Managing Partitions & File systems

Partitioning is a means to divide a single hard drive into many logical drives. A partition is a contiguous set of blocks on a drive that are treated as an independent disk. A partition table is an index that relates sections of the hard drive to partitions.

Encapsulate your data. Since file system corruption is local to a partition, you stand to lose only some of your data if an accident occurs.

Disk Partitioning Criteria:

MBR = MASTER BOOT RECORD

P= PRIMARY PARTITION

EXTENDED= EXTENDED PARTITION

L= LOGICAL PARTITION

FREE= FREE SPACE

The Structure of Disk Partition

- On the disk where O/S is installed, will have the first partition as MBR.
- MBR is a Master Boot Record, which contains two important utilities, IPL (Initial Program Loader) and PTI (Partition Table information)
- IPL is responsible for booting the operating the system, because it contains the boot loader.

THE CRITERIA OF DISK PARTITIONING:

- Every disk can have only 3 Primary Partitions.
- **Primary Partition** is a partition which usually holds the operating system. Only one amongst the 3 primary partitions can be active which will be booted by MBR to load the operating system.
- **Extended Partition** is a special type of primary partition which can be subdivided into multiple logical partitions. As there can be only 3 primary partitions per disk, and if the user is required to make further partitions then all the space remaining on the disk should be allocated to extended partition, which can be used to create the logical partitions later. There can be only **one extended partition per disk**.
- **Logical partitions** are the partitions which are created under extended partition, all the space in the extended partition can be used to create any number of logical partitions.

Disk Identification:

Different type of disks will be having different initials in Linux

- IDE drive will be shown as /dev/had
- SCSI drive will be shown as /dev/sda
- Virtual drive will be shown as /dev/vda

FILE SYSTEM:

It is method of storing the data in an organized fashion on the disk. Every partition on the disk except MBR and Extended partition should be assigned with some file system in order to make them store the data. File system is applied on the partition by formatting it with a particular type of file system.

Types of file systems used in RHEL 6:

- The file systems supported in Linux are ext2, ext3 and in RHEL 6 ext4, vfat, etc.

S.NO	EXT2	EXT3	EXT4
1.	Stands for Second Extended File System	Stands for Third Extended File System	Stands for Fouth Extended File System
2.	It was introduced in 1993	It was introduced in 2001	It was introduced in 2008.
3.	Does not have journaling feature.	Supports Journaling Feature.	Supports Journaling Feature.
4.	Maximum File size can be from **16 GB to 2 TB**	Maximum File Size can be from **16 GB to 2 TB**	Maximum File Size can be from **16 GB to 16 TB**
5.	Maximum ext2 file system size can be from **2 TB to 32 TB**	Maximum ext3 file system size can be from **2 TB to 32 TB**	Maximum ext4 file system size is **1 EB (Exabyte). 1 EB = 1024 PB (Petabyte). 1 PB = 1024 TB (Terabyte).**
6.	Cannot convert ext file system to ext2.	You can convert an ext2 file system to ext3 file system directly (without backup/restore).	All previous ext file systems can easily be converted into ext4 file system. You can also mount an existing ext3 f/s as ext4 f/s (without having to upgrade it).

MOUNTING:-

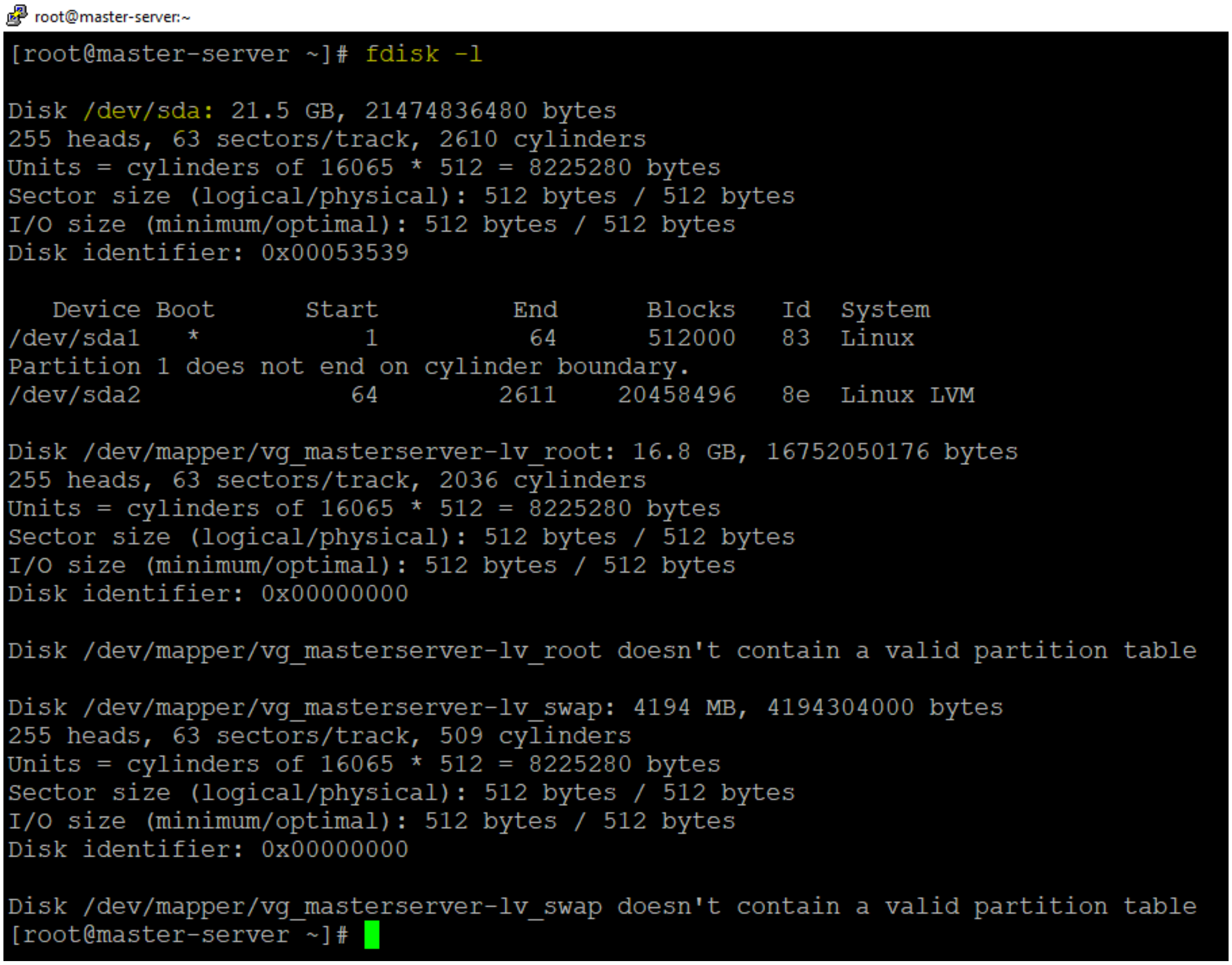
- Attaching a directory to the file system in order to access the partition and its file system is known as mounting.
- The mount point is the directory (usually an empty one) in the currently accessible file system to which a additional file system is mounted.
- The /mnt directory exists by default on all Unix-like systems. It, or usually its subdirectories (such as /mnt/floppy and /mnt/usb), are intended specifically for use as mount points for removable media such as CDROMs, USB key drives and floppy disks.

