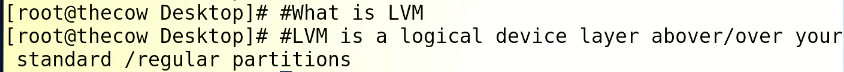
**Lecture 17**

**LVM3-Add-Extend-Remove-Logical Volume**

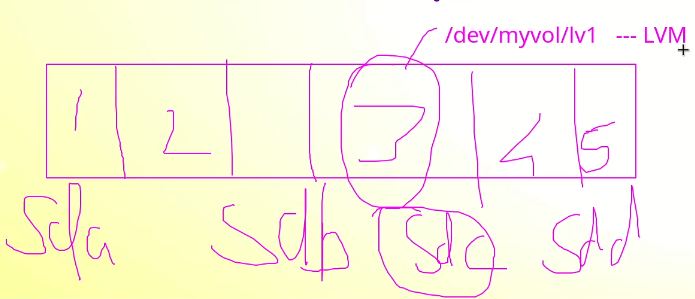
LVM working

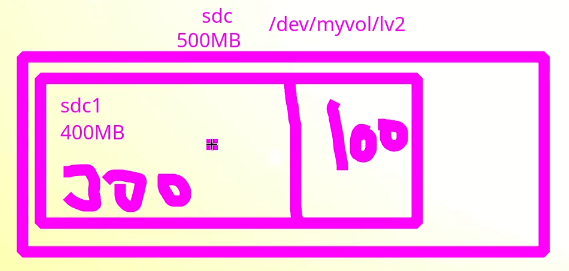
1. Creation of physical volumes: The first step in using LVM is to create physical volumes (PVs) out of disk partitions or entire disks. Each physical volume is assigned a unique identifier, called a UUID, which is used to track the disk space used by LVM.
2. Grouping physical volumes into volume groups: Next, the physical volumes are grouped into volume groups (VGs). The volume group acts as a pool of disk space, which can be divided into logical volumes.
3. Creating logical volumes: Once the physical volumes are grouped into a volume group, logical volumes (LVs) can be created. Logical volumes are the virtual disks that are exposed to the operating system. They can be created in any size up to the available space in the volume group.
4. Mapping logical volumes to physical volumes: When data is written to a logical volume, LVM maps the data to physical blocks on the underlying physical volumes. This mapping is stored in a metadata area on each physical volume.
5. Resizing logical volumes: LVM allows you to resize logical volumes dynamically, without affecting the data stored on them. This can be done by either increasing or decreasing the size of a logical volume.
6. Moving data between physical volumes: LVM also allows you to move data between physical volumes. For example, you can move data from a slow disk to a faster disk to improve performance.
7. Snapshotting logical volumes: LVM provides a snapshot feature, which allows you to create a read-only copy of a logical volume. This can be useful for backing up data or for creating a temporary copy of a volume for testing purposes.



**How to check LVM is present in my system**

$ lsblk  stands for “list block device” It is used to list all available block devices, such as hard drives, solid-state drives, USB drives, and CD-ROMs, and their associated partitions and file systems. The **lsblk** command provides information about the size, type, and other properties of the block devices and their partitions, making it useful for disk management and troubleshooting.

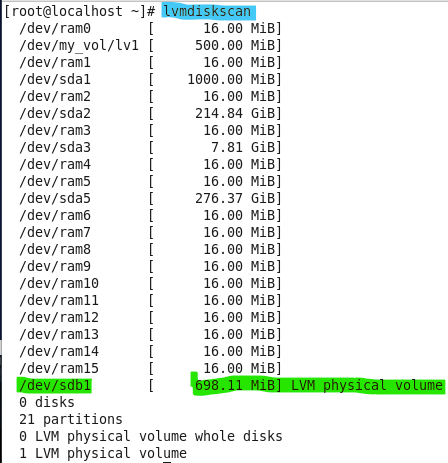


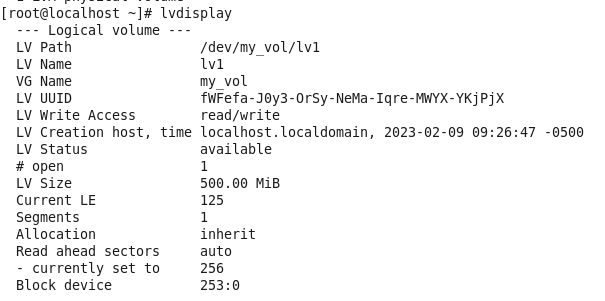


$ df -h 

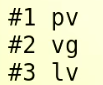
$ lvmdicskscan to check lvm is available or not  it will also show total drivers related to partitions.

