Disk Partitions

lsblk -> To list all block devices (or) storage devices attached in server. For SATA & iSCSI disks, it uses ‘s’. For NVME disks, it uses same word ‘nvme’. For IDE disks, it uses ‘h’. Disk file type is block (‘b’ in permissions)

fdisk -l -> To get detail about all disks attached to server. Used to manage MBR disk.

fdisk -l /dev/sda -> To get detail about a particular disk attached in server. Here it is for disk sda. Used to manage MBR disk.

dmesg -> To get information related to disks. Like during boot time. Ex- dmesg | grep sda, dmesg |grep nvme etc.

lshw -> To get detail about attached hardware in server.

badblocks -ws /dev/sda -> To override data in storage to avoid original data recover in case of sale of storage.

dd if=/dev/zero of=/dev/sda -> Will replace all data with zeros.

Note: We can add SATA & SCSI drive in on the go i.e in running server. We can’t do with NVME & IDE type.

Note: To scan for newly added disks on running server during run-

ls /sys/class/scsi_host/ | while read host ; do echo "- - -" > /sys/class/scsi_host/$host/scan ; done

Note: MBR disk can have maximum of 4 primary partitions. Or 3 primary and 1 extended partition. Extended partition can further have 15 logical partitions. Fisk command is Used to manage MBR disk.

Note: GPT disk can have maximum of 128 primary partitions. gdisk command is Used to manage GPT disk.

Note: partprobe command is Used to manage both GPT & MBR disk.

cat /proc/partitions -> Detail about partitions on any disk along with disk.

MBR Disk Partition Steps Using fdisk:-

1. Check all blocks available.

```
[root@rhel9-server /]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda           8:0    0   10G  0 disk
└─sda1        8:1    0    1G  0 part
sdb           8:16   0   10G  0 disk
sdc           8:32   0   10G  0 disk
sdd           8:48   0   10G  0 disk
sr0          11:0    1  7.9G  0 rom  /repodata
nvme0n1     259:0    0  100G  0 disk
├─nvme0n1p1 259:1    0    2G  0 part /boot
├─nvme0n1p2 259:2    0   50G  0 part /
└─nvme0n1p3 259:3    0    4G  0 part [SWAP]
[root@rhel9-server /]#
```

2. Creating MBR partition in disk sda.

```
[root@rhel9-server /]# fdisk /dev/sda

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help):
```

3. Add a new partition, use keyword n.

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

4. Type p for primary partition. Select no. 2 for new primary partition as 1 is already created. Keep first sector as it is & hit enter, use +2G to create 2GB partition & hit enter. Type will be default ‘Linux’. We can change it as per our requirement.

```
Select (default p): p
Partition number (2-4, default 2): 2
First sector (2099200-20971519, default 2099200):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2099200-20971519, default 20971519): +2G

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

Command (m for help):
```

5. Type w and hit enter to save this partition.

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

[root@rhel9-server /]#
```

6. To verify, use lsblk command.

```
[root@rhel9-server /]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda           8:0    0   10G  0 disk
├─sda1        8:1    0    1G  0 part
└─sda2        8:2    0    2G  0 part
sdb           8:16   0   10G  0 disk
sdc           8:32   0   10G  0 disk
sdd           8:48   0   10G  0 disk
sr0          11:0    1  7.9G  0 rom  /repodata
nvme0n1     259:0    0  100G  0 disk
├─nvme0n1p1 259:1    0    2G  0 part /boot
├─nvme0n1p2 259:2    0   50G  0 part /
└─nvme0n1p3 259:3    0    4G  0 part [SWAP]
[root@rhel9-server /]#
```

7. Next, we need to format this partition using xfs file system.

```
[root@rhel9-server /]# mkfs.xfs /dev/sda2
meta-data=/dev/sda2              isize=512    agcount=4, agsize=131072 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=524288, 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@rhel9-server /]#
```

8. Next, mount this partition in /disks/waikiki_data_drive directory.

```
[root@rhel9-server /]# mount /dev/sda2 /disks/waikiki_data_drive/
[root@rhel9-server /]#
```

9. To confirm it, use df -Th.

