# Assignment

Logical Volume Manager (LVM) is a disk management technology in Linux that allows you to manage disk drives and storage volumes with greater flexibility compared to traditional partitioning methods.

**Key Concepts:**

**1) Physical Volume (PV)**: Represents a physical disk or a partition that LVM can use.

**2) Volume Group (VG)**: A pool of storage created by grouping multiple physical volumes.

**3) Logical Volume (LV)**: The equivalent of a partition created from the space within a volume group, which can be resized as needed.

To create Volumes for the drive, lets say three:

**pvcreate /dev/sdb /dev/sdc /dev/sdd**

Before this command, we need to add three disks. You are initializing three physical disks (/dev/sdb, /dev/sdc, and /dev/sdd) as Physical Volumes (PVs) in LVM.

**pvdisplay**

The pvdisplay command in LVM is used to display detailed information about physical volumes. After creating physical volumes with pvcreate, you can use pvdisplay to get detailed information like the size, volume group association, status, and more.

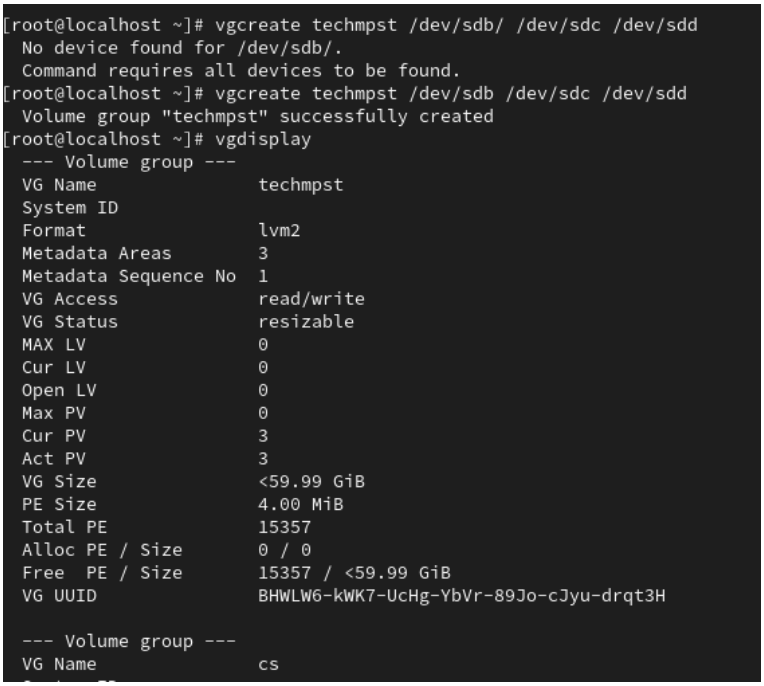
```
[root@localhost ~]# pvcreate /dev/sdb /dev/sdc /dev/sdd
  Physical volume "/dev/sdb" successfully created.
  Physical volume "/dev/sdc" successfully created.
  Physical volume "/dev/sdd" successfully created.
[root@localhost ~]# pvdisplay
  --- Physical volume ---
  PV Name               /dev/sda2
  VG Name               cs
  PV Size               <19.00 GiB / not usable 3.00 MiB
  Allocatable           yes (but full)
  PE Size               4.00 MiB
  Total PE              4863
  Free PE               0
  Allocated PE          4863
  PV UUID               Hi50Ea-PTBv-8PbO-3vHK-P1nY-g5rO-Q1XMox

  "/dev/sdb" is a new physical volume of "20.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb
  VG Name
```

**vgcreate techmpst /dev/sdb /dev/sdc /dev/sdd**

It creates a Volume Group (VG) named techmpst by combining the three physical volumes /dev/sdb, /dev/sdc, and /dev/sdd. This volume group pools the storage from these disks, allowing you to create flexible logical volumes (LVs) on top of it.

**vgdisplay**

The vgdisplay command in LVM provides detailed information about volume groups, including their name, size, free space, and physical extent details. It displays whether the volume group is allocatable, the number of physical extents (PEs), and the unique identifier (UUID) of the volume group. This helps in monitoring and managing storage pools created from physical volumes.




**lvcreate -L 8000 techmpst -n sales /dev/sdb /dev/sdc /dev/sdd**

It is used to create a Logical Volume (LV) named sales with a size of 8000MB (8GB) from the Volume Group (VG) named techmpst. The -L 8000 option specifies the desired size for the logical volume, while the -n sales option sets its name. Importantly, there is no need to specify the individual physical volumes (such as /dev/sdb, /dev/sdc, or /dev/sdd) again, as they are already included in the techmpst volume group. After executing this command, the

logical volume can be formatted with a filesystem and mounted for use in your system.

