The black screenshot is Debian and the gray is CentOS.

Recapitulation:

```
$ lsblk -f                                  # list in detail block devices (total disks & roms)
$ df -h                                     # See mounted devices
$ mount                                     # See all mounted devices
# fdisk /dev/sdb1                           # Create partition /dev/sdb1 for MBR
# gdisk /dev/sdc                            # Same as “fdisk” but only for GPT
# parted /dev/sdc                           # Work both for MBR and GPT
# mkfs.ext4 /dev/sdb1                       # Create EXT4 file system
# mkfs.xfs /dev/sdb1                        # Create XFS file system
# mkfs.btrfs /dev/sdb1                      # Create BTRFS file system
# mkswap /dev/sdb1                          # Create SWAP file system
# swapon /dev/sdb1                          # Turn on the SWAP
# e2label /dev/sdb1 Storage1                # change the label's disk, only for ext4
# xfs_admin -L Storage1 /dev/sdb2           # Rename label's disk for xfs
# btrfs filesystem label /dev/sdb2 Storage2 # Rename btrfs's disk label
# mount /dev/sdb1 /home/user1/Storage       # Temporary mount the file system in directory
# umount /dev/sdb1                          # Unmount the disk
# nano /etc/fstab                           # Edit for permanent mount
```

LINUX STORAGE

```
$ lsblk                                     # list block devices (total disk and cd rom)
# lsblk -o name,mountpoint,size,uuid        # list the name, mountpoint,size, and universal ID
# e2label /dev/sda1 label1                  # Set the label of sda1 as label1
$ cat /etc/fstab                            # see the file system table
# fdisk /dev/sdc                            # Create partitions for MBR
# gdisk /dev/sdc                            # Same as “fdisk” but only for GPT
# parted /dev/sdc                           # Work both for MBR and GPT
```

BACKUPS

RAID 1 – Mirroring
- Any written data to Disk 1 is mirrored over to Disk 2
- If loosing Disk 1, everything is in Disk 2

RAID 0 – Striping
- Break the data and spread in over Disk 1 and 2
- If loosing Disk 1 or 2, will loose the data

CREATING PARTITIONS

Hard drives were usually diced up into partitions: Take one hard drive and divide it into multiple virtual hard drives

Types of partition managements:

- Master Boot Record (MBR): Record where those partitions are :
  -- Up to 4 partitions (including SWAP, Boot file, and Rest of the hard drives)
  -- No large hard drive supported
  -- Used by old linux on embaded systems (Web cameras, routers, ...)
- Guid (Globally unique Identifier) Partition Table (GPT):
  -- Identify partitions
  -- Up to 128 partitions
  -- Most recent linux use GPT

- List the block devices to see all hard drives
- Then, see if /dev/sdb is empty with **fdisk**, only for MBR:

```
root@debian10-server:~# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0   16G  0 disk
├─sda1   8:1    0   15G  0 part /
├─sda2   8:2    0    1K  0 part
└─sda5   8:5    0  975M  0 part [SWAP]
sdb      8:16   0    8G  0 disk
sr0     11:0    1 1024M  0 rom
root@debian10-server:~# fdisk -l /dev/sdb
Disk /dev/sdb: 8 GiB, 8589934592 bytes, 16777216 sectors
Disk model: VBOX 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: dos
Disk identifier: 0x7407d534
root@debian10-server:~# _
```

- Create partitions of 5GB with **fdisk** by clicking “n” for new

```
root@debian10-server:~# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.33.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-16777215, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-16777215, default 16777215): +5G

Created a new partition 1 of type 'Linux' and of size 5 GiB.
```

- Create a new partition for the rest of the storage, set everything as default.
- Then, click “p” for printing, “CTRL+C” to exit without save, and “w” to write and exit.

```
Command (m for help): n
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (2-4, default 2):
First sector (10487808-16777215, default 10487808):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (10487808-16777215, default 16777215):

Created a new partition 2 of type 'Linux' and of size 3 GiB.

Command (m for help): p
Disk /dev/sdb: 8 GiB, 8589934592 bytes, 16777216 sectors
Disk model: VBOX 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: dos
Disk identifier: 0x7407d534

Device     Boot    Start      End  Sectors Size Id Type
/dev/sdb1           2048 10487807 10485760   5G 83 Linux
/dev/sdb2       10487808 16777215  6289408   3G 83 Linux

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@debian10-server:~#
```