**$ lvmdiskscan-l**  the -l flag will display only LVM devices or partitions.



* $ pvdisplay  to check PV (physical volume) details.
* $ vgdisplay  VG details
* $ lvdisplay  all details related to LV
* 

**to remove LVM**

* Remember While creating LVM the steps were,
* 
* Now for Removing the LVM the steps would be,
* Text

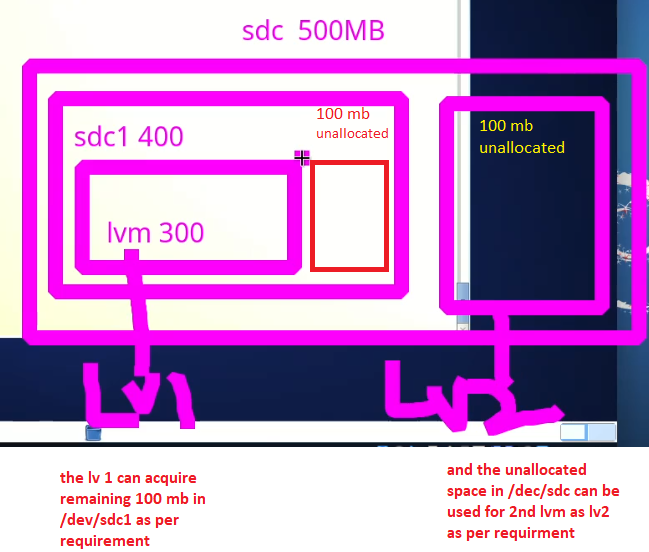
  Description automatically generated

**Steps to remove.**

* Steps
  1. Umount /common1
  2. $ lvremove /dev/my\_vol/lv1
  3. $ vgremove my\_vol
  4. $ pvremove /dev/sdb1
  5. Remove entry or comment the entry in “fstab”
  6. Remove the /dev/sdb1 partition with “fstab” if requires to completely remove the HDD partition
* Text, letter

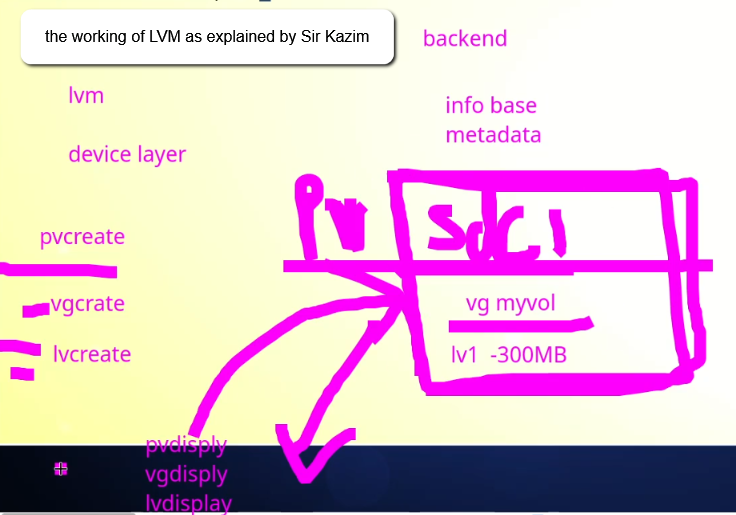
  Description automatically generated
* Text

  Description automatically generated
* To crosscheck
* **$ pvdisplay, vgdisplay, lvdisplay**  there would be no details after running this command

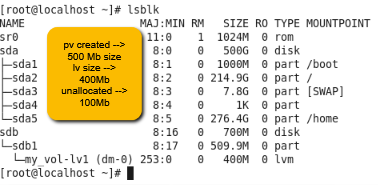


* It is possible to include unallocated space in /dev/sdc in lv1 instead of making another LVM

