2. Logical Volume Management and RAID Levels

1.   What is LVM and why we go for LVM?
         Lvm means Logical Volume Management. The combination of 2 or more physical disk in order to make a
big   logical disk is called Logical Volume.
         If normal Linux partition is full and an application requires some more disk space, then normal partition
cannot   be extended for that application requirement. For this first we have to take a backup of that normal
partition,   delete that partition and again create that partition with more disk space, format and mount that
partition and   finally restore the application from the backup. This process requires down time.
         So, to overcome this problem LVM concept is coming into the picture. Using this LVM we can extend or
reduce   the file systems as per requirement without loss of any data.

2.   What are the components of the LVM?
Physical Volume (PV)
Physical Extent (PE)
Volume Group (VG)
Logical Volume (LV)
Logical Extent (LE)

Physical Volume (PV) :
It is the standard partition that we add to the LVM. Normally a physical volume is a standard primary or logical
partition with the partition code as **8e**.

Physical Extent (PE) :
It is chunk of disk space. Every physical volume is divided into a number of equal sized PEs.

Volume Group (VG) :
It is composed of a group of physical volumes and logical volumes. It is the organizational group of LVM.

Logical Volume (LV) :
It is composed of a group of LEs. We can format (make a file system) and mount any file system on the logical
volume. The size of these logical volumes can easily be increased or decreased as per the requirement.

Logical Extent (LE) :
         It is also a chunk of disk space. Every logical extent is mapped to a specific physical extent.

3.   How to create the LVM, make a file system and mount that permanently?
         (i) Take two physical disks for example **/dev/sdb**   and   **/dev/sdc**. if there is no second disk then make
the   required partitions using **# fdisk** command and change the partition code as **8e**.
         (ii) Convert the Physical disk into physical volumes by,
                # pvcreate   /dev/sdb   /dev/sdc
(iii) Then create the volume group by combining physical volumes by,
         # vgcreate <volume group name><physical volume names>   or
         # vgcreate   -s <PE size in MBs><volume group name><physical volume names>
(iv) Then create the logical volume on the above created volume group by,
         # lvcreate -L +<size in MBs>   -n <logical volume name><Volume group name> or
         # lvcreate -l <no. of PEs>   -n <logical volume name><volume group name>
(v) Make a file system on the above created logical volume by,
         # mkfs.ext2/ext3/ext4/xfs   /dev/<volume group name>/<logical volume name>

(vi) Create a mount point to mount the above created LVM file system by,
         # mkdir   /mnt/<directory name>
(vii) Mount the LVM on the above created mount point temporarily by,
         # mount   /dev/<volume group name>/<logical volume name><mount point>or
         Mount the LVM on mount point permanently by,
         # vim /etc/fstab
         /dev/<VG name>/<LV name>   /mnt/<directory>   <file system type>   defaults 0
         0
         Esc+:+wq!

 # mount   -a

 # df   -hT   (to see the mounted partitions with file system types)

4.   How to see the details of the Physical Volumes?

　　　# pvs   (displays all physical volumes with less details)

　　　# pvdisplay   (displays all physical volumes with more details)

　　　# pvdisplay   <physical volume name>   (displays the details of the specified physical volume)

　　　# pvscan   (to scan all the physical volumes)

　　　#pvscan   <PV name>   (to scan the specified physical volume)

5.   How to see the details of the Volume Groups?

　　　# vgs   (displays all volume groups with less details)

　　　# vgdisplay   (displays all volume groups with more details)

　　　# vgdisplay   <VG name>   (displays the specified volume group with more details)

　　　# vgscan   (to scan all the volume groups)

　　　# vgscan   <VG name>   (to scan the specified volume group)

6.   How to see the details of the Logical Volumes?

　　　# lvs   (displays all logical volumes with less details)

　　　# lvdisplay   (displays all logical volumes with more details)

　　　# lvdisplay   <LV name>   (displays the specified logical volume details)

　　　# lvscan   (to scan all the logical volumes)

　　　# lvscan   <LV name>   (to scan the specified logical volume)

7.   How to extend the Volume Group?

Extending the volume group is actually adding a new physical volume to the volume group.

