

Logical Volume Manager (LVM)

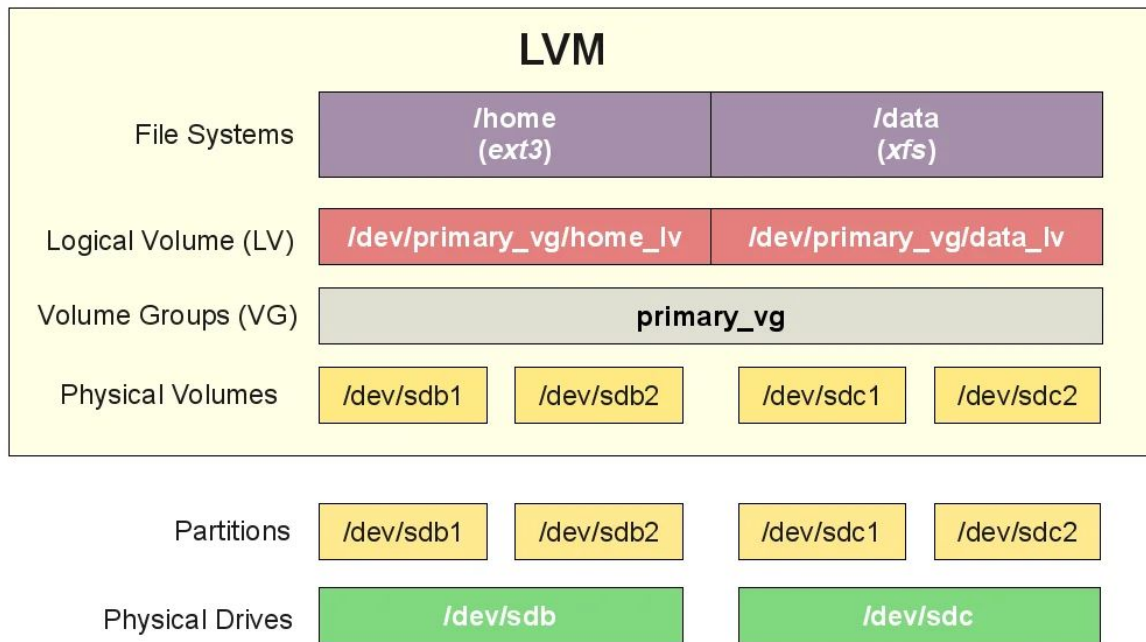
黃柏竣 (bjhuang)

What is LVM?

Logical Volume Manager

- A device mapper framework (/dev/mapper/)
 - Provides logical volume management for the Linux kernel
- Making disks/partitions a storage **pool**
 - Uses any number of disks as one big disk
 - Abstract storage and able to use “virtual partitions”
 - Able to resize volume easily
 - No need to care about partition precedence

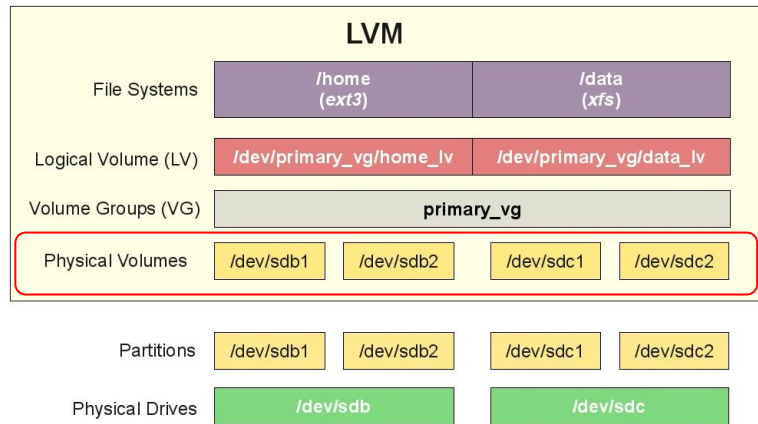
Architecture



Source: [Arch Linux Tutorial - How to configure LVM on Arch Linux 2017](#)

