Karthik Pradeep Hegadi

2KE20CS032

Assignment 18

Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.

STORAGE

==========

Assignment 1- : Test Disk Performance OPS With Fio utility for random read write

Operations.

==========

Instructions

Run this command :

(fio—randrepeat=1—joengine=libaio—direct=1--gtodreduce=1--name=test--filename=test)

Or

(fio --randrepeat=1 --ioengine=libaio --direct=1 --gtod_reduce=1 --name=/path/to/testfile --filename=/path/to/testfile --bs=2k --iodepth=64 --size=2G --readwrite=randrw --rwmixread=75

)

This command will create a 2 GB file(-size option), and perform 2KB reads and writes(--bs option) using 75%/25%, with 64 operations(-iodepth option) running at a time. You can change these value and test as well.

To measure random write OPS use option:

--readwrite=randwrite

To measure random read OPS use option:

--readwrite=randread

Check & record the output of this command to know the disk performance in terms of read operations per second (IOPS) and write operations per second.

What is fio?

fio is a tool that will spawn a number of threads or processes doing a particular type of 1/0 action as specified by the user. The typical use of fio is to write a job file matching the 1/0 load

one wants to simulate

Paremeters:

iodepth: Number of 1/0 units to keep in flight against the file.

bs: filesystem block size

randrepeat: Seed the random number generator used for random I/0 patterns in a predictable way so the pattern is repeatable across runs. **Default**: true. Value 1 is false

ioengine: Defines how the job issues 1/O to the file. libaio is Linux native asynchronous 1/0

-direct: direct or buffered

-god_ reduce: tod(time of day). Whether to reduce gettimeofday call during the operation

-filename: name of the file created. You can see in current directory

--bs: Block Size

-odepth: Number of I/0 units to keep in flight against the file.

-size: Size of the file to be created for test

-readwrite: Type of I/0 pattern. "randrw" Random mixed reads and writes.

-rwmixread: Percentage of a mixed workload that should be reads. Default: 50.

```
~ (0.239s)
ssh lusy@10.211.55.6

lusy@localhost.localdomain ~ (3.226s)
sudo yum install fio
[sudo] password for lusy:
Last metadata expiration check: 1:17:57 ago on Sat 19 Aug 2023 10:41:06 AM IST.
Package fio-3.35-1.el9.aarch64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

```
lusy@localhost.localdomain ~ (18.835s)
sudo fio --randrepeat=1 --ioengine=libaio --direct=1 --gtod_reduce=1 --name=/path/to/testfile --filename=/path/to/testfile --bs=2k
-rwmixread=75
/path/to/testfile: (g=0): rw=randrw, bs=(R) 2048B-2048B, (W) 2048B-2048B, (T) 2048B-2048B, ioengine=libaio, iodepth=64
fio-3.35
Starting 1 process
/path/to/testfile: Laying out IO file (1 file / 2048MiB)
Jobs: 1 (f=1): [m(1)][100.0%][r=84.1MiB/s,w=28.0MiB/s][r=43.1k,w=14.3k IOPS][eta 00m:00s]
/path/to/testfile: (groupid=0, jobs=1): err= 0: pid=26680: Sat Aug 19 12:31:50 2023
  read: IOPS=47.2k, BW=92.2MiB/s (96.7MB/s)(1535MiB/16645msec)
   bw (  KiB/s): min=14416, max=157708, per=100.00%, avg=94498.24, stdev=45405.00, samples=33
   iops        : min= 7208, max=78854, avg=47249.09, stdev=22702.51, samples=33
  write: IOPS=15.8k, BW=30.8MiB/s (32.3MB/s)(513MiB/16645msec); 0 zone resets
   bw (  KiB/s): min= 4744, max=52852, per=100.00%, avg=31576.09, stdev=15202.05, samples=33
   iops        : min= 2372, max=26426, avg=15788.00, stdev=7601.05, samples=33
  cpu          : usr=5.49%, sys=33.34%, ctx=74890, majf=0, minf=17
  IO depths    : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
     submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
     complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.1%, >=64=0.0%
     issued rwts: total=785920,262656,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=64

Run status group 0 (all jobs):
   READ: bw=92.2MiB/s (96.7MB/s), 92.2MiB/s-92.2MiB/s (96.7MB/s-96.7MB/s), io=1535MiB (1610MB), run=16645-16645msec
  WRITE: bw=30.8MiB/s (32.3MB/s), 30.8MiB/s-30.8MiB/s (32.3MB/s-32.3MB/s), io=513MiB (538MB), run=16645-16645msec