In the same manner,

**lvcreate -L 8000 techmpst -n mkt /dev/sdb /dev/sdc /dev/sdd**

**lvcreate -L 4000 techmpst -n IT /dev/sdb /dev/sdc /dev/sdd**

**lvcreate -L 4000 techmpst -n researchdept /dev/sdb /dev/sdc /dev/sdd**

```
[root@localhost ~]# lvcreate -L 8000 techmpst -n sales /dev/sdb /dev/sdc /dev/sdd
  Logical volume "sales" created.
[root@localhost ~]# lvcreate -L 8000 techmpst -n mkt /dev/sdb /dev/sdc /dev/sdd
  Logical volume "mkt" created.
[root@localhost ~]# lvcreate -L 4000 techmpst -n IT /dev/sdb /dev/sdc /dev/sdd
  Logical volume "IT" created.
```

```
[root@localhost ~]# lvcreate -L 4000 techmpst -n researchdept /dev/sdb /dev/sdc /dev/sdd
  Logical volume "researchdept" created.
```

## lvdisplay

It is used to show detailed information about logical volumes (LVs) in LVM (Logical Volume Manager). When executed, it provides a summary of each logical volume, including its name, the volume group it belongs to, the device path, size, number of allocated and current logical extents, type, access permissions, status, creation time, and unique identifier (UUID).

```
[root@localhost ~]# lvdisplay
  --- Logical volume ---
  LV Path                /dev/techmpst/sales
  LV Name                sales
  VG Name                techmpst
  LV UUID                e0Cr6l-HXNu-5Z1L-zC25-A50t-CXBc-YVRla4
  LV Write Access        read/write
  LV Creation host, time localhost.localdomain, 2024-09-24 17:47:25 +0530
  LV Status              available
  # open                 0
  LV Size                7.81 GiB
  Current LE             2000
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:3

  --- Logical volume ---
  LV Path                /dev/techmpst/mkt
  LV Name                mkt
  VG Name                techmpst
  LV UUID                76hGzH-7vqI-T3bJ-u8Ee-5ePV-bse3-aLXAex
  LV Write Access        read/write
  LV Creation host, time localhost.localdomain, 2024-09-24 17:47:38 +0530
  LV Status              available
  # open                 0
  LV Size                7.81 GiB
  Current LE             2000
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
```

**mkdir /sales /mkt /IT /researchdept**

It creates four directories in the root filesystem: /sales, /mkt, /IT, and /researchdept. Each directory serves as a separate folder for organizing files and data related to their respective departments or purposes.

Breakdown of the Command:

mkdir: Stands for "make directory," which is the command used to create new directories.

/sales: Creates a directory named sales.

/mkt: Creates a directory named mkt (possibly for marketing).

/IT: Creates a directory named IT (for information technology).

/researchdept: Creates a directory named researchdept (for the research department).

**mount -a /techmpst /sales /sales**

Breakdown of the Commands:

mount -a:

This command mounts all filesystems listed in the /etc/fstab file that are not currently mounted. It does not require any additional parameters.

mount /dev/techmpst /sales:

This command mounts the logical volume located at /dev/techmpst to the directory /sales. Make sure that the /sales directory exists before you attempt to mount.

The same manner,

**mount -a /techmpst /mkt /mkt**

**mount -a /techmpst /IT /IT**

**mount -a /techmpst /researchdept / researchdept**

**mount**

This command is used to attach filesystems to a specified mount point in the directory tree. When you run the mount command without any arguments, it displays a list of all currently mounted filesystems along with their mount points and other details.

## RAID:

RAID (Redundant array of independent disks) is a storage technology that combines multiple disk drives into one logical unit. It can help improve performance and data redundancy. CentOS, a Linux distribution, supports RAID configurations during installation and offers a few ways to use RAID.

Categories of RAID:

**RAID 0:**

RAID 0, or “striping,” is a RAID configuration involving the division of data across the disks where there is no redundancy of the disks. It increases performance since it provides both read and write access, while it does not support data backup. Data is only safe when it’s stored in two drives at once; if one fails, then the data is gone. RAID 0 is used for applications that demand very high speed but do not concern themselves with data protection.