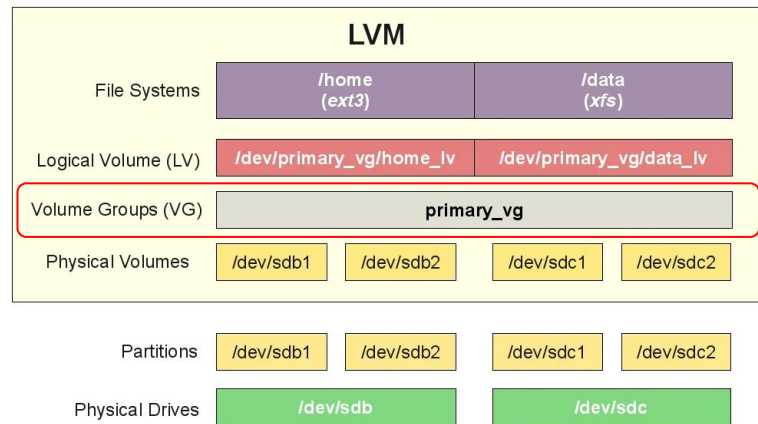
Physical Volume (PV)

- Base storage of LVM
 - Hard disk, Hard disk partition, Linux software RAID (/dev/mdXXX), ...
 - After creating, LVM-related header will be added to device(s)
 - `pvcreate <device>`
 - `pvcreate /dev/sdb1 /dev/sdb2 /dev/sdc1 /dev/sdc2`



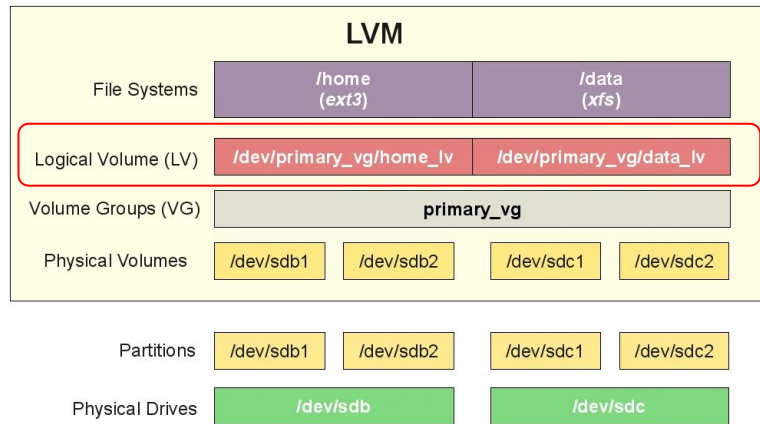
Volume Group (VG)

- Consists of one or several PV(s)
 - `vgcreate <VG name> <PV name> ...`
 - `vgextend <VG name> <PV name> ...`
 - `vgcreate primary_vg /dev/sdb1 /dev/sdb2 /dev/sdc1 /dev/sdc2`



Logical Volume (LV)

- LV sits inside a VG
 - Form a virtual partition
 - `lvcreate -l 100%FREE <VG name> [PV path] -n <LV name>`
 - `lvcreate -l 50%FREE primary_vg -n home_lv`
 - `lvcreate -L 50G <VG name> -n <LV name>`
 - `lvcreate -L 50G primary_vg -n data_lv`
- Paths
 - `/dev/<VG name>/<LV name>`
 - `/dev/mapper/<VG name>-<LV name>`