To extend the volume group we need to create a new partition using **# fdisk** command and make sure that it's partition id should be **8e**, save the changes and update the partition table by **# partprobe**

Create a physical volume on the newly created partition using **# pvcreate** command.

Add the partition to the volume group using **# vgextend** command

　　　Example : # fdisk   /dev/sdb

　　　　　　　Command (m for help) : n

　　　　　　　First cylinder : press Enter for default one

　　　　　　　Last cylinder : +500M   (create 500MB partition)

　　　　　　　Command (m for help) : t   (to change the partition id)

　　　Select the partition : type the partition number

　　　Specify the Hexa code : 8e

　　　Command (m for help) : w   (to save the changes)

　　　# partprobe   /dev/sdb1

　　　# pvcreate   /dev/sdb1

　　　# vgextend   <VG name>   /dev/sdb1

　　　# vgdisplay   <VG name>   (to check the size of the volume group)

8.   How to extend the logical volume and update it's file system?

Sometimes the file system size may be full, so we need to increase the size of the logical volume to continue adding the data in it.

The size of the logical volume can be increased online, no downtime required.

Check current size of the logical volume by **# lvdisplay <LV name>**and the size of the file system by   **# df -hT** command.

Increase the size of the logical volume by **# lvextend or # lvresize** commands.

Then finally update the file system by **# resize2fs or # xfs_growfs** commands.

　　　Example :   # df   -hT

　　　　　　　# lvextend   -L   +<size in MB></dev/vgname/lvname>   or

　　　　　　　# lvresize   -L   +<size in MB></dev/vgname/lvname>

　　　　　　　# resize2fs   </dev/vgname/lvname>

　　　　　　　# lvdisplay   </dev/vgname/lvname>   (to check the size of the logical volume)

　　　　　　　# df   -hT   (to check the size of the file

system)

9.   How to reduce the logical volume and update the file system?

Reducing the size of the logical volume is a complicated task and we have remember some points before reducing the logical volume, otherwise the file system may be damaged .

Logical volume size cannot be reduced online and it requires downtime because we have to unmount the file system by **# umount   <file system mount point>**command.

Check the consistency of the file system by **# e2fsck   <device or partition name>** command.

Reduce the logical volume by **# lvreduce   -L   - <Size of in MB></dev/vgname/lvname>** command.

Then update the file system by **# resize2fs   </dev/vgname/lvname>**

Finally mount the file system by **# mount   -a**

        Example :   # umount   <file system mount point>

                    # e2fsck <device or partition name>

                    # lvreduce   -L   -<size in MB></dev/vgname/lvname>

                    # resize2fs </dev/vgname/lvname>

                    # lvdisplay   </dev/vgname/lvname>   (to check the size of the logical volume>

                    # mount   -a   (to mount the file system)

                    # df   -hT   (to check the size of the file system)

10.   How to move or migrate the logical volume data from one physical volume to another physical volume?

There might be a situation where the physical volume might be failing and it is required to replaced. In such case, we need to migrate or move the logical volume data from the failed physical volume new physical volume and isolate (remove) the failed physical volume.

First access the mount point of the failing physical volume and check the data in it.

Verify the size of the physical volume by **#pvs**   or   **#pvdisplay   </dev/vgname/lvname>**command.

Unmount the file system of that physical volume by **# umount   <file system mount point>**

Add a new physical volume and the size should be same size or higher than that failing physical volume.

 Migrate the physical volume contents to the new physical volume using **# pvmove <old PV><new PV>**

Mount back the logical volume, access the mount point and verify the data in it.

Remove the failed the physical volume by **#vgreduce   <vgname><pvname>** command.