After running lsblk again, we can see the two partition sdb1 and sdb2

```
stella@debian10-server:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0   16G  0 disk
├─sda1   8:1    0   15G  0 part /
├─sda2   8:2    0    1K  0 part
└─sda5   8:5    0  975M  0 part [SWAP]
sdb      8:16   0    8G  0 disk
├─sdb1   8:17   0    5G  0 part
└─sdb2   8:18   0    3G  0 part
sr0     11:0    1 1024M  0 rom
stella@debian10-server:~$ _
```

CREATING LINUX FILE SYSTEM:

ext: Extended file system
- came in 1990s
- Default file system for most Linux
- Support very large files
- It's a journaling file system: you don't loose the file if you loose power but get the previous version
- Most Linux use Ext4

xfs: Extended File System:
- Better for numerous small files
- Default for RedHad
- Originally designed for multi-media

btrfs: B-tree file system:
- Used to divide the data in more than one hard drive
- Used to resize a partitions
- Used to divide the partition in more than one server like a distributed file system
- Not as reliable as ext and xfs
- Less common

- It is not possible to convert from xfs to ext and vice versa

Create a file system
 # mkfs.ext4 /dev/sdb1 # for ext4
 # mkfs.xfs /dev/sdb2 # for xfs

```
[root@localhost ~]# mkfs.xfs /dev/sdb1
meta-data=/dev/sdb1              isize=512    agcount=4, agsize=327680 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=1310720, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@localhost ~]#
```

mkfs.btrfs /dev/sdc1 # Make btrfs file system

```
[root@localhost ~]# mkfs.btrfs /dev/sdb2
btrfs-progs v4.9.1
See http://btrfs.wiki.kernel.org for more information.

Label:              (null)
UUID:               8aa7631e-d57f-4cf9-a9b6-a196cee7fb02
Node size:          16384
Sector size:        4096
Filesystem size:    3.00GiB
Block group profiles:
  Data:             single            8.00MiB
  Metadata:         DUP             153.50MiB
  System:           DUP               8.00MiB
SSD detected:       no
Incompat features:  extref, skinny-metadata
Number of devices:  1
Devices:
   ID        SIZE  PATH
    1     3.00GiB  /dev/sdb2

[root@localhost ~]#
```

mkswap /dev/sdc2 # Make swap file system
swapon /dev/sdc2 # turn on the swap

```
root@debian10-server:~# mkswap /dev/sdb2
Setting up swapspace version 1, size = 3 GiB (3220172800 bytes)
no label, UUID=5e37cffc-ee05-4c8b-9052-937964f51638
```

```
root@debian10-server:~# swapon /dev/sdb2
root@debian10-server:~#
```

# e2label /dev/sdb1 Storage1 # change the label = rename the disk for ext4
# reboot # Reboot the computer to see changes
$ lsblk -f # list with the label and file system type

```
stella@debian10-server:~$ lsblk -f
NAME   FSTYPE LABEL    UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
├─sda1 ext4           4dc8aae3-7a92-4ff5-98b1-2eb8b1b92279   12.5G    10% /
├─sda2
└─sda5 swap           19398479-dc95-4413-af5a-76deb0596de3                [SWAP]
sdb
├─sdb1 ext4   Storage1 0784fcd0-b684-4680-b8a7-dec5c7f9ae7b
└─sdb2 swap           5e37cffc-ee05-4c8b-9052-937964f51638
sr0
stella@debian10-server:~$
```

xfs_admin -L Storage2 /dev/sdb2 # Rename the disk for xfs

```
[root@localhost ~]# xfs_admin -L Storage2 /dev/sdb1
writing all SBs
new label = "Storage2"
[root@localhost ~]# lsblk -f
NAME   FSTYPE LABEL    UUID                                 MOUNTPOINT
sda
├─sda1 xfs            d396f709-e8c6-488b-8f87-5bbf521f6067 /boot
├─sda2 swap           635208cb-dbf6-4c96-94e4-11d4609023a5 [SWAP]
└─sda3 xfs            45f67943-5f70-4744-be4b-aa770694437d /
sdb
├─sdb1 xfs    Storage2 fa768867-0562-4b2c-8d37-608ae1759ca0
└─sdb2 btrfs          8aa7631e-d57f-4cf9-a9b6-a196cee7fb02
sr0
[root@localhost ~]#
```