**RAID 1:**

RAID 1 is defined as the ‘mirroring’ in which the same set of data is written onto two or more drives. It will help to have duplication, and if one of the drives fails, the data is still available in the other drives. Although it provides extremely high data reliability and data protection features, this concept does not enhance performance and demands double space as compared to the data.

**RAID 2:**

It consist of Bit-level Striping. RAID 2 records Error Correction Code (ECC) using hamming code parity. In this level each data bit in a word is recorded on a separate disk and ECC codes of the data words are stored on a different set of disk.

**Advantages–**

1. In case of Error Correction it uses hamming code.
2. It Uses one designated drive to store parity.

**Disadvantage –**

1. It has complex structure and high cost due to extra drive.
2. It require extra drive for error detection.

**RAID 3:**

RAID 3 consists of Byte-level Striping. It stripes the data onto multiple disk. The parity bit generated for each disk section and stored on a different dedicated disk. This level overcome the single disk failure.

**Advantages –**

1. Data can be transferred in bulk.
2. Data can be accessed in parallel.

**Disadvantages –**

1. It require an additional drive for parity.
2. In case of small size files it performs slowly

**RAID 4:**

Block-level striping and a separate parity disk are used in RAID 4. This configuration stores the parity information for each data block on a different disk, and divides the data into blocks that are striped across multiple disks.

**RAID-5 (Block-Level Stripping with Distributed Parity):**

This is a slight modification of the RAID-4 system where the only difference is that the parity rotates among the drives.

**RAID-6 (Block-Level Stripping with two Parity Bits):**

Raid-6 helps when there is more than one disk failure. A pair of independent parities are generated and stored on multiple disks at this level. Ideally, you need four disk drives for this level.

There are also hybrid RAIDs, which make use of more than one RAID level nested one after the other, to fulfill specific requirements

## Implementation:

Initially, remove the logical volume and volume group named techmpst:

**lvremove techmpst**

**vgremove techmpst**

Then, remove the physical volume:

**pvremove /dev/sdb /dev/sdc /dev/sdd**

**mdadm -C /dev/md0 -n3 /dev/sdb /dev/sdc /dev/sdd -l5**

It is used to create a RAID 5 array. Let's break it down:

mdadm: This is the command-line tool used for managing RAID arrays.

-C: This option specifies that we want to create a new array.

/dev/md0: This is the name of the RAID array we're creating. It will be accessible as a block device.

-n3: This option specifies the number of disks (or devices) that will be part of the RAID array. In this case, it's 3.

/dev/sdb /dev/sdc /dev/sdd: These are the paths to the physical disks that will be used to create the RAID array.

-l5: This option specifies the RAID level. In this case, it's RAID 5.

```
[root@localhost ~]# lvremove techmpst
Do you really want to remove active logical volume techmpst/sales? [y/n]: y
  Logical volume "sales" successfully removed.
Do you really want to remove active logical volume techmpst/mkt? [y/n]: y
  Logical volume "mkt" successfully removed.
Do you really want to remove active logical volume techmpst/IT? [y/n]: y
  Logical volume "IT" successfully removed.
Do you really want to remove active logical volume techmpst/researchdept? [y/n]: y
  Logical volume "researchdept" successfully removed.
[root@localhost ~]# vgremove techmpst
  Volume group "techmpst" successfully removed
[root@localhost ~]# pvremove /dev/sdb /dev/sdc /dev/sdd
  Labels on physical volume "/dev/sdb" successfully wiped.
  Labels on physical volume "/dev/sdc" successfully wiped.
  Labels on physical volume "/dev/sdd" successfully wiped.
[root@localhost ~]# mdadm -C /dev/md0 -n3 /dev/sdb /dev/sdc /dev/sdd -l5
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

## mdadm -D /dev/md0

It is used to display detailed information about a RAID array named /dev/md0

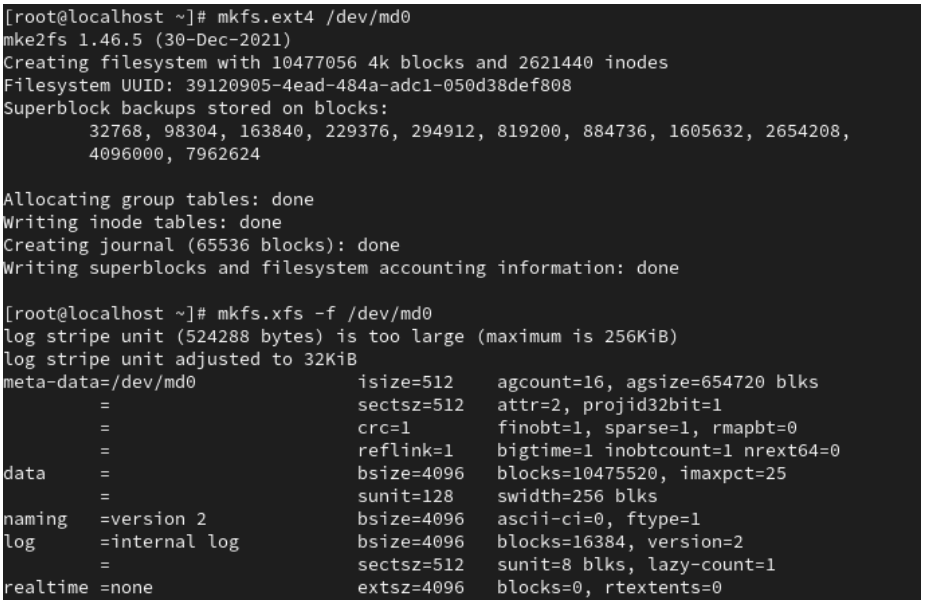
```
[root@localhost ~]# mdadm -D /dev/md0
/dev/md0:
           Version : 1.2
     Creation Time : Tue Sep 24 21:41:35 2024
        Raid Level : raid5
        Array Size : 41908224 (39.97 GiB 42.91 GB)
     Used Dev Size : 20954112 (19.98 GiB 21.46 GB)
      Raid Devices : 3
     Total Devices : 3
       Persistence : Superblock is persistent