        Example :   # cd   <file system mount point>

                    # ls

                    # pvs <pvname>   or # pvdisplay <pvname>

                    # umount   <file system mount point>

                    # pvcreate   <device or partition name>

                    #vgextend   <vgname><pvname>

                    # pvmove <old pvname><new pvname>

                    # mount   -a

                    # vgreduce   <vgname><failed pvname>

                    # cd   <file system mount point>

                    # ls

11.   How to delete or remove the logical volume?

To delete or remove the logical volume, first unmount the file system by **# umount   <mount point>**

Remove the entry in /etc/fstab file.

Remove the logical volume by **# lvremove   </dev/vgname/lvname>**command.

Verify whether the logical volume is removed or not by **# lvs   or   # lvdisplay**   command.

        Example :   # umount <file system mount point>

                    # vim /etc/fstab   (delete the entry of the logical volume)

                    Esc+:+wq!   (save and exit the file)

                    # lvremove   </dev/vgname/lvname>

                    # lvs   or   # lvdisplay   (to verify whether logical volume is removed or not)

12.   How to delete or remove the volume group?

To delete or remove the volume group, first make sure that any logical volume should not be mounted because while removing a volume group it will delete or remove the logical volumes in that volume group.

Then delete or remove the volume group by **# vgremove   <vgname>**command.

Verify whether the volume group is remove or not by **# vgs   or   # vgdisplay**   command.

Example :   # umount   <file system mount point>   (to unmount the file system if there is any LV)

                 # vim   /etc/fstab   (delete the entry of the logical volume)
                 Esc+:+wq!   (save and exit the file)
                 # vgremove   <vgname>
                 # vgs   or   # vgdisplay   (to verify whether volume group is removed or not)

13.   How to delete or remove the physical volume?

Deleting or removing a physical volume is very simple and the only thing we should check that the physical volume we are going to delete should not belong to any volume group ie., we can only delete or remove the physical volume which is free.

Then delete or remove the physical volume by **# pvremove   <pvname>**command.

Verify whether the physical volume is removed or not by **# pvs   or   #pvdisplay** command.

          Example : # pvremove <pvname>
                            # pvs   or   #pvdisplay   (to verify whether the physical volume is removed or not)

14.   How to restore the volume group which is removed mistakenly?

First unmount file system by **# umount   <file system mount point>** command.

Check the volume group backup list by **# vgcfgrestore   --list   <volume group name>**command.

Then remove the logical volume by **# lvremove   </dev/vgname/lvname>** command.

Copy the backup file which is taken backup before removed the volume group from the above backup list and paste it in this command **# vgcfgrestore   -f   <paste the above copied file name><vgname>**

The logical volume is created automatically after restoring the volume group but the volume group and logical volumes both will be in inactive state. So, check the state of the volume group by **#vgscan**and the logical volume state by **# lvscan** commands.

Then activate that volume group by **# vgchange   -ay   <volume group name>**commandand activate the logical volume by # **lvchange   -ay   <logical volume name>**command.

Mount the logical volume file system by **# mount   -a** command.

          Example :   # umount   <file system mount point>
                           # vgcfgrestore   --list <volume group name>   (copy the backup file from the list)
                           # lvremove   </dev/vgname/lvname>
                           # vgcfgrestore   -f   <paste the above copied file><volume group name>
                           # vgscan   (to check the status of the volume group)

                           # lvscan   (to check the status of the logical volume)

                           # vgchange   -ay   <volume group name>   (activate the volume group if it is in inactive state)

                           # lvchange   -ay   <logical volume name>   (activate the logical volume if it is in inactive state)

┌───┐
│   **Note:** The option **a**   means active VG or LV and option   **y**   means yes.   │
└───┘

                           # mount   -a

15.   How to change the volume group name and other parameters?

          # vgrename   <existing volume group name><new volume group name> (to rename the volume group)
          By default, unlimited logical volumes can be created per volume group. But we can control this limit by
          # vgchange   -l <no.><volume group>   (to limit max. no. of logical volumes to the specified number)
          Example : # vgchange -l 2 <vgname>   (to limit max. 2 logical volumes cab be created in this volume group)
          # vgchange   -p <no.><volume group>   (to limit max. no. of physical volumes to the specified number)