```
[root@rhel9-server /]# df -Th
Filesystem     Type      Size  Used Avail Use% Mounted on
devtmpfs       devtmpfs  856M     0  856M   0% /dev
tmpfs          tmpfs     875M     0  875M   0% /dev/shm
tmpfs          tmpfs     350M  5.2M  345M   2% /run
/dev/nvme0n1p2 xfs        50G  2.0G   48G   4% /
/dev/sr0       iso9660   8.0G  8.0G     0 100% /repodata
/dev/nvme0n1p1 xfs       2.0G  222M  1.8G  11% /boot
tmpfs          tmpfs     175M     0  175M   0% /run/user/0
tmpfs          tmpfs     175M     0  175M   0% /run/user/1000
/dev/sda2      xfs       2.0G   47M  2.0G   3% /disks/waikiki_data_drive
[root@rhel9-server /]#
```

10. To unmount this partition, use umount command shown in screenshot.

```
[root@rhel9-server /]# umount /dev/sda2
[root@rhel9-server /]#
```

11. Now verify it again using df -Th.

```
[root@rhel9-server /]# df -Th
Filesystem     Type      Size  Used Avail Use% Mounted on
devtmpfs       devtmpfs  856M     0  856M   0% /dev
tmpfs          tmpfs     875M     0  875M   0% /dev/shm
tmpfs          tmpfs     350M  5.2M  345M   2% /run
/dev/nvme0n1p2 xfs        50G  2.0G   48G   4% /
/dev/sr0       iso9660   8.0G  8.0G     0 100% /repodata
/dev/nvme0n1p1 xfs       2.0G  222M  1.8G  11% /boot
tmpfs          tmpfs     175M     0  175M   0% /run/user/0
tmpfs          tmpfs     175M     0  175M   0% /run/user/1000
[root@rhel9-server /]#
```

12. This mount will be temporary & get remove automatically once server reboot. To make it permanent, we need to add its entry in fstab file in etc directory in two ways.

(a) Using partition name.

```
#
# /etc/fstab
# Created by anaconda on Mon Sep 26 10:25:48 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=4bec8248-9eb3-48da-902d-b0b7c7e10d16 /                       xfs     defaults        0 0
UUID=1b97b83e-8c41-4cd9-a9e6-894073b7299f /boot                   xfs     defaults        0 0
UUID=89f90bba-41f6-4381-941e-8b8d7dc8b66e none                    swap    defaults        0 0
/dev/sr0                                  /repodata               iso9660 defaults        0 0
/dev/sda2                        /disks/waikiki_data_drive        xfs     defaults        0 0
```

(b) Using Partition UUID.

```
#
# /etc/fstab
# Created by anaconda on Mon Sep 26 10:25:48 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=4bec8248-9eb3-48da-902d-b0b7c7e10d16 /                                 xfs     defaults        0 0
UUID=1b97b83e-8c41-4cd9-a9e6-894073b7299f /boot                             xfs     defaults        0 0
UUID=89f90bba-41f6-4381-941e-8b8d7dc8b66e none                              swap    defaults        0 0
/dev/sr0                                  /repodata                         iso9660 defaults        0 0
UUID=49c4d6de-3156-4900-9967-a46f267db90c /disks/waikiki_data_drive         xfs     defaults        0 0
```

To get UUID, use two commands-

(i) lsblk -f

```
[root@rhel9-server /]# lsblk -f
NAME        FSTYPE  FSVER           LABEL                   UUID                                   FSAVAIL FSUSE% MOUNTPOINTS
sda
├─sda1      xfs                                             b1d8efeb-9a48-45d3-b762-275cd0660acd
└─sda2      xfs                                             49c4d6de-3156-4900-9967-a46f267db90c
sdb
sdc
sdd
sr0         iso9660 Joliet Extension RHEL-9-0-0-BaseOS-x86_64 2022-08-05-06-19-40-00                     0   100% /repodata
nvme0n1
├─nvme0n1p1 xfs                                             1b97b83e-8c41-4cd9-a9e6-894073b7299f    1.8G    11% /boot
├─nvme0n1p2 xfs                                             4bec8248-9eb3-48da-902d-b0b7c7e10d16     48G     4% /
└─nvme0n1p3 swap    1                                       89f90bba-41f6-4381-941e-8b8d7dc8b66e                [SWAP]
```

(ii) blkid /dev/sda2