btrfs filesystem label /dev/sdb2 Storage2 #Rename btrfs label

```
[root@localhost ~]# btrfs filesystem label /dev/sdb2 Storage_btrfs
[root@localhost ~]# lsblk -f
NAME   FSTYPE LABEL         UUID                                 MOUNTPOINT
sda
├─sda1 xfs                  d396f709-e8c6-488b-8f87-5bbf521f6067 /boot
├─sda2 swap                 635208cb-dbf6-4c96-94e4-11d4609023a5 [SWAP]
└─sda3 xfs                  45f67943-5f70-4744-be4b-aa770694437d /
sdb
├─sdb1 xfs    Storage2      fa768867-0562-4b2c-8d37-608ae1759ca0
└─sdb2 btrfs  Storage_btrfs 8aa7631e-d57f-4cf9-a9b6-a196cee7fb02
sr0
[root@localhost ~]#
```

MOUNTING FILE SYTEMS

mount /dev/sdb1 [directory mount point] # mount temporary the file system in a directory
df -h # see if it is mounted
lsblk -f # also see if it is mounted
mount # also see all mounted devices but it is messy

```
root@debian10-server:~# mount /dev/sdb1 /home/stella/Storage1/
root@debian10-server:~#
root@debian10-server:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            480M     0  480M   0% /dev
tmpfs            99M  1.7M   98M   2% /run
/dev/sda1        15G  1.5G   13G  11% /
tmpfs           494M     0  494M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           494M     0  494M   0% /sys/fs/cgroup
tmpfs            99M     0   99M   0% /run/user/1000
/dev/sdb1       4.9G   24K  4.6G   1% /home/stella/Storage1
root@debian10-server:~#
```

```
[root@localhost ~]# mkdir /mnt/Storage2
[root@localhost ~]# mkdir /mnt/Storage_btrfs
[root@localhost ~]#
[root@localhost ~]# mount /dev/sdb1 /mnt/Storage2
[root@localhost ~]#
[root@localhost ~]# mount /dev/sdb2 /mnt/Storage_btrfs
[root@localhost ~]#
[root@localhost ~]# lsblk -f
NAME   FSTYPE LABEL         UUID                                 MOUNTPOINT
sda
├─sda1 xfs                  d396f709-e8c6-488b-8f87-5bbf521f6067 /boot
├─sda2 swap                 635208cb-dbf6-4c96-94e4-11d4609023a5 [SWAP]
└─sda3 xfs                  45f67943-5f70-4744-be4b-aa770694437d /
sdb
├─sdb1 xfs    Storage2      fa768867-0562-4b2c-8d37-608ae1759ca0 /mnt/Storage2
└─sdb2 btrfs  Storage_btrfs 8aa7631e-d57f-4cf9-a9b6-a196cee7fb02 /mnt/Storage_btrfs
sr0
[root@localhost ~]#
```

umount /dev/sdb1 # Unmount the disk

<u>Permanent mount:</u>
- Add this in /etc/fstab
/dev/sdb1 [mount point] [file system type] [option] [dump_Command] [critical disk]

[option] ro for read-only
[if critical_disk] = 1 like / or /boot, if simple disk then 0

```
/dev/sdb1       /home/stella/Storage1   ext4    defaults        0       0
```

```
/dev/sdb1       /mnt/Storage2           xfs     ro      0       0
/dev/sdb2       /mnt/Storage_btrfs      btrfs   ro      0       0
```

 # mount -a # to update the mounts
mount # see all mounted devices

```
/dev/sdb1 on /mnt/Storage2 type xfs (ro,relatime,seclabel,attr2,inode64,noquota)
/dev/sdb2 on /mnt/Storage_btrfs type btrfs (ro,relatime,seclabel,space_cache,subvolid=5,subvol=/)
[root@localhost ~]#
```

<u>Linux Filesystem Hierarchy Standard (FHS)</u>

/bin a link to /usr/bin for binaries: the applications that we run
/sbin a link to /usr/sbin: s stands for system containing binaries which require system boot
/boot for the boot files for system boot
/etc for configuration files like software and services
/opt for optional software
/dev for devices
/home for home folders
/lib for libraries
/media for temporary mounts
/proc for process
/run for temporary mounts
/srv for servers like webservers
/sys for boot files
/tmp for temporary storage
/var fro various storage

$ which zip # to know where it is installed
$ whereis zip # to know where is its location