Example : # vgchange -p 2 <vgname>   (to limit max. 2 physical volumes can be added to this volume group)

vgchange   -s <block size in no.><volume group>   (to change the block size of the volume group)

Example : # vgchange   -s 4 <vgname>   (to change the volume group block size to 4MB)

16.   How to change the logical volume name and other parameters?
lvrename <existing lvname><new lvname>   (to rename the logical volume)
lvchange   -pr <logical volume>   (to put the logical volume into read only mode)
lvs   (to see the logical volume permissions)
lvchange   -prw <logical volume> (to put the logical volume into read and write mode)

17.   How to disable the volume group and logical volume?
vgchange   -an   <volume group>   (to disable the volume group)
lvchange   -an   <logical volume>   (to disable the logical volume)

18.   How to take a backup of the volume group?
vgcfgbackup   (to take a backup of all volume groups)
vgcfgbackup <volume group>   (to take a backup of the specified volume group)

19.   What is the configuration file of the logical volume?
cat /etc/lvm/lvm.conf   (to see the contents of the LVM configuration file)

20.   What are the locations of the logical volume and volume groups?
cd /etc/lvm/backup   (the logical volumes backup location)
cd /etc/lvm/archive   (the volume groups backup location)

21.   How to know the current version of the LVM package?
rpm -qa lvm*   (to know the current version of the LVM package)

22.   What are the attributes of the volume group?
vgs   (to see the attributes of the volume group)
[The attributes are w ----> writable   z ----> extendable   n ----> normal]
vgs   -v   (to check the UUID of the volume group)

23.   How to extend the logical volume to max. disk space and half of the disk space?
lvextend   -l +100% FREE   <logical volume>   (to extend the logical volume by adding the
                            volume group's   total available space)
lvextend   -l 50% <vgname><lvname>   (to extend the logical volume by adding the 50%
                                                  free space   of the volume group)

24.   How to check on which physical volume the data is writing in the logical volume?
lvdisplay   -m   (to check on which physical volume the data is currently writing from all
          logical volumes)
lvdisplay   -m <lvname>   (to check on which physical volume the data is writing from the
                                  Specified   logical volume)

25.   How many types of file systems available?
ext2 ---->Second extended file system (default in RHEL - 3 & 4)
ext3 ---->Third extended file system (default in RHEL - 5)
ext4 ----> Fourth extended file system (default in RHEL - 6)
xfs ----> Extended file system (default in RHEL - 7)
ufs ----> Unix file system (default in Solaris)

jfs ----> Journal file system (default in IBM-AIX)
hfs ----> High performance file system (default in HP-UX)
vxfs ----> Veritas file system
procfs ----> Process file system (temporary)
tempfs ----> Temporary file system (temporary)
cdfs ----> Compact disk file system
hdfs ----> DVD file system
iso9660 ----> To read the CD/DVD.iso image format files in Linux

26. **How to scan and detect the luns over the network?**
ls /sys/class/fc_host   (to check the available fibre channels)
echo "---" > /sys/class/scsi_host/<lun no.>/scan   (to scan and detect the luns over the network)

27. **How to mount a pen drive in Linux?**
lsusb   or   # fdisk -l   (to know the pen drive name)
mkdir /mnt/pendrive   (to create a mount point for pen drive)
mount   <pen drive name><mount point>   (to mount the pen drive on the above created mount point)
cd   /mnt/pendrive   (to access the pen drive)

28. **How to mount a CD/DVD ROM drives in Linux?**
The CD/DVD ROM device name in Linux is /dev/cdrom
mkdir /mnt/mycdrom   (to create the mount point for CD/DVD)
mount /dev/cdrom /mnt/mycdrom   (to mount the CD/DVD on the above created mount point)
cd /mnt/mycdrom   (to access the CD/DVD ROM drives)

29. **How to mount the " .iso " image files in Linux?**
mount -t iso9660 /root/rhel6.iso /iso -o ro, loop   (to mount the .iso image files)
cdrecord /root/Desktop/rhel6.iso   (to write the CD/DVD ROM. Before executing this command put the empty CD/DVD into CD/DVD drive)
eject   (to eject the CD/DVD drive tray)
eject   -t   (to insert and close the CD/DVD drive tray)