       Update Time : Tue Sep 24 21:42:00 2024
             State : clean, degraded, recovering
    Active Devices : 2
   Working Devices : 3
    Failed Devices : 0
     Spare Devices : 1

            Layout : left-symmetric
        Chunk Size : 512K

Consistency Policy : resync

    Rebuild Status : 21% complete

              Name : localhost.localdomain:0  (local to host localhost.localdomain)
              UUID : c8b2c92d:face6182:0d6b4666:df0d9ead
            Events : 4

    Number   Major   Minor   RaidDevice State
       0       8       16        0      active sync   /dev/sdb
       1       8       32        1      active sync   /dev/sdc
       3       8       48        2      spare rebuilding   /dev/sdd
```

## mkdir /raid

It creates a new directory named "raid" in the root directory of your filesystem.

**mkfs.ext4 /dev/md0**

It is used to format a RAID array named /dev/md0 with the EXT4 filesystem.



```
[root@localhost ~]# mkfs.ext4 /dev/md0
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 10477056 4k blocks and 2621440 inodes
Filesystem UUID: 39120905-4ead-484a-adc1-050d38def808
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624

Allocating group tables: done
Writing inode tables: done
Creating journal (65536 blocks): done
Writing superblocks and filesystem accounting information: done

[root@localhost ~]# mkfs.xfs -f /dev/md0
log stripe unit (524288 bytes) is too large (maximum is 256KiB)
log stripe unit adjusted to 32KiB
meta-data=/dev/md0               isize=512    agcount=16, agsize=654720 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1 nrext64=0
data     =                       bsize=4096   blocks=10475520, imaxpct=25
         =                       sunit=128    swidth=256 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=16384, version=2
         =                       sectsz=512   sunit=8 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

**mount /dev/md0 /raid**

It mounts the RAID array /dev/md0 to the directory /raid, making its contents accessible.

**cd /raid**

It is used to change the current directory to /raid.

**touch t1 t2 t3**

It is used to create three empty files named t1, t2, and t3 in the current directory i.e raid.

To remove HDD if disk is faulty:

**mdadm -f /dev/md0 /dev/sdd**

It marks the disk /dev/sdd as failed in the RAID array /dev/md0, triggering a rebuild process.

**mdadm -r /dev/md0 /dev/sdd**

It starts a rebuild process for the failed disk /dev/sdd in the RAID array /dev/md0.

```
[root@localhost raid]# mdadm -f /dev/md0 /dev/sdd
mdadm: set /dev/sdd faulty in /dev/md0
[root@localhost raid]# mdadm -r /dev/md0 /dev/sdd
mdadm: hot removed /dev/sdd from /dev/md0
```

**mdadm -a /dev/md0 /dev/sdd**

It is used to add a disk to a RAID array.

```
[root@localhost raid]# mdadm -a /dev/md0 /dev/sdd
mdadm: added /dev/sdd
```

**mdadm -S /dev/md0 /dev/sdd**

It is used to stop the rebuild process for a specific disk in a RAID array.