```
[root@rhel9-server /]#
[root@rhel9-server /]# blkid /dev/sda2
/dev/sda2: UUID="49c4d6de-3156-4900-9967-a46f267db90c" BLOCK_SIZE="512" TYPE="xfs" PARTUUID="a6e09235-02"
[root@rhel9-server /]#
[root@rhel9-server /]#
```

GPT Disk Partition Steps Using gdisk:-

1. Check all blocks available

```
[root@rhel9-server /]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda           8:0    0   10G  0 disk
├─sda1        8:1    0    1G  0 part
├─sda2        8:2    0    2G  0 part
├─sda3        8:3    0    1G  0 part
├─sda4        8:4    0  512B  0 part
├─sda5        8:5    0    1G  0 part
├─sda6        8:6    0    1G  0 part
├─sda7        8:7    0    2G  0 part
└─sda8        8:8    0    1G  0 part
sdb           8:16   0   10G  0 disk
sdc           8:32   0   10G  0 disk
sdd           8:48   0   10G  0 disk
sr0          11:0    1  7.9G  0 rom  /repodata
nvme0n1     259:0    0  100G  0 disk
├─nvme0n1p1 259:1    0    2G  0 part /boot
├─nvme0n1p2 259:2    0   50G  0 part /
└─nvme0n1p3 259:3    0    4G  0 part [SWAP]
[root@rhel9-server /]#
```

2. Creating MBR partition in disk sdb.

```
[root@rhel9-server /]# gdisk /dev/sdb
GPT fdisk (gdisk) version 1.0.7

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries in memory.

Command (? for help):
```

3. Add a new partition, use keyword n. Select no. 1 for new primary partition. Keep first sector as it is & hit enter, use +1G to create 1GB partition & hit enter. Type will be default ‘Linux Filesystems’. We can change it as per our requirement.

```
Command (? for help): n
Partition number (1-128, default 1): 1
First sector (34-20971486, default = 2048) or {+-}size{KMGTP}:
Last sector (2048-20971486, default = 20971486) or {+-}size{KMGTP}: +1G
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'
```

4. Type w and hit enter to save this partition. Now type Y to proceed.

```
Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): Y
OK; writing new GUID partition table (GPT) to /dev/sdb.
The operation has completed successfully.
[root@rhel9-server /]#
```

5. To verify, use lsblk command.

```
[root@rhel9-server /]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda           8:0    0   10G  0 disk
├─sda1        8:1    0    1G  0 part
├─sda2        8:2    0    2G  0 part
├─sda3        8:3    0    1G  0 part
├─sda4        8:4    0  512B  0 part
├─sda5        8:5    0    1G  0 part
├─sda6        8:6    0    1G  0 part
├─sda7        8:7    0    2G  0 part
└─sda8        8:8    0    1G  0 part
sdb           8:16   0   10G  0 disk
└─sdb1        8:17   0    1G  0 part
sdc           8:32   0   10G  0 disk
sdd           8:48   0   10G  0 disk
sr0          11:0    1  7.9G  0 rom  /repodata
nvme0n1     259:0    0  100G  0 disk
├─nvme0n1p1 259:1    0    2G  0 part /boot
├─nvme0n1p2 259:2    0   50G  0 part /
└─nvme0n1p3 259:3    0    4G  0 part [SWAP]
[root@rhel9-server /]#
```

6. To check GPT disk detail, use gdisk -l /dev/sdb.

```
[root@rhel9-server /]# gdisk -l /dev/sdb
GPT fdisk (gdisk) version 1.0.7

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.
Disk /dev/sdb: 20971520 sectors, 10.0 GiB
Model: VMware Virtual S
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): C0DB0D2D-0E29-47CB-9FA4-D00402235A36
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 20971486
Partitions will be aligned on 2048-sector boundaries
Total free space is 18874301 sectors (9.0 GiB)

Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048         2099199   1024.0 MiB  8300  Linux filesystem
[root@rhel9-server /]#
```

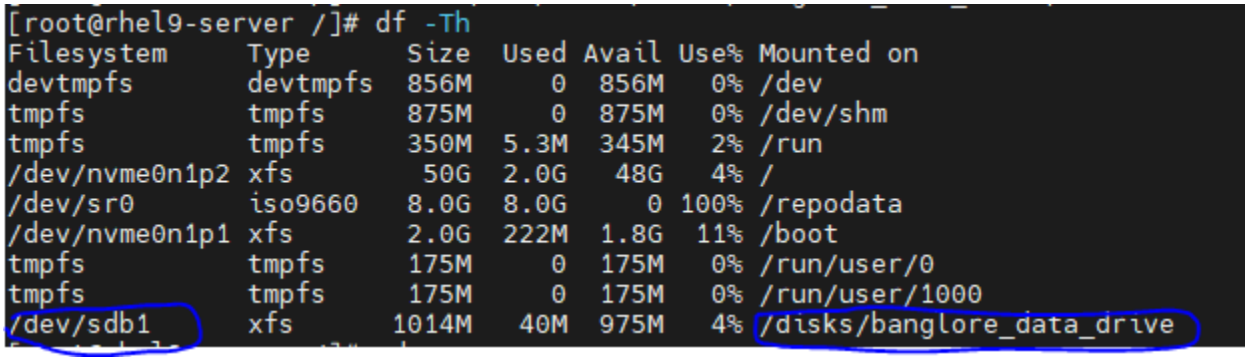
7. Next, we need to format this partition using xfs file system.

```
[root@rhel9-server /]# mkfs.xfs /dev/sdb1
meta-data=/dev/sdb1              isize=512    agcount=4, agsize=65536 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=262144, 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@rhel9-server /]#
```

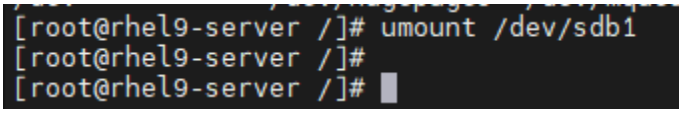
8. Next, mount this partition in /disks/banglore_data_drive directory.

```
[root@rhel9-server /]#
[root@rhel9-server /]# mount /dev/sdb1 /disks/banglore_data_drive/
[root@rhel9-server /]#
```

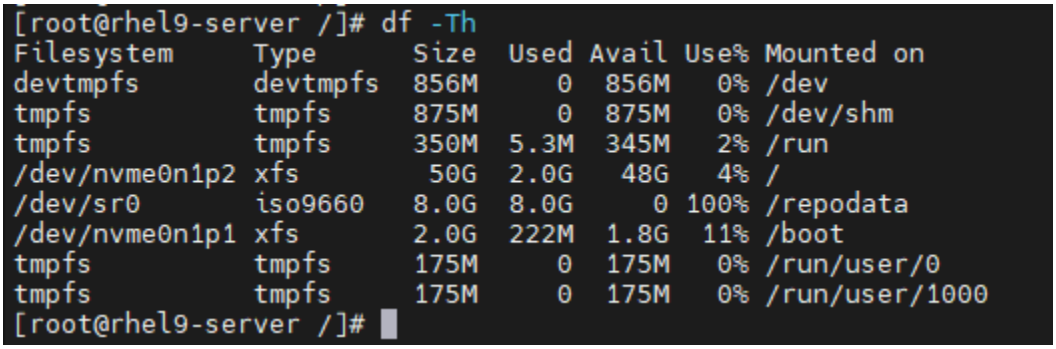
9. Now verify it again using df -Th.



10. To unmount this partition, use umount command shown in screenshot.



11. Now verify it again using df -Th.



12. This mount will be temporary & get remove automatically once server reboot. To make it permanent, we need to add its entry in fstab file in etc directory in two ways. Please refer MBR disk partition steps to do this.