**LVM working explained by Sir Kazim.**



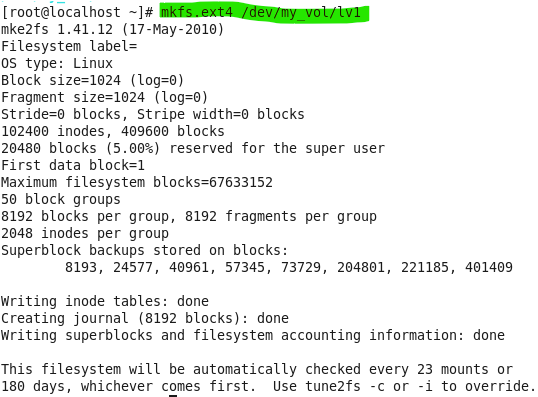
**Hands on practice**

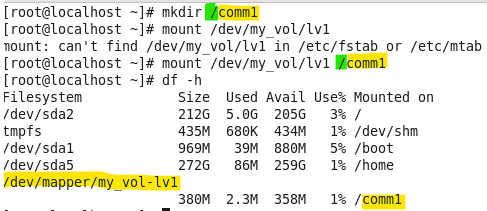


2 drivers are also created,

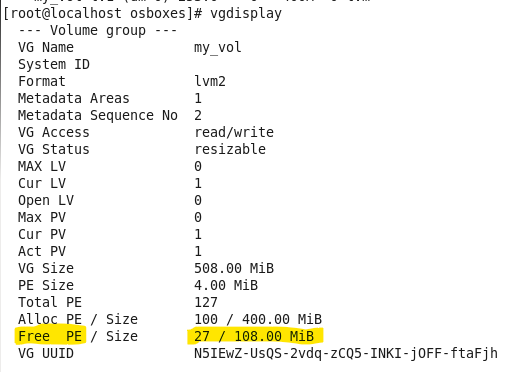
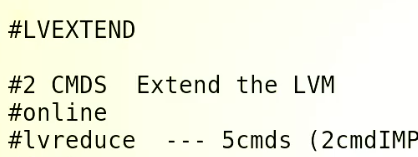
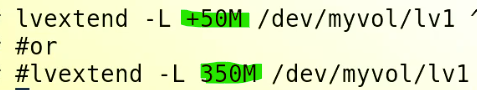
1. 
2. 

- the next step is to format,

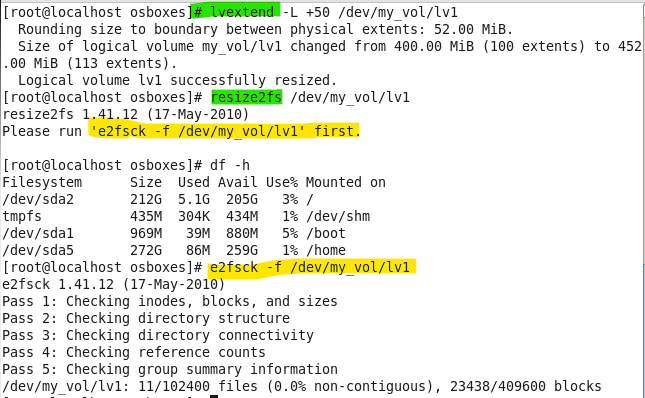


* Then, mounting to  /comm1  remember “/” means the mount point is in root or “/” it very important to keep in mind
* 

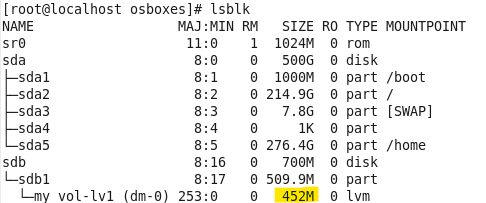
Entry into fstab

* Now,
* How to increase the size,
* To check the availability of the space,
* $ vgdisplay
* 
* It shows 100Mb free space available.
* To extend,
* 2 commands online  online means without downtime
* 
* **Remember:-** $ lvreduce  is offline means it will have downtime.
* There 2 ways to increase the size,
* 
  + - * Mention the size to be increased.
      * Mention the total size that is required.
* Then 2nd command

$ resize2 fs /dev/my\_vol/lv1 in CentOS 7 & 8 “xfs” file system is used so this command will not work there, the command would be $ xfs\_growfs /dev/my\_vol/lv1 …

* 

Size is increased.