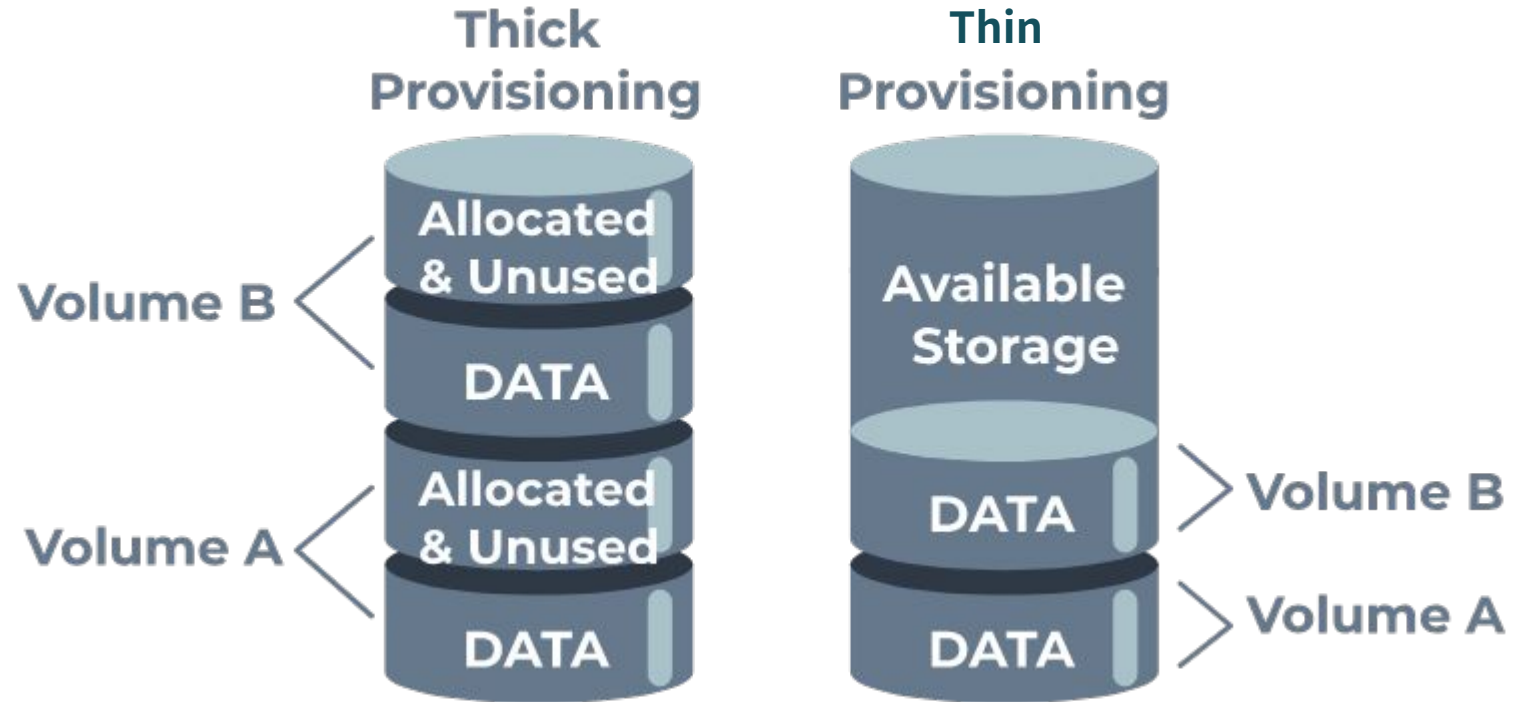


Physical Extent (PE) / Logical Extent (LE)

- PE
 - Smallest contiguous extent in the PV that can be assigned to a LV
 - default is 4MB
 - PEs are allocated from a VG for a LV
 - PV size must be a multiple of PE
 - `pvdisplay` and `vgdisplay` can show PE size and number
- LE
 - Similar to PE, but at Logical Volume level
 - Same size as PE
 - `lvdisplay` can show LE number

LVM thin provisioning

- Useful for virtual machine host
- Two kinds of LV for thin provision
 - Thin Pool
 - Consists of 2 hidden component LVs: metadata & data
 - `lvcreate --type thin-pool -n <pool name> -L <Size> <VG name>`
 - Thin LV
 - Resides in a thin pool
 - `lvcreate --type thin -n <LV name> -V <VirtualSize> --thinpool <pool name> <VG name>`
 - Virtual size means the upper bound size of thin LV