30. **What is RAID? What is the use of the RAID and how many types of RAIDs available?**
RAID stands for Redundant Array of Independent Disks.
It provides fault tolerance, load balancing using stripping, mirroring and parity concepts.
There are mainly two types of RAIDs available.
(i) Hardware RAID (Depends on vendors and also more expensive)
(ii) Software RAID (Does not depends on vendors and less expensive when compared to Hardware RAID and also it is maintained by system administrator only.

31. **How many types of software RAIDs available and their requirements?**
(i) RAID - 0   ---- Stripping   ----   Minimum 2 disks required
(ii) RAID - 1   ---- Mirroring   ----   Minimum 2 disks required
(iii) RAID - (1+0) ---   Mirroring + Stripping ---- Minimum 4 disks required
(iv) RAID - (0+1) ---   Stripping + Mirroring ---- Minimum 4 disks required
(v) RAID - 5   ---- Stripping with parity ----   Minimum 3 disks required

32. **How to configure RAID - 0 in Linux?**
To configure RAID - 0, minimum 2 disks are required and the partition id is "fd".
Reading and writing is very fast. So, it produces high performance.
if one disk is failed we cannot recover the data.

So, there is no redundancy and fault tolerance in RAID - 0.

Example : For example if the data is 1, 2, 3, 4, 5 and 6 then ....

```
┌─────────┐        ┌─────────┐
│ 1       │        │ 2       │
│ 3       │        │ 4       │
│ 5       │        │ 6       │
│         │        │         │
└─────────┘        └─────────┘
  Disk - 1            Disk - 2
```

If the Disk - 1 is /dev/sdb and the Disk - 2 is /dev/sdc then,
mdadm -Cv /dev/md0 -n 2 /dev/sdb /dev/sdc -l 0 (to create the RAID - 0 using disk - 1 and disk - 2)

Command	Description
# cat /proc/mdstat	(to check the RAID - 0 is created or not)
# mkfs.ext4 /dev/md0	(to create the ext4 file system on the RAID - 0)
# mkdir /mnt/raid0	(to create the RAID - 0 mount point)
# mount /dev/md0 /mnt/raid0	(to mount RAID - 0 on the mount point)
# mdadm -D /dev/md0	(to see the details of the RAID - 0 partition)
# mdadm /dev/md0 -f /dev/sdb	(to failed the disk manually)
# mdadm /dev/md0 -r /dev/sdb	(to remove the above failed disk)
# mdadm /dev/md0 -a /dev/sdd	(to add the new disk in place of failed disk)
# umount /mnt/raid0	(to unmount the raid file system)
# mdadm --stop /dev/md0	(to stop the RAID - 0 volume)
# mdadm /dev/md0 --add /dev/sde	(to add third disk to the RAID - 0 volume)
# mdadm --grow /dev/md0 --raid_device=3	(to grow the RAID - 0 file system)

33. How to configure RAID - 1 in Linux?

To configure RAID - 1, minimum 2 disks are required and the partition id is "fd".
In this the same data will be written on 2 disks ie., exact copy on both the disks.
if one disk is failed we can recover the data from another disk.
So, there is a high availability, redundancy and fault tolerance in RAID - 1.
In this writing speed is slow compared to RAID - 0.

Example : For example if the data is 1, 2, 3, 4, 5 and 6 then ....

```
┌──────┬──────┐
│ 1    │ 1    │
│ 2    │ 2    │
│ 3    │ 3    │
│ 4    │ 4    │
│ 5    │ 5    │
│ 6    │ 6    │
└──────┴──────┘
 Disk-1  Disk-2
```

If the Disk - 1 is /dev/sdb and the Disk - 2 is /dev/sdc then,
mdadm -Cv /dev/md0 -n 2 /dev/sdb /dev/sdc -l 1 (to create the RAID - 1 using disk - 1 and disk - 2)