* 

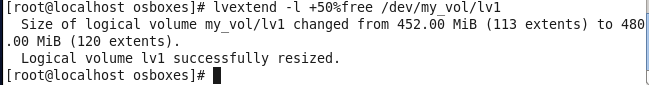
Things to remember,

* Table

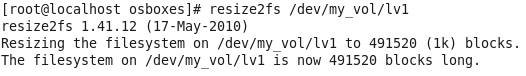
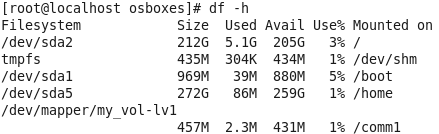
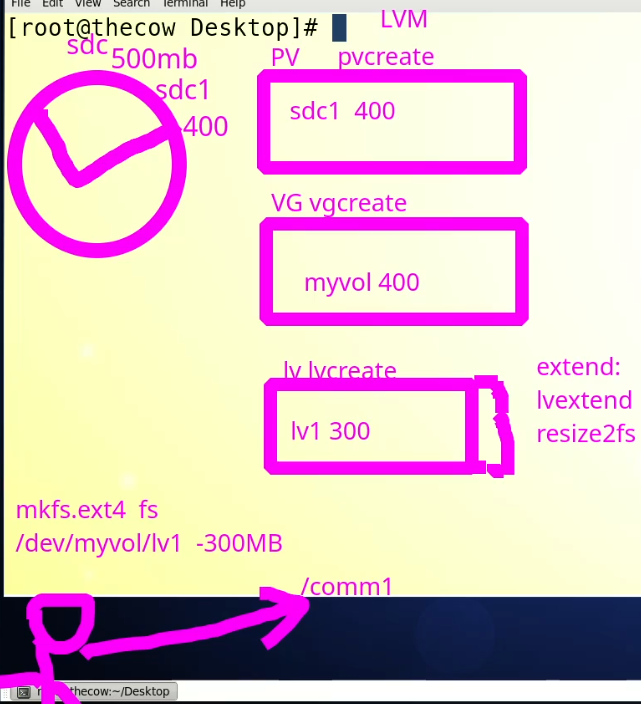
  Description automatically generated
* Remaining 56M can be added to lv1 by

**$ lvextend -L +56 /dev/my\_vol/lv1**  but this is not a recommended approach and can cause errors, so the best command to utilize remaining space is,

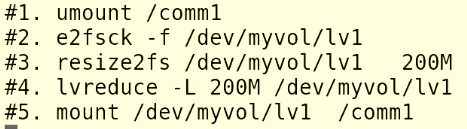
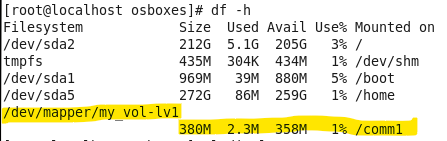
$ lvextend -l +100%free /dev/my\_vol/lv1  by this method remaining space will merge into lv1 but some space will remaining unallocated to be used for metadata etc.  **instead of 100%, 50%, 20% or 30% can also be used according to requirements.**

* 

**Remember “-L” flag is used to give size and “-l” flag is used to mention PEs.**

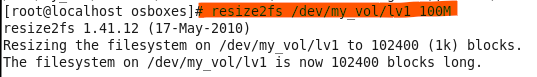
* And in the end **$ resize2fs ** for file system entry or entry into partition table
* 
* 
* Explanation by Sir Kazim Sheikh
* 

**LV reduce $ lvreduce (shrinking) – offline method requires downtime**

* Total 5 commands  only 2 are important.
* 
* Steps,
* 

1. $ umount
2. Text, table

   Description automatically generated
3. $e2fsck -f /dev/my\_vol/lv1
4. Text

   Description automatically generated
5. **e2fsck** is a utility for checking the consistency of a Linux ext2, ext3, or ext4 file system. It can be used to fix various issues, such as missing or corrupted inodes, incorrect block counts, and more.
6. $ resize2fs /dev/my\_vol/lv1 <size\_teb\_reduced>M in CentOS 7 & 8 “xfs” file system is used so this command will not work there, the command would be $ xfs\_growfs /dev/my\_vol/lv1…
7. 
8. $ lvreduce -L <size\_to\_be\_reduced>M /dev/my\_vol/lv1
9. Text

   Description automatically generated
10. $ mount /comm1

Remember

* 
* sir told us lv reduce method according to the centos6.
* but in centos 7 or 8 the process is different .
* the lv reduce for centos 7 or 8 are as following:
* 1  umount  /comm1
* 2   mke2fs -n  /dev/myvol/lv1
* 3    xfs\_growfs   -L   /dev/myvol/lv1   200M
* 4    lvreduce  -L  200M   /dev/myvol/lv1
* 5    [mkfs.xfs](https://mkfs.xfs/)    -f  /dev/myvol/lv1
* 6     mount -a
* 7     df -h