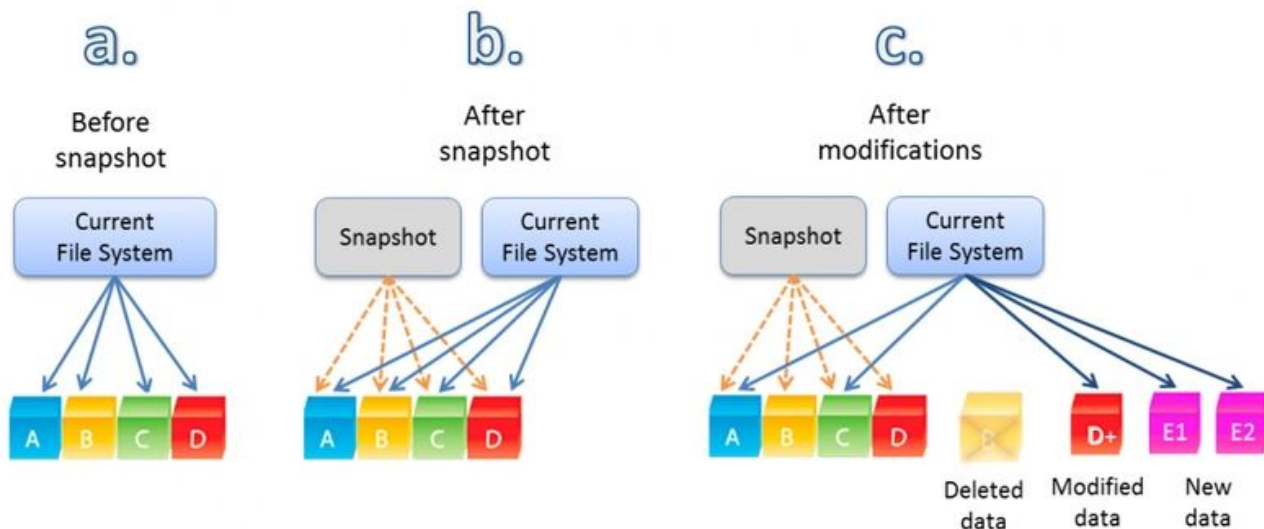


Source: [What is thin provisioning \(TP\)? | DiskInternals](#)

Snapshot

- LVM supports CoW (Copy-on-Write) snapshots
 - Fast because it doesn't copy data (just pointers initially)
 - Save space
 - Not backup!
- Create LVM snapshot for LV
 - `lvcreate -L <SIZE> --snapshot --name <snapshot LV name> <target LV name>`
 - Size is usually set about 20-30% of target LV storage
- Snapshot rollback
 - Unmount target LV, then run the following command:
 - `lvconvert --merge <snapshot LV name>`

Copy on write (CoW)



Source: [Mastering LVM Snapshots: Efficient Data Protection with Copy-on-Write \(CoW\)](#)

When to use LVM?

- Multiple disks on the machine
- One disk at present, but plan to extend more
- You don't want to manage disk partition precedence
- Not sure about the partition size required for different mountpoints/applications
 - LVM is also useful when using single disk

How to use LVM?

Create from bottom to top

1. Create PV(s)
2. Create VG with previous PV
3. Create LV with from previous VG
4. Make desired filesystem (ext4, btrfs, ...)
5. Mount LV to desired mount point

Examples

- Extend disk size for rootfs
 - Add disk for data mount point
 - Remove disk
-
- To ensure data consistency, **backup** the original data before any examples.

Extend disk size for rootfs

1. `echo 1 > /sys/block/<disk>/device/rescan` if disk size doesn't fit after revising VM's config
2. Use `fdisk` command to create new partition for the space
3. `pvcreate` to add partition(s) into PV
4. `vgextend` for adding new PV to storage pool (existing VG)
5. `lvextend` for adding space to LV
6. `resize2fs <device>`

Extend disk (lsblk)

Using lsblk to check all block devices (disks, CD-ROMs, ...)

```
root@trixws:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                                 8:0    0   48G  0 disk
├─sda1                             8:1    0   976M  0 part /boot/efi
├─sda2                             8:2    0   977M  0 part /boot
├─sda3                             8:3    0    24G  0 part
└─trixws--vg-root                 254:0    0  22.8G  0 lvm  /
   └─trixws--vg-swap_1            254:1    0    1.2G  0 lvm  [SWAP]
```


Extend disk (fdisk)

```
root@trixws:~# fdisk /dev/sda
```

```
Welcome to fdisk (util-linux 2.41).
```

```
Changes will remain in memory only, until you decide to write them.  
Be careful before using the write command.
```

```
This disk is currently in use - repartitioning is probably a bad idea.  
It's recommended to umount all file systems, and swapoff all swap  
partitions on this disk.
```

```
Command (m for help): m
```

Help:

GPT

M enter protective/hybrid MBR

Generic

d delete a partition
F list free unpartitioned space
l list known partition types
n add a new partition
p print the partition table
t change a partition type
v verify the partition table
i print information about a partition
e resize a partition
T discard (trim) sectors

Misc

m print this menu
x extra functionality (experts only)

Script

I load disk layout from sfdisk script file
O dump disk layout to sfdisk script file

Save & Exit

w write table to disk and exit
q quit without saving changes

Create a new label

g create a new empty GPT partition table
G create a new empty SGI (IRIX) partition table
o create a new empty MBR (DOS) partition table
s create a new empty Sun partition table

Extend disk (fdisk)

```
Command (m for help): p
```

```
Disk /dev/sda: 48 GiB, 51539607552 bytes, 100663296 sectors
```

```
Disk model: QEMU HARDDISK
```

```
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: gpt
```

```
Disk identifier: 840E2D44-4FB2-4488-B162-BEE185F878FF
```

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	2000895	1998848	976M	EFI System
/dev/sda2	2000896	4001791	2000896	977M	Linux filesystem
/dev/sda3	4001792	54333439	50331648	24G	Linux filesystem

```
Command (m for help): n
```

```
Partition number (4-128, default 4):
```

```
First sector (54333440-100663262, default 54333440):
```

```
Last sector, +/-sectors or +/-size{K,M,G,T,P} (54333440-100663262, default 100661247):
```

```
Created a new partition 4 of type 'Linux filesystem' and of size 22.1 GiB.
```

Extend disk (fdisk)

- Change GPT partition type
- Print before writing
- Write (Most important!)

```
Command (m for help): t
Partition number (1-4, default 4):
Partition type or alias (type L to list all): 44
```

```
Command (m for help): p
```

```
Disk /dev/sda: 48 GiB, 51539607552 bytes, 100663296 sectors
Disk model: QEMU HARDDISK
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: gpt
Disk identifier: 840E2D44-4FB2-4488-B162-BEE185F878FF
```

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	2000895	1998848	976M	EFI System
/dev/sda2	2000896	4001791	2000896	977M	Linux filesystem
/dev/sda3	4001792	54333439	50331648	24G	Linux filesystem
/dev/sda4	54333440	100661247	46327808	22.1G	Linux LVM

```
Command (m for help): w
The partition table has been altered.
Syncing disks.
```

Extend disk (PV & VG)

```
root@trixws:~# pvcreate /dev/sda4
Physical volume "/dev/sda4" successfully created.
```

```
root@trixws:~# pvs
PV          VG          Fmt  Attr  PSize   PFree
/dev/sda3   trixws-vg  lvm2  a--   <24.00g    0
/dev/sda4   trixws-vg  lvm2  a--   <22.09g <22.09g
```

```
root@trixws:~# vgs
VG          #PV #LV #SN Attr   VSize   VFree
trixws-vg   1   2   0 wz--n- <24.00g    0

root@trixws:~# vgextend trixws-vg /dev/sda4
Volume group "trixws-vg" successfully extended

root@trixws:~# vgs
VG          #PV #LV #SN Attr   VSize   VFree
trixws-vg   2   2   0 wz--n- <46.09g <22.09g
```

Extend disk (LV)

You can also use `--resizefs` in `lvextend` to avoid next step

```
root@trixws:~# lvs
  LV      VG      Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  root    trixws-vg -wi-ao---- <22.77g
  swap_1  trixws-vg -wi-ao----  <1.23g

root@trixws:~# lvextend /dev/trixws-vg/root /dev/sda4
Size of logical volume trixws-vg/root changed from <22.77 GiB (5829 extents) to <44.86 GiB (11484 extents).
Logical volume trixws-vg/root successfully resized.

root@trixws:~# lvs
  LV      VG      Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  root    trixws-vg -wi-ao---- <44.86g
  swap_1  trixws-vg -wi-ao----  <1.23g
```

Extend disk (FS)

```
root@trixws:~# df -h
Filesystem                Size  Used Avail Use% Mounted on
...
/dev/mapper/trixws--vg-root 23G   1.2G   20G    6% /
...

root@trixws:~# resize2fs /dev/trixws-vg/root
resize2fs 1.47.2 (1-Jan-2025)
Filesystem at /dev/trixws-vg/root is mounted on /; on-line resizing required
old_desc_blocks = 3, new_desc_blocks = 6
The filesystem on /dev/trixws-vg/root is now 11759616 (4k) blocks long.

root@trixws:~# df -h
Filesystem                Size  Used Avail Use% Mounted on
...
/dev/mapper/trixws--vg-root 44G   1.2G   41G    3% /
...
```

Add disk for data mount point

1. `pvcreate` to add disk into PV
2. `vgcreate` to create another storage pool (To split rootfs with data)
3. `lvcreate` for new mount point from the new VG
4. `mkfs <device>` to utilize new LV
5. mount the LV to mount point

Add disk (lsblk)

Using lsblk to check all block devices (disks, CD-ROMs, ...)

```
root@trixws:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                                 8:0    0   48G  0 disk
├─sda1                             8:1    0   976M  0 part /boot/efi
├─sda2                             8:2    0   977M  0 part /boot
├─sda3                             8:3    0    24G  0 part
│   └─trixws--vg-root              254:0    0  44.9G  0 lvm  /
│       └─trixws--vg-swap_1        254:1    0    1.2G  0 lvm  [SWAP]
└─sda4                             8:4    0   22.1G  0 part
    └─trixws--vg-root              254:0    0  44.9G  0 lvm  /
sdb                                 8:16    0     4G  0 disk
```