Disk stats (read/write):
    dm-0: ios=784151/262062, merge=0/0, ticks=596440/266730, in_queue=863170, util=99.53%, aggrios=785109/262600, aggrmerge=963/10
946, aggrutil=99.02%
  sda: ios=785109/262600, merge=963/100, ticks=598438/263453, in_queue=861946, util=99.02%
```

==========

Assignment 2- Setup NFS server in one linux system and access the shared directory from another linux system which will act as NFS client.

Refer Network file system slide for this Assignment

Note- Here to comolete this assignment I have cloned two machines

image 1 : *which is orignale image of centos.*

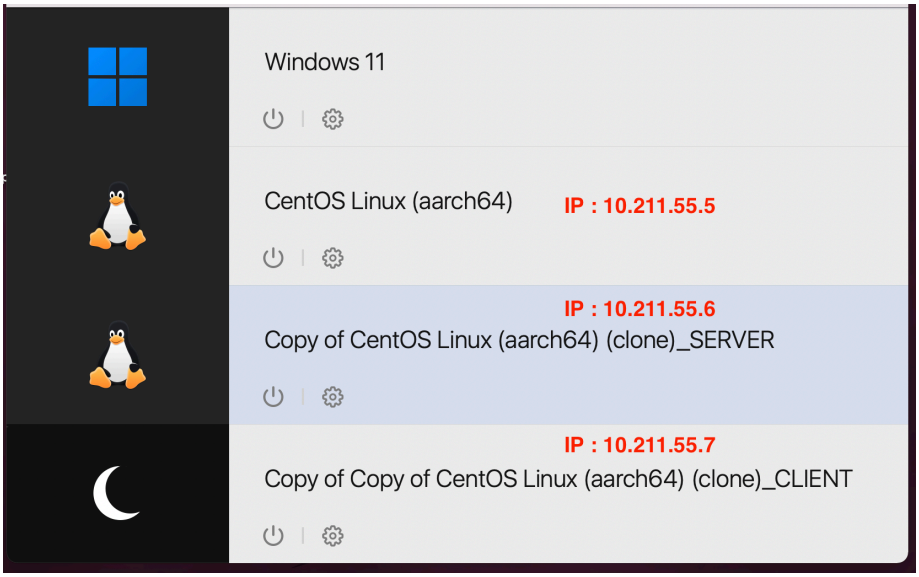
image 2 : *which i took from origmal machine :* ***Clone 1***

image 3 : *which I took form clone 1 machine :* ***Clone 2***

For Clone 1: *Machine I have added new user called* ***SERVER (also added sudo previledge to it).***

For Clone 2: *Machine I have added new user called* ***CLIENT (also added sudo previledge to it).***

==========



SERVER configuration:

```
lusy@localhost.localdomain ~ (18m 44.49s)
su SERVER

Password:
[SERVER@localhost lusy]$ cd
[SERVER@localhost ~]$ sudo yum install nfs-utils
[sudo] password for SERVER:
Last metadata expiration check: 3:32:45 ago on Sat 19 Aug 2023 10:41:06 AM IST.
Package nfs-utils-1:2.5.4-20.el9.aarch64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

```
[SERVER@localhost ~]$ sudo systemctl enable nfs-server
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service → /usr/lib/system
```

```
[SERVER@localhost ~]$ sudo systemctl start nfs-server
[SERVER@localhost ~]$ sudo systemctl status nfs-server
● nfs-server.service - NFS server and services
     Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; preset: d>
    Drop-In: /run/systemd/generator/nfs-server.service.d
             └─order-with-mounts.conf
     Active: active (exited) since Sat 2023-08-19 13:30:51 IST; 43min ago
   Main PID: 18690 (code=exited, status=0/SUCCESS)
        CPU: 13ms

Aug 19 13:30:51 localhost.localdomain systemd[1]: Starting NFS server and services.>
Aug 19 13:30:51 localhost.localdomain systemd[1]: Finished NFS server and services.
```

```
[SERVER@localhost ~]$ sudo systemctl enable rpcbind
[SERVER@localhost ~]$ sudo systemctl start rpcbind
[SERVER@localhost ~]$ sudo systemctl status  rpcbind
● rpcbