possible. **LVM** allows also relatively easy encryption of multiple HDD volumes
with single password.

Discs can be organized in **volume groups (so lets say 2 of the server Attached** conventional **Hard Disks, SCSI** or **SSDs** can be attached to LVM1 and another **3 Hard Drives** could be attached to **LVM2**

group etc.

LVM has been an integral part of Linux kernel since 1998.

lvm is available for install via **apt, yum, dhf** etc. under a package called **lvm2**, so to install it on **Debian / Ubuntu Fedora** Linux (if it is not already installed on the servers with).

- Install LVM2 On Debian / Ubuntu

```
debian:~# apt-get install --yes lvm2
```

- Install LVM2 on Fedora / CentOS (Redhat RPM based distros)

```
[root@centos ~]:# yum install -y lvm2
```

or

```
[root@fedora ~]:#  dhf install -y lvm2
```

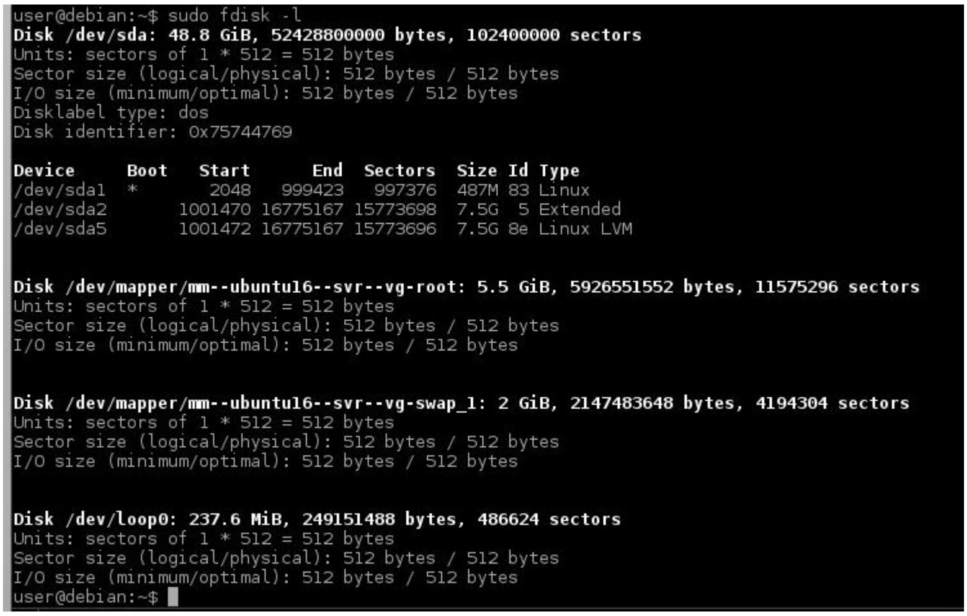
I. Mounting LVM2 on Linux server after broken DISK change part of a LVM Volume

For example the ***HDD failed - due to bad sectors and physical HDD head damage damage*** - *the easiest way to figure that out if the server is running* ***smartd*** *or via a simple HDD test check from* ***BIOS*** *(* as the ROOT partition is on a LVM it fails to boot properly. You have changed the broken **HDD** with a brand new and you need to remount the LVM either physically on the server console or remotely via some kind of **BIOS KVM** interface).