Command	Description
# cat /proc/mdstat	(to check the RAID - 1 is created or not)
# mkfs.ext4 /dev/md0	(to create the ext4 file system on the RAID - 1)
# mkdir /mnt/raid1	(to create the RAID - 1 mount point)
# mount /dev/md0 /mnt/raid1	(to mount RAID - 1 on the mount point)
# mdadm -D /dev/md0	(to see the details of the RAID - 1 partition)
# mdadm /dev/md0 -f /dev/sdb	(to failed the disk manually)
# mdadm /dev/md0 -r /dev/sdb	(to remove the above failed disk)
# mdadm /dev/md0 -a /dev/sdd	(to add the new disk in place of failed disk)
# umount /mnt/raid1	(to unmount the raid file system)
# mdadm --stop /dev/md0	(to stop the RAID - 1 volume)
# mdadm /dev/md0 --add /dev/sde	(to add third disk to the RAID - 1 volume)
# mdadm --grow /dev/md0 --raid_device=3	(to grow the RAID - 1 file system)

34. How to configure RAID - 5 in Linux?

To configure RAID - 5, minimum 3 disks are required and the partition id is "fd".
In every disk approximately 25 - 30% of space is reserved for parity.
Reading and writing is very fast. So, it produces high performance.
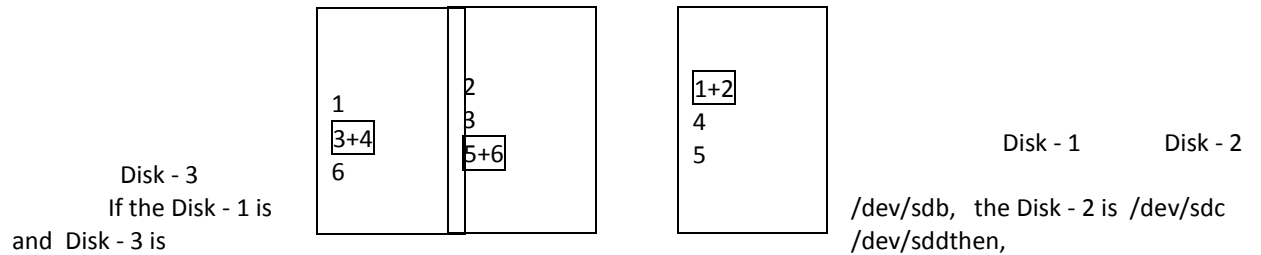This is used Stripping with parity concept.
if one disk is failed we can recover the data using remaining two disks and parity.
If two disks are failed, then we cannot recover the data.
So, there is no redundancy and fault tolerance in RAID - 5.

Example : For example if the data is 1, 2, 3, 4, 5 and 6 then ....

Disk - 3

Disk - 1   Disk - 2

If the Disk - 1 is   /dev/sdb, the Disk - 2 is /dev/sdc
and Disk - 3 is   /dev/sddthen,

    # mdadm -Cv /dev/md0 -n 2 /dev/sdb /dev/sdc -l 5   (to create the RAID - 5 using disks - 1, 2 and 3)

    # cat /proc/mdstat   (to check the RAID - 5 is created or not)
    # mkfs.ext4 /dev/md0   (to create the ext4 file system on the RAID - 5)
    # mkdir /mnt/raid5   (to create the RAID - 5 mount point)
    # mount /dev/md0 /mnt/raid5   (to mount RAID - 5 on the mount point)
    # mdadm -D /dev/md0   (to see the details of the RAID - 5 partition)
    # mdadm /dev/md0   -f /dev/sdb   (to failed the disk manually)
    # mdadm /dev/md0   -r /dev/sdb   (to remove the above failed disk)
    # mdadm /dev/md0   -a /dev/sde   (to add the new disk in place of failed disk)
    # umount /mnt/raid5   (to unmount the raid file system)
    # mdadm --stop /dev/md0   (to stop the RAID - 5 volume)
    # mdadm /dev/md0 --add /dev/sdf   (to add fourth disk to the RAID - 5 volume)
    # mdadm   --grow /dev/md0 --raid_device=4   (to grow the RAID - 5 file system)

35.   **What are the main advantages of RAID - 5**
        RAID - 5 uses Stripping with parity and requires only three disks. Because of Stripping the data reading and
            writing will be fast.And by usingparity we can recover the data if one of the three disks failed. So,
        the main   advantage of RAID - 5 we can get fast writing, reading and also redundancy fault
        tolerance with less expensive.