Files which is related to mounting in Linux:

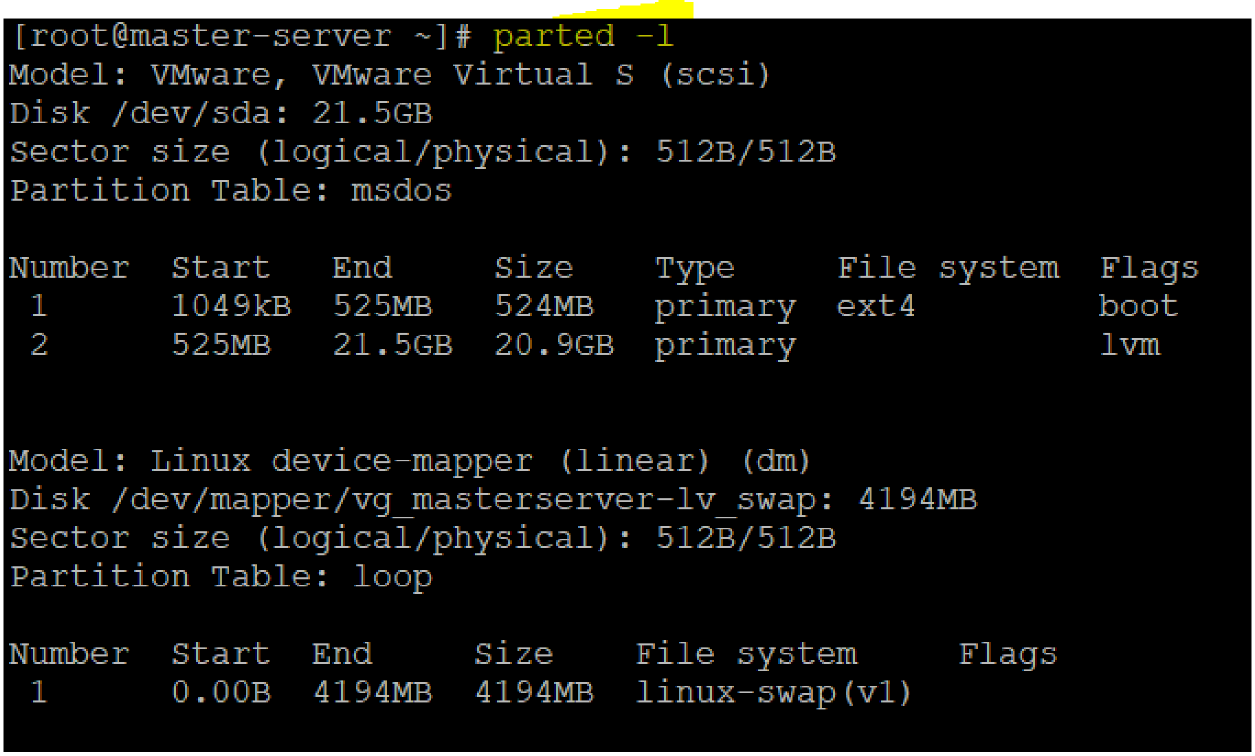
- /etc/mtab is a file which stores the information of all the currently mounted file systems; it is dynamic and keeps changing.
- /etc/fstab is the file which is keeps information about the permanent mount point. If you want to make your mount point permanent, so that it will be mounted even after reboot, then you need to make an appropriate entry in this file.

To view the existing partitions:

```
# fdisk –l  or parted –l
```



```
[root@master-server ~]# fdisk -l

Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00053539

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          64      512000   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2              64        2611    20458496   8e  Linux LVM

Disk /dev/mapper/vg_masterserver-lv_root: 16.8 GB, 16752050176 bytes
255 heads, 63 sectors/track, 2036 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/mapper/vg_masterserver-lv_root doesn't contain a valid partition table

Disk /dev/mapper/vg_masterserver-lv_swap: 4194 MB, 4194304000 bytes
255 heads, 63 sectors/track, 509 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/mapper/vg_masterserver-lv_swap doesn't contain a valid partition table
[root@master-server ~]#
```



```
[root@master-server ~]# parted -l
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sda: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End     Size    Type     File system  Flags
 1      1049kB  525MB   524MB   primary  ext4         boot
 2      525MB   21.5GB  20.9GB  primary               lvm


Model: Linux device-mapper (linear) (dm)
Disk /dev/mapper/vg_masterserver-lv_swap: 4194MB
Sector size (logical/physical): 512B/512B
Partition Table: loop

Number  Start  End     Size    File system     Flags
 1      0.00B  4194MB  4194MB  linux-swap(v1)
```

Partition Administration using fdisk:

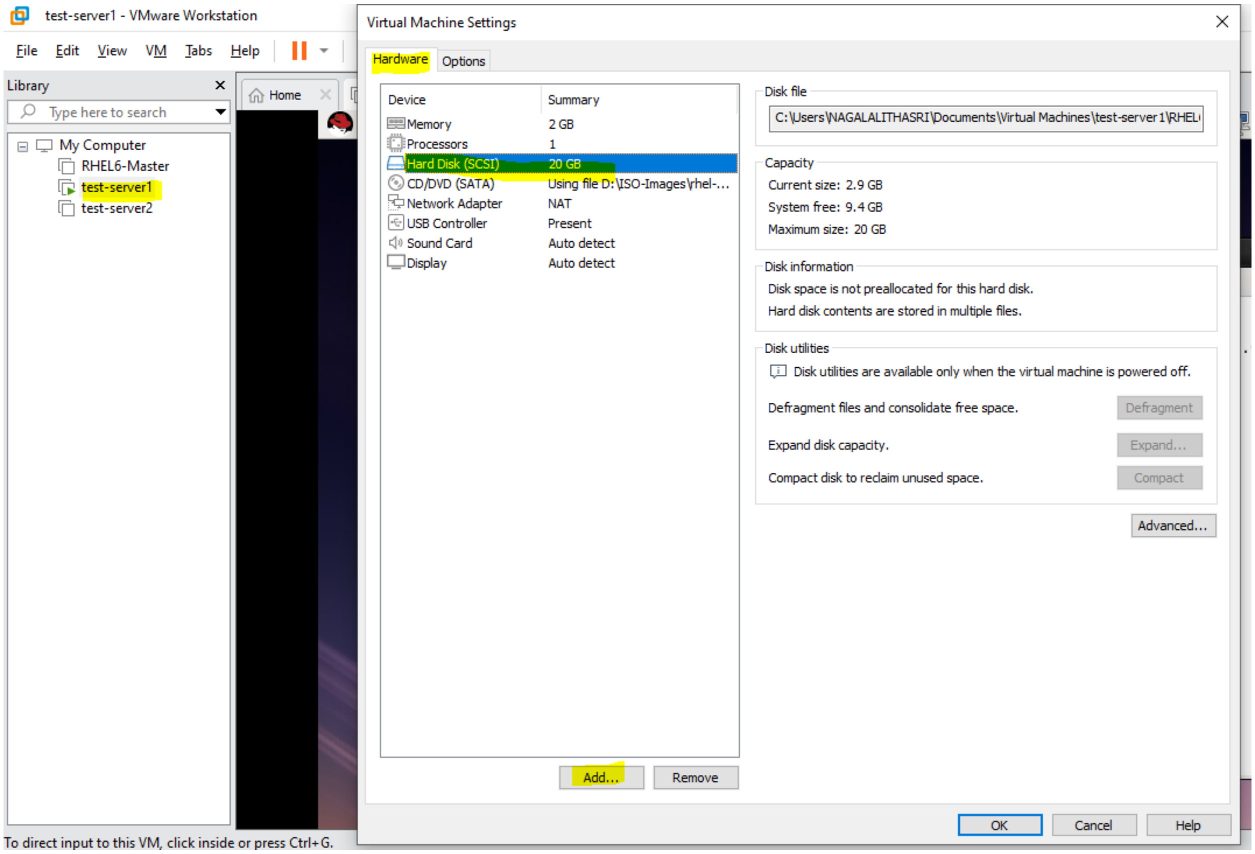
To enter into disk utility, the syntax is

#fdisk <diskname>

#fdisk /dev/sdb

Note: add one disk like below

Goto v-center and select your server -> right click – settings – select harddisk – click on add disk-



Device	Summary
Memory	2 GB
Processors	1
Hard Disk (SCSI)	20 GB
CD/DVD (SATA)	Using file D:\ISO-Images\rhel-...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

Disk file

C:\Users\NAGALALITHASRI\Documents\Virtual Machines\test-server1\RHEL

Capacity

Current size: 2.9 GB

System free: 9.4 GB

Maximum size: 20 GB

Disk information

Disk space is not preallocated for this hard disk.

Hard disk contents are stored in multiple files.

Disk utilities

Disk utilities are available only when the virtual machine is powered off.

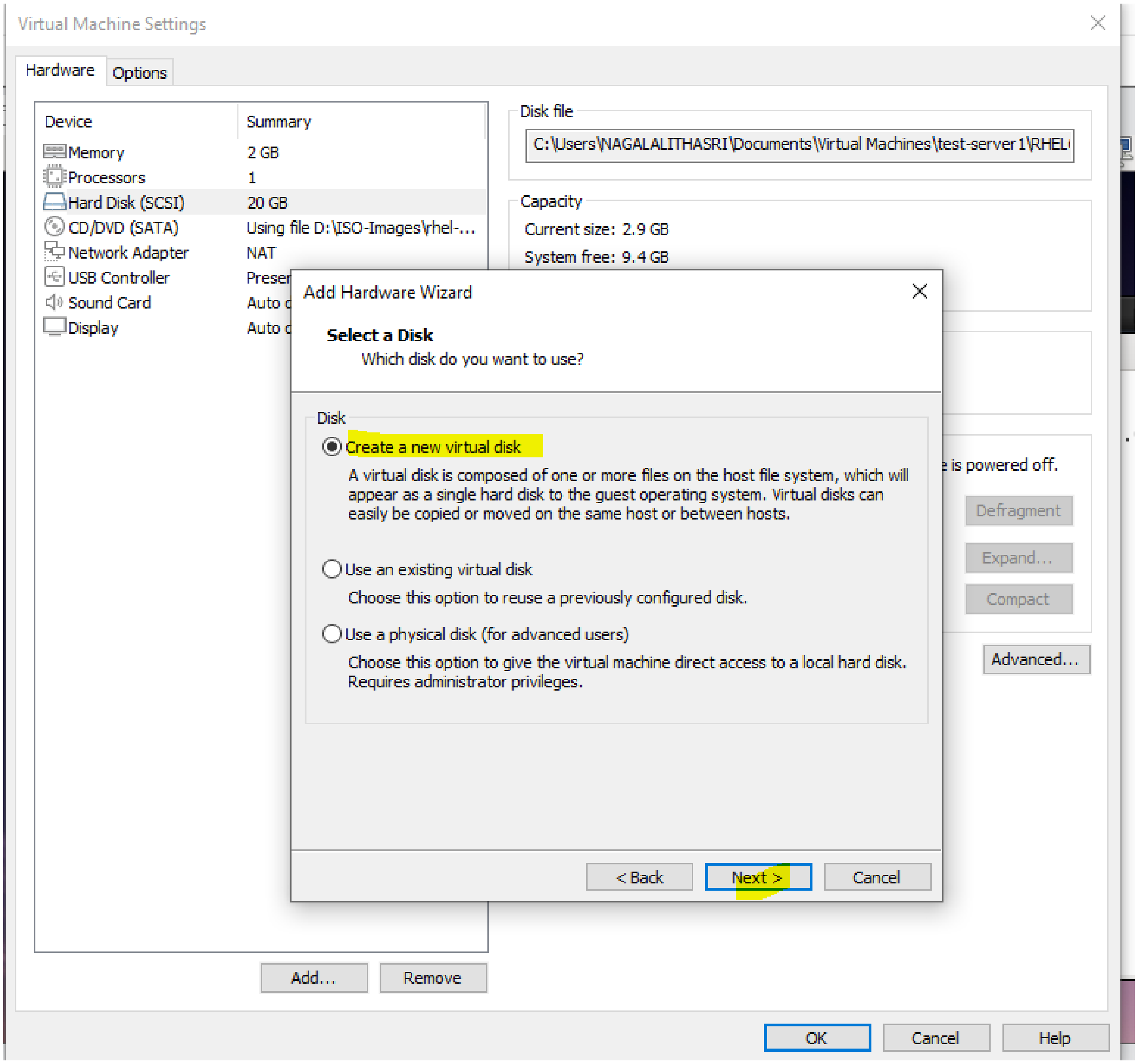
Defragment files and consolidate free space.

Expand disk capacity.

Compact disk to reclaim unused space.

To direct input to this VM, click inside or press Ctrl+G.

Click on add then select – create new disk – give size 20Gib



Device	Summary
Memory	2 GB
Processors	1
Hard Disk (SCSI)	20 GB
CD/DVD (SATA)	Using file D:\ISO-Images\rhel-...
Network Adapter	NAT
USB Controller	Preser
Sound Card	Auto d
Display	Auto d

Disk file

C:\Users\NAGALALITHASRI\Documents\Virtual Machines\test-server1\RHEL

Capacity

Current size: 2.9 GB

System free: 9.4 GB

e is powered off.

Add Hardware Wizard

Select a Disk

Which disk do you want to use?

Disk

● Create a new virtual disk

A virtual disk is composed of one or more files on the host file system, which will appear as a single hard disk to the guest operating system. Virtual disks can easily be copied or moved on the same host or between hosts.

○ Use an existing virtual disk

Choose this option to reuse a previously configured disk.

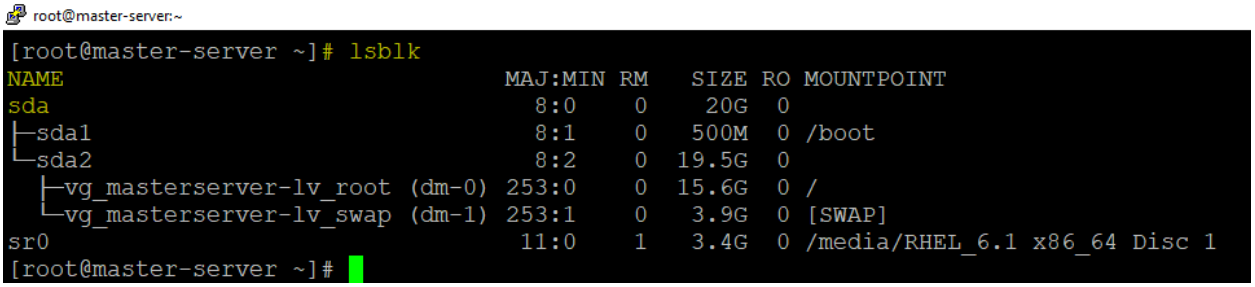
○ Use a physical disk (for advanced users)

Choose this option to give the virtual machine direct access to a local hard disk. Requires administrator privileges.

Note: after adding the disk you cannot see the newly added disk. Check below command

As of now one one disk only showing. So we need to scan the lun.

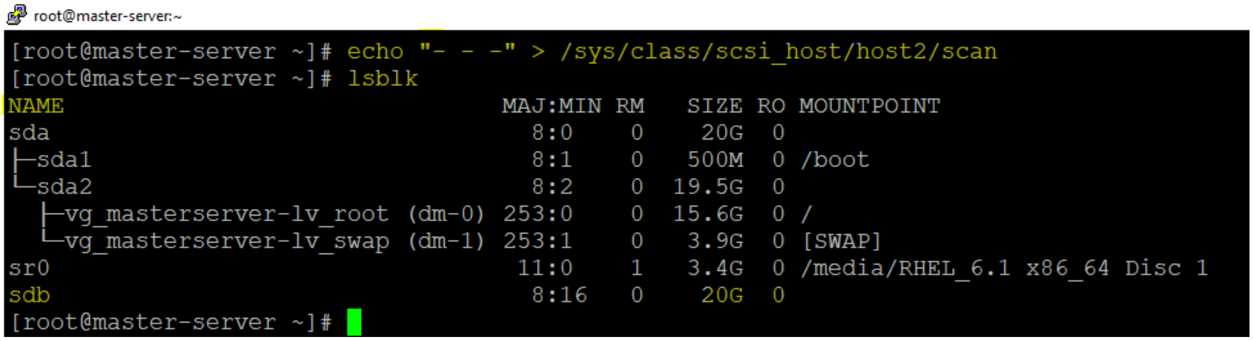
#lsblk



```
[root@master-server ~]# lsblk
NAME                              MAJ:MIN RM   SIZE RO MOUNTPOINT
sda                                 8:0    0    20G  0
├─sda1                              8:1    0   500M  0 /boot
└─sda2                              8:2    0  19.5G  0
  ├─vg_masterserver-lv_root (dm-0) 253:0    0  15.6G  0 /
  └─vg_masterserver-lv_swap (dm-1) 253:1    0   3.9G  0 [SWAP]
sr0                                11:0    1   3.4G  0 /media/RHEL_6.1 x86_64 Disc 1
[root@master-server ~]#
```

We need to scan the lun by using

#echo "- - -" > /sys/class/scsi_host/host2/scan



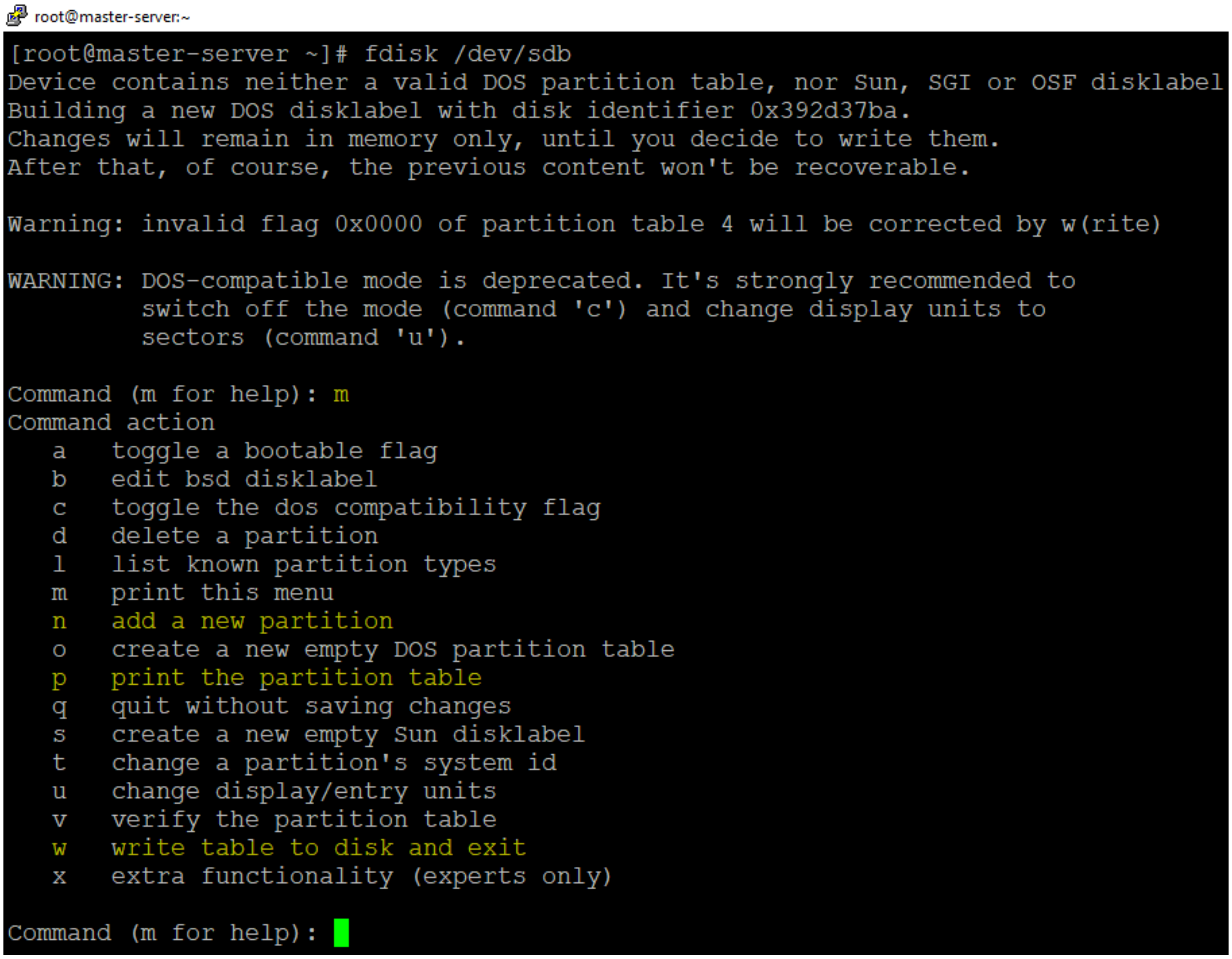
```
[root@master-server ~]# echo "- - -" > /sys/class/scsi_host/host2/scan
[root@master-server ~]# lsblk
NAME                              MAJ:MIN RM   SIZE RO MOUNTPOINT
sda                                 8:0    0    20G  0
├─sda1                              8:1    0   500M  0 /boot
└─sda2                              8:2    0  19.5G  0
  ├─vg_masterserver-lv_root (dm-0) 253:0    0  15.6G  0 /
  └─vg_masterserver-lv_swap (dm-1) 253:1    0   3.9G  0 [SWAP]
sr0                                11:0    1   3.4G  0 /media/RHEL_6.1 x86_64 Disc 1
sdb                                 8:16   0    20G  0
[root@master-server ~]#
```

Now create a partitions

To enter into disk utility, the syntax is

#fdisk <diskname>

#fdisk /dev/sdb

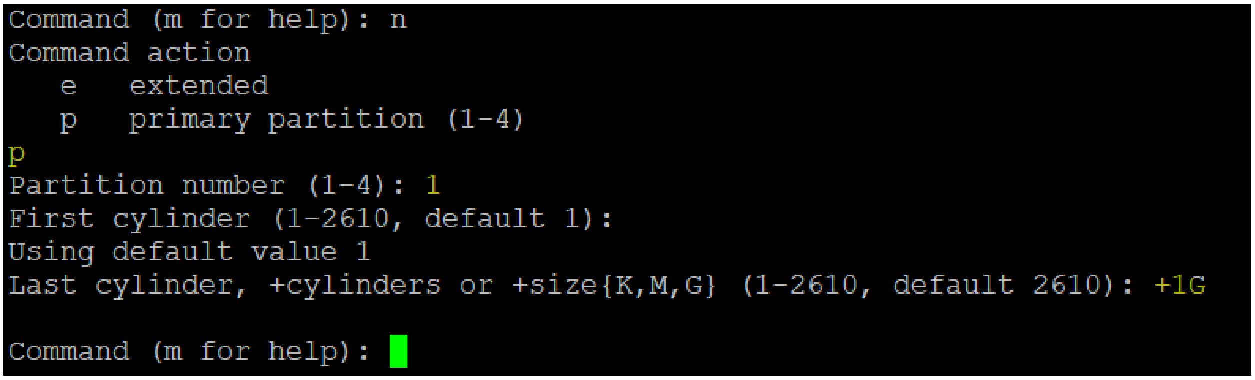


```
[root@master-server ~]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0x392d37ba.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)

Command (m for help):
```

Creating a new partition:

#fdisk /dev/sdb

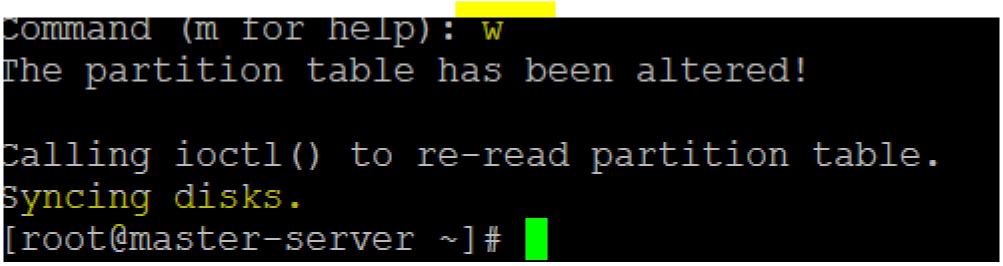
- ➔ Use p to list out the partition information first and
- ➔ Use n to create a new partition.



```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-2610, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-2610, default 2610): +1G

Command (m for help):
```

Saving the partition changes:

Every time you make a partition or delete a partition, the changes made has to be saved using w, otherwise the creation and deletion will not be considered to be happen. For practice purpose you can make any no. of partition and delete it and just quit using q so that it will not be saved.

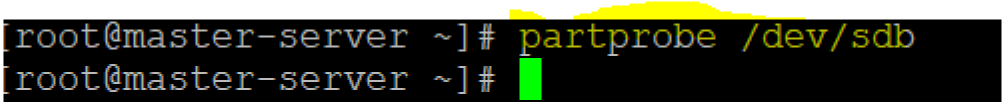


```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
[root@master-server ~]#
```

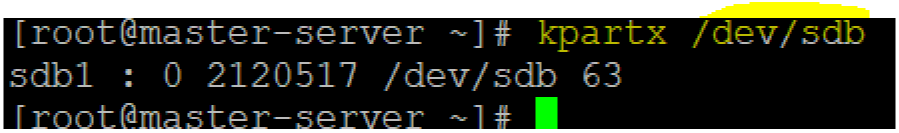
Updating the partition table without restarting the system

After creating or deleting a partition the changes will be effected in the partition table only after the restart of the system. But there is a way to avoid this circumstance. We can use partprobe or partx command to update the partition information without restarting the system

#partprobe /dev/sdb or #partx –a /dev/sda or #kpartx /dev/sda



```
[root@master-server ~]# partprobe /dev/sdb
[root@master-server ~]#
```



```
[root@master-server ~]# kpartx /dev/sdb
sdb1 : 0 2120517 /dev/sdb 63
[root@master-server ~]#
```

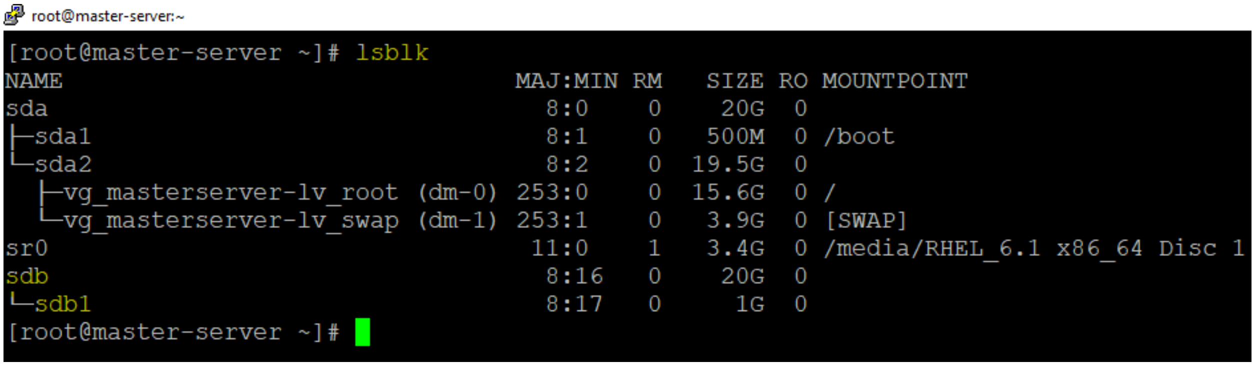
Formatting a partition with ext4 filesystem

After creating a partition we need to assign some file system to it so that we can start storing the data into it. To format a partition the following syntax is used.

# mkfs.ext4 <filesystem type> <partition name>

#mkfs.ext4 /dev/sdb1 (where sdb1 is our newly created partition)

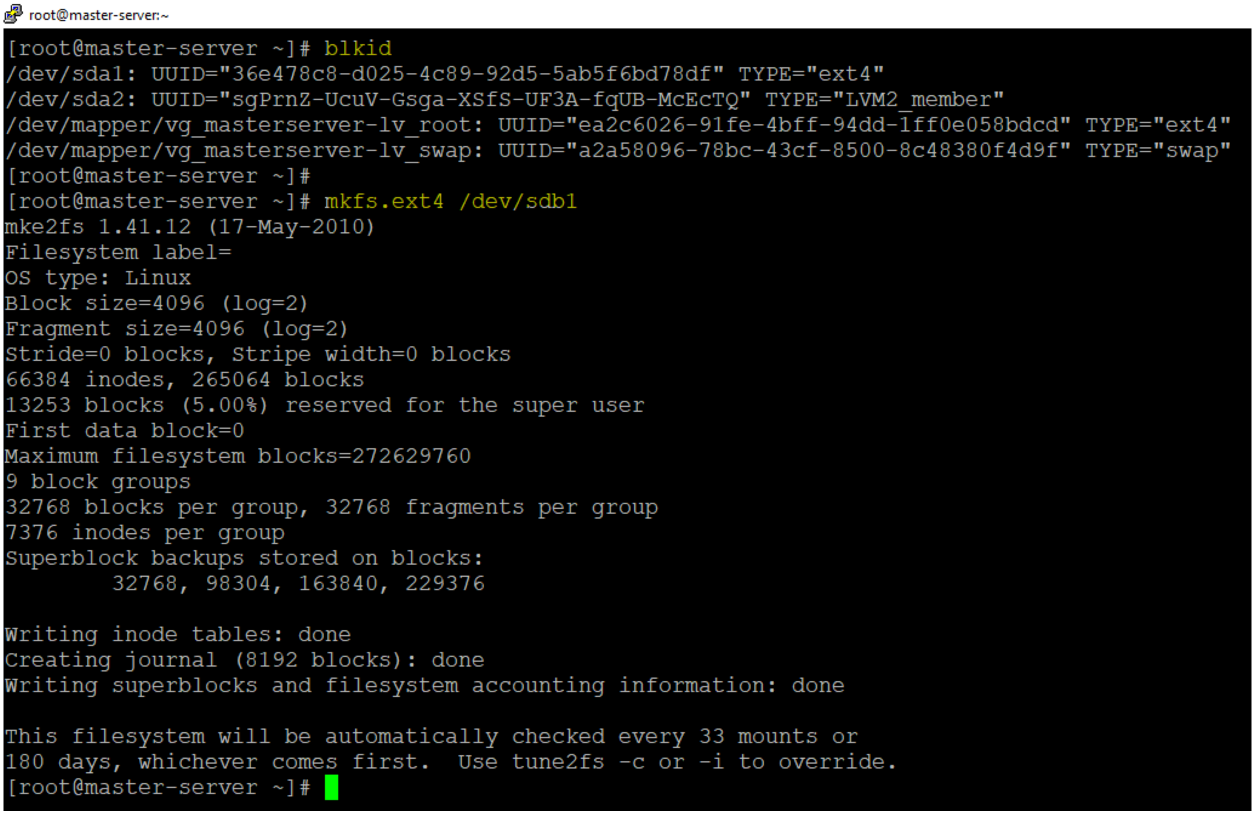
#fdisk –l or #lsblk



```
[root@master-server ~]# lsblk
NAME                                 MAJ:MIN RM   SIZE RO MOUNTPOINT
sda                                    8:0    0    20G  0
├─sda1                                 8:1    0   500M  0 /boot
└─sda2                                 8:2    0  19.5G  0
  ├─vg_masterserver-lv_root (dm-0) 253:0    0  15.6G  0 /
  └─vg_masterserver-lv_swap (dm-1) 253:1    0   3.9G  0 [SWAP]
sr0                                   11:0    1   3.4G  0 /media/RHEL_6.1 x86_64 Disc 1
sdb                                    8:16   0    20G  0
└─sdb1                                 8:17   0     1G  0
[root@master-server ~]#
```

Note: before mkfs check #blkid the disk should check weather is formatted or not then only format the disk

#mkfs.ext4 /dev/sdb1



```
[root@master-server ~]# blkid
/dev/sda1: UUID="36e478c8-d025-4c89-92d5-5ab5f6bd78df" TYPE="ext4"
/dev/sda2: UUID="sgPrnZ-UcuV-Gsga-XSfS-UF3A-fqUB-McEcTQ" TYPE="LVM2_member"
/dev/mapper/vg_masterserver-lv_root: UUID="ea2c6026-91fe-4bff-94dd-1ff0e058bdcd" TYPE="ext4"
/dev/mapper/vg_masterserver-lv_swap: UUID="a2a58096-78bc-43cf-8500-8c48380f4d9f" TYPE="swap"
[root@master-server ~]#
[root@master-server ~]# mkfs.ext4 /dev/sdb1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
66384 inodes, 265064 blocks
13253 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=272629760
9 block groups
32768 blocks per group, 32768 fragments per group
7376 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 33 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@master-server ~]#
```

Mounting a partition:

Mounting is a procedure where we attach a directory to the file system. There are two types of mounting which will be used in Linux or any UNIX.

1. Temporary Mounting   2. Permanent Mounting

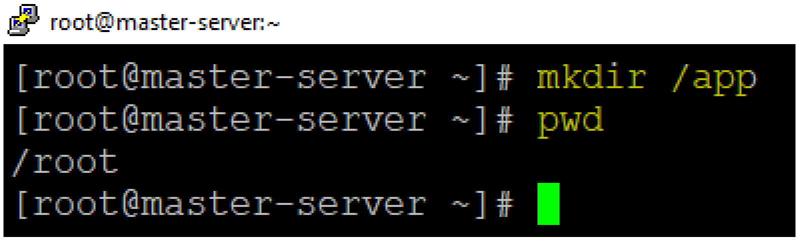
Temporary Mounting:

In a temporary mount point we will create a directory and mount it, but this mount point will last only till the system is up, once it is rebooted the mounting will be lost.

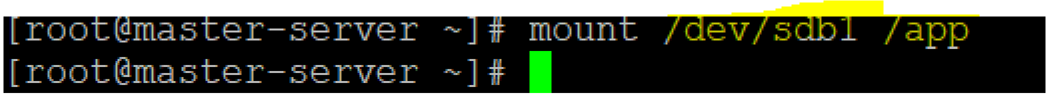
Syntax:

##mount <devicename> <directoryname>

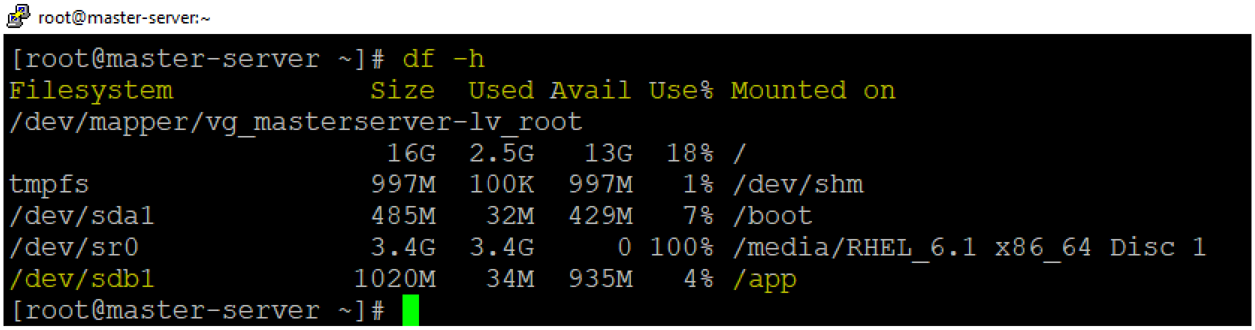
#mount /dev/sdb1 /app1



```
[root@master-server ~]# mkdir /app
[root@master-server ~]# pwd
/root
[root@master-server ~]#
```



```
[root@master-server ~]# mount /dev/sdb1 /app
[root@master-server ~]#
```



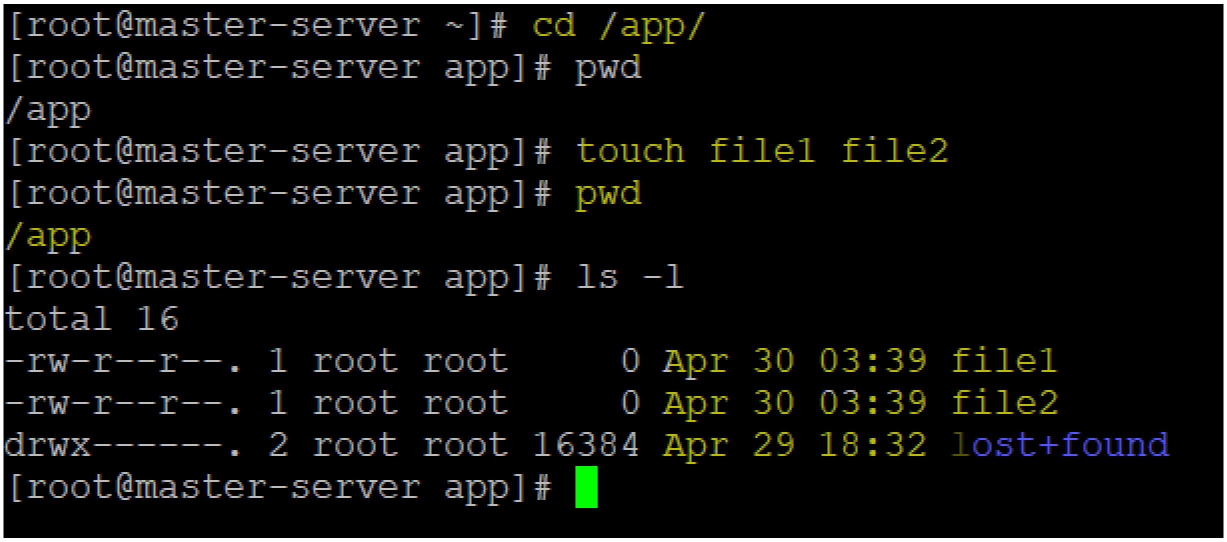
```
[root@master-server ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg_masterserver-lv_root
                       16G  2.5G   13G  18% /
tmpfs                 997M  100K  997M   1% /dev/shm
/dev/sda1             485M   32M  429M   7% /boot
/dev/sr0              3.4G  3.4G     0 100% /media/RHEL_6.1 x86_64 Disc 1
/dev/sdb1            1020M   34M  935M   4% /app
[root@master-server ~]#
```

Now we have successfully mounted the partition we can access it and can store the data

To add the data access the mount point

#cd /app

Add the data and exit the directory



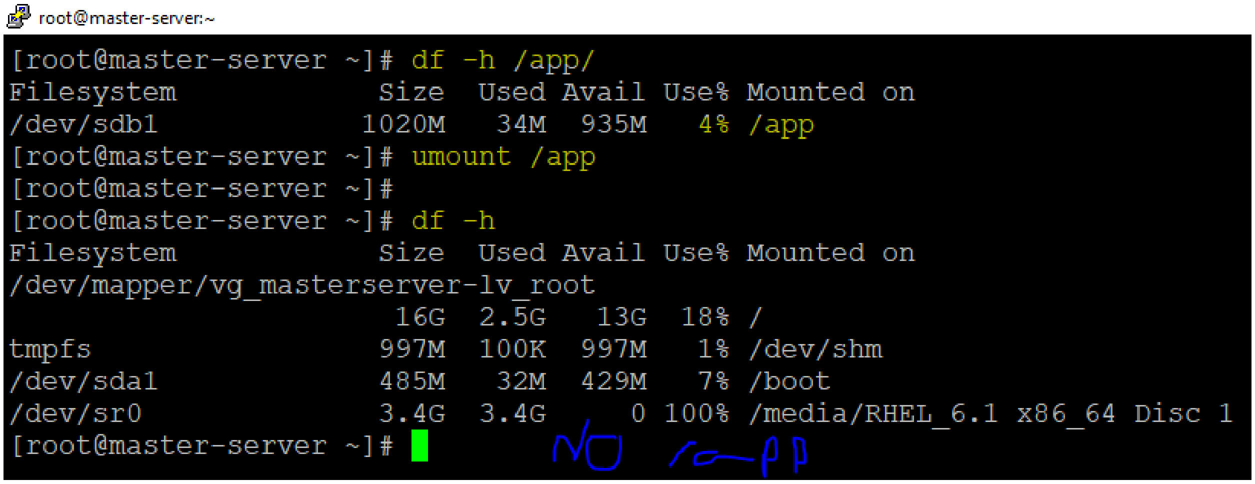
```
[root@master-server ~]# cd /app/
[root@master-server app]# pwd
/app
[root@master-server app]# touch file1 file2
[root@master-server app]# pwd
/app
[root@master-server app]# ls -l
total 16
-rw-r--r--. 1 root root     0 Apr 30 03:39 file1
-rw-r--r--. 1 root root     0 Apr 30 03:39 file2
drwx------. 2 root root 16384 Apr 29 18:32 lost+found
[root@master-server app]#
```

Unmounting a partition

#umount <directory name>

#umount /app

verify it with mount command.



```
[root@master-server ~]# df -h /app/
Filesystem            Size  Used Avail Use% Mounted on
/dev/sdb1            1020M   34M  935M   4% /app
[root@master-server ~]# umount /app
[root@master-server ~]#
[root@master-server ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg_masterserver-lv_root
                       16G  2.5G   13G  18% /
tmpfs                 997M  100K  997M   1% /dev/shm
/dev/sda1             485M   32M  429M   7% /boot
/dev/sr0              3.4G  3.4G     0 100% /media/RHEL_6.1 x86_64 Disc 1
[root@master-server ~]#
```

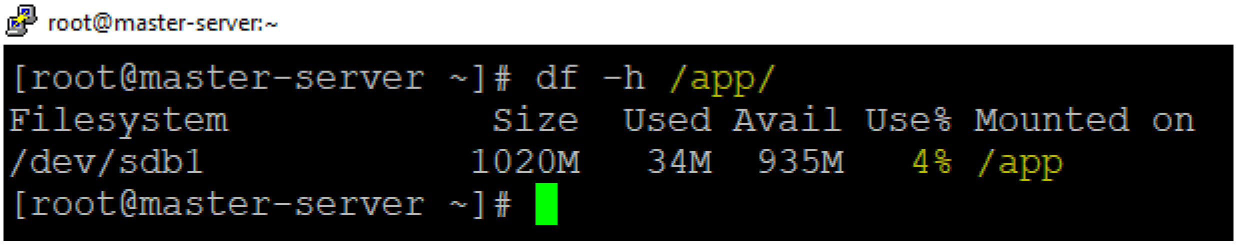
NO /app

Note: Till now we called as temporary mount point

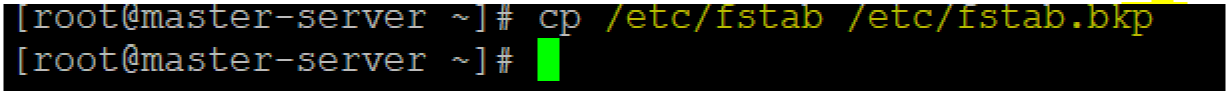
Permanent Mounting:

Permanent mounting procedure is exactly same like temp mounting, but here we will update the /etc/fstab file with the mounting details, so that it will be mounted even after the system is reboot.