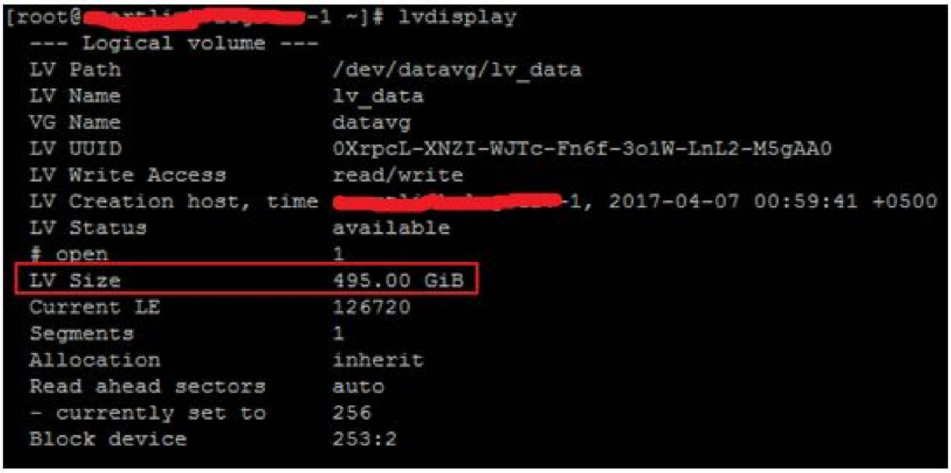
In my experience working for Santrex this was a common sysadmin job, as many of the **Virtual Client servers** as well as others irons situated in various **DataCenters,** were occasionally failing to boot and the monitoring system was reporting about the issues and we had to promptly react and bring the servers up.

Here is shortly how we managed to re-mount the **LVM after the SSDs / HDDS** were substituted:

1.1. Execute fdisk, vgscan / lvdisplay command

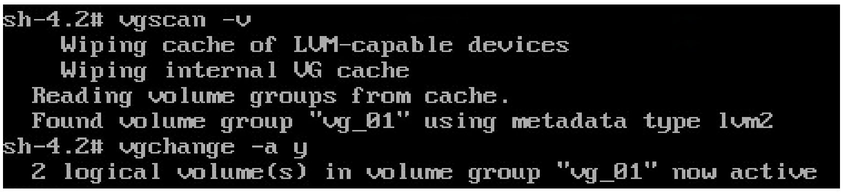


```
user@debian:~$ sudo fdisk -l
Disk /dev/sda: 48.8 GiB, 52428800000 bytes, 102400000 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x75744769

Device     Boot   Start      End  Sectors  Size Id Type
/dev/sda1  *       2048   999423   997376  487M 83 Linux
/dev/sda2       1001470 16775167 15773698  7.5G  5 Extended
/dev/sda5       1001472 16775167 15773696  7.5G 8e Linux LVM


Disk /dev/mapper/mm--ubuntu16--svr--vg-root: 5.5 GiB, 5926551552 bytes, 11575296 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/mapper/mm--ubuntu16--svr--vg-swap_1: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop0: 237.6 MiB, 249151488 bytes, 486624 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
user@debian:~$ █
```



```
[root@[redacted]-1 ~]# lvdisplay
  --- Logical volume ---
  LV Path                /dev/datavg/lv_data
  LV Name                lv_data
  VG Name                datavg
  LV UUID                0XrpcL-XNZI-WJTc-Fn6f-3o1W-LnL2-M5gAA0
  LV Write Access        read/write
  LV Creation host, time [redacted]-1, 2017-04-07 00:59:41 +0500
  LV Status              available
  # open                 1
  LV Size                495.00 GiB
  Current LE             126720
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:2
```

vgscan scans all supported LVM block devices in the system for VGs (Virtual Groups)



```
sh-4.2# vgscan -v
    Wiping cache of LVM-capable devices
    Wiping internal VG cache
  Reading volume groups from cache.
  Found volume group "vg_01" using metadata type lvm2
sh-4.2# vgchange -a y
  2 logical volume(s) in volume group "vg_01" now active
```

1.2. Next issue vgchange command to activate volume

```
vgchange -ay
```

1.3. Type *lvs* command to get information about logical volumes

```
lvs
```

1.4. Create a mount point using the *mkdir* command

That's because we wanted to check the LVM will get properly mounted on next server reboot).

1.5. Mount an LVM volume using

```
server:~# mount /dev/mapper/DEVICE /path/to/mount_point
```

1.6. To check the size of the LVM (mount points, mounted LVM /dev/names sizes and the amount of free space on each of them use)

```
server:~# df -T
```