36.   **How will you troubleshoot if one of the eight disks failed in LVM?**
        First umount the file system and add the new disk with same size of the failed disk to the volume group.
            Then move the data from failed physical volume to newly added physical volume and then
        remove the failed physical volume from the volume group. And finally mount the file system.

37.   **What is pvmove and when it is used in LVM?**
        The pvmove command is used to move the data from failed physical volume to newly added physical
        volume. This command is used when one of the physical volume is failed in the LVM.

38.   **How to inform the client and then troubleshoot if the disk is full?**
        First check which files are accessing more disk space by **#du -h |sort - r** command. if any temporary and
        junk files are present remove them from the disk to make a room for new or updated data. Then inform
        the actual   situation to the client, take the permission from the client to get the lun from storage
        and extend the file system   by adding that lun to the LVM.

39.   **Did you work on storage?**

Actually I did not work on storage but I know the procedure how to export the lun from storage to client using iSCSI target. Then scan that lun at cleint side and add the lun to the LVM. I also know the storage hardware from   Emc square, Netapp and others. And I am dreaming to work on storage, cloud and virtualization.

40. **I have four disks each 1TB in RAID - (1+0). So, total how much disk space can I utilize in that RAID – (1+0)?** RAID - (1+0) means Mirroring + Stripping. It requires 4 disks, ie., 2 disks for mirroring and remaining 2 disks for   stripping. And 5 - 10% disk space is used for superblock information. So, finally we can utilize 2TB - 2TB X 10%   disk space in that RAID - (1+0).

41. **If two disks failed in RAID - (1+0), can we recover the data?**
The RAID - (1+0) requires minimum 4 disks and it uses Mirroring + Stripping. If one disk is failed we can recover the data, but if two disks are failed we cannot recover the data.

42. **How many types of disk space issues can we normally get?**
(i) Disk is full.
(ii) Disk is failing or failed.
(iii) File system corrupted or crashed.
(iv) O/S is not recognizing the remote luns when scanning, ...etc.,

43. **What is a link file and how many types?**
Link file is a short cut file to the original file. Creating and removing (deleting) inks between two files is known   as managing links. There are two types of links files available in Linux.
    (i)   Soft link
    (ii)   Hard link
44. **What is soft link and how to create it?**
Soft link is nothing but a short cut file. If original file is deleted, no use of short cut file. ie., we cannot access the original data   by selecting the link file. Soft link can be applied on both directories and files. These files can be stored in any of the file system. ie., the original file may be in one file system and the link file may be on another file system. If   we edit any file, the link files are also updated automatically. When we create a soft link file, the permissions are   full permissions. The soft link file and the original file inode no's are different. The size of the soft link file is same   as the length of the original file name. The soft link can be created by
ln   -s   <original file or directory><link file or directorywith path>   (to create a soft link)
ln   -s /root/script   /root/Desktop/script   (to create a link file for the script and stored on root Desktop)

45. **What is hard link and how to create it?**
Hard link in nothing but a backup file. If the original file is deleted, there is no effect on hard link file. i.e., we can   access the original file data even though the link file is deleted. Hard links can be applied on files only not on   directories. Hard link files can be stored in the same file system. ie., original and hard link files both should be in   the same file system not on different file systems. The inode no's are same for original and hard link files. If the   original is edited, the updations are applied on both original and hard link files. The size of the hard link file is   same as the size of the original file.

46. **What are the commands to search files and directories?**
To search files and directories there are two commands.
    (i)   # locate
    (ii)   # find

47. Explain the locate command and how to use it?