Add disk (LVM)

If there are more disks added to the host for /data, just use `vgextend` and `lvextend` like previous extending space example

```
root@trixws:~# pvcreate /dev/sdb
Physical volume "/dev/sdb" successfully created.

root@trixws:~# vgcreate data-vg /dev/sdb
Volume group "data-vg" successfully created

root@trixws:~# lvcreate -l 100%FREE -n data_vol data-vg
Logical volume "data_vol" created.
```

Add disk (FS)

You can revise `/etc/fstab` to ensure `/data` is mounted after reboot

```
root@trixws:~# mkfs.btrfs /dev/data-vg/data_vol
...
Number of devices: 1
Devices:
  ID      SIZE  PATH
  1      4.00GiB /dev/data-vg/data_vol

root@trixws:~# mkdir /data
root@trixws:~# mount /dev/data-vg/data_vol /data

root@trixws:~# df -h
Filesystem                Size  Used Avail Use% Mounted on
...
/dev/mapper/trixws--vg-root 44G   1.2G   41G   3% /
/dev/mapper/data--vg-data_vol 4.0G   5.8M   3.5G   1% /data
...
```

Remove disk

1. `resize2fs` to shrink target filesystem
2. `lvresize` to shrink LV
3. `vgreduce` to remove PV from VG
4. `pvremove` to remove disk from LVM
5. Eject the disk from the machine

Remove disk (lsblk)

Aims to remove sda2 from data-vol LV, and need to ensure used space is less than remained PV size

```
root@trixws:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                                  8:0      0    4G  0 disk
├─sda1                              8:1      0    2G  0 part
│   └─data--vg-data--vol1 254:2      0    4G  0 lvm  /data
└─sda2                              8:2      0    2G  0 part
    └─data--vg-data--vol1 254:2      0    4G  0 lvm  /data
sdb                                  8:16     0   48G  0 disk
├─sdb1                              8:17     0   976M  0 part /boot/efi
├─sdb2                              8:18     0   977M  0 part /boot
├─sdb3                              8:19     0    24G  0 part
│   └─trixws--vg-root      254:0      0  44.9G  0 lvm  /
│   └─trixws--vg-swap_1    254:1      0    1.2G  0 lvm  [SWAP]
└─sdb4                              8:20     0   22.1G  0 part
    └─trixws--vg-root      254:0      0  44.9G  0 lvm  /

root@trixws:~# df -h
Filesystem                Size  Used Avail Use% Mounted on
/dev/mapper/data--vg-data--vol1 4.0G  5.8M  3.5G   1% /data
```

Remove disk (FS)

Shrink filesystem size (Take BTRFS for example)

```
root@trixws:~# btrfs filesystem resize -2g /data/  
Resize device id 1 (/dev/mapper/data--vg-data--vol) from 3.99GiB to 1.99GiB  
  
root@trixws:~# df -h  
Filesystem                Size  Used Avail Use% Mounted on  
...  
/dev/mapper/data--vg-data--vol 2.0G  5.8M  1.5G   1% /data
```

Remove disk (LVM)

- Reduce LV size with the size of disk/partition
- Then, remove from top to bottom

```
root@trixws:~# lvreduce --size -2g /dev/data-vg/data-vol
File system btrfs found on data-vg/data-vol.
File system size (1.99 GiB) is equal to the requested size (1.99 GiB).
File system reduce is not needed, skipping.
Size of logical volume data-vg/data-vol changed from 3.99 GiB (1022 extents) to 1.99 GiB (510 extents).
Logical volume data-vg/data-vol successfully resized.

root@trixws:~# vgreduce data-vg /dev/sda2
Removed "/dev/sda2" from volume group "data-vg"

root@trixws:~# pvremove /dev/sda2
Labels on physical volume "/dev/sda2" successfully wiped.
```

Reference

- [Arch Linux Tutorial - How to configure LVM on Arch Linux 2017](#)
- [LVM - ArchWiki](#)
- <https://wiki.debian.org/LVM>
- [lvmtthin\(7\) - Linux manual page](#)
- [LVM, Thin Provisioning, and Monitoring Storage Use: A Case Study](#)
- [systemd-gpt-auto-generator](#)