#vi /etc/fstab

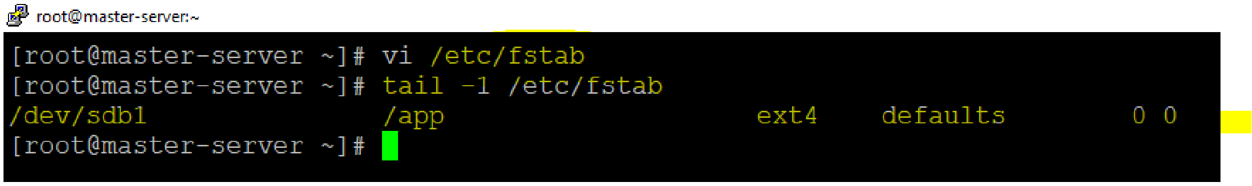


```
root@master-server:~
[root@master-server ~]# df -h /app/
Filesystem            Size  Used Avail Use% Mounted on
/dev/sdb1            1020M   34M  935M   4% /app
[root@master-server ~]#
```

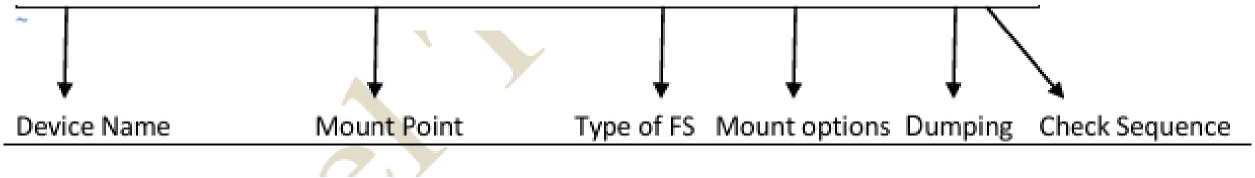


```
[root@master-server ~]# cp /etc/fstab /etc/fstab.bkp
[root@master-server ~]#
```

Note: Always take backup a file when you are doing any changes.



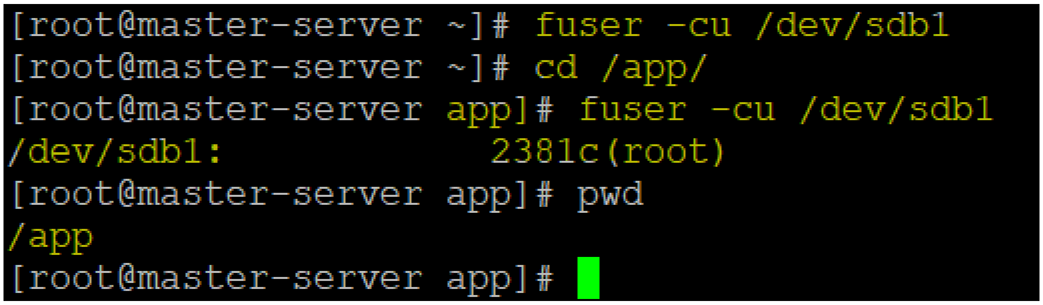
```
root@master-server:~
[root@master-server ~]# vi /etc/fstab
[root@master-server ~]# tail -1 /etc/fstab
/dev/sdb1               /app                    ext4    defaults        0 0
[root@master-server ~]#
```



You can now access the directory and add, delete or modify the contents and can also unmount the file system at any point.

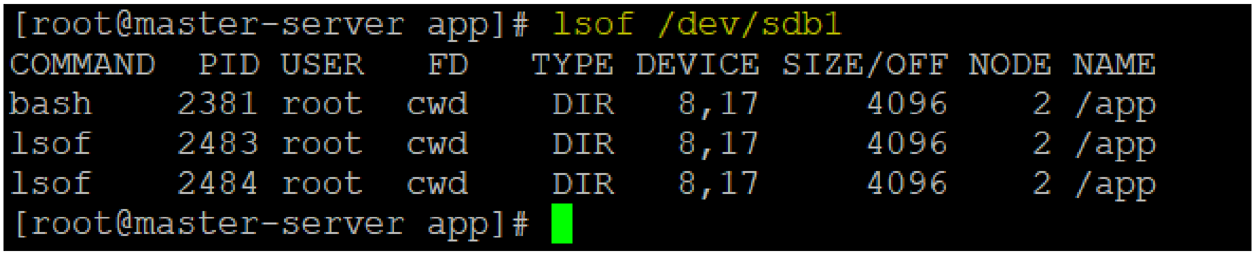
Sometimes a directory reflects error while unmouting, the possible causes for it are

- ➔ You are in the same directory and trying to unmount it. Check with pwd command
- ➔ Some users are present in the directory and using the contents in it.
- ➔ Check with fuser –cu /dev/sdb1



```
[root@master-server ~]# fuser -cu /dev/sdb1
[root@master-server ~]# cd /app/
[root@master-server app]# fuser -cu /dev/sdb1
/dev/sdb1:            2381c(root)
[root@master-server app]# pwd
/app
[root@master-server app]#
```

- ➔
- ➔ Check for the files which are open with #lsof /dev/sdb1



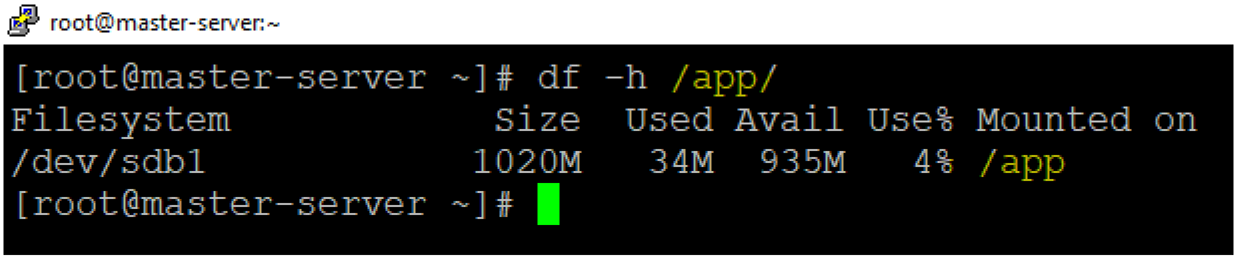
```
[root@master-server app]# lsof /dev/sdb1
COMMAND  PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
bash    2381 root  cwd    DIR   8,17     4096    2 /app
lsof    2483 root  cwd    DIR   8,17     4096    2 /app
lsof    2484 root  cwd    DIR   8,17     4096    2 /app
[root@master-server app]#
```

- Kill the open connections using fuser –ck /kernel/hello where hello is the file which is open.
- Now you can use umount command to unmount the file system.

To view the usage information of mounted partition:

To view the usage information of mounted partition use the command df –h

#df –h



```
[root@master-server ~]# df -h /app/
Filesystem            Size  Used Avail Use% Mounted on
/dev/sdb1            1020M   34M  935M   4% /app
[root@master-server ~]#
```

To view the size of a file or directory

To view the size of the file or directory uses the command du –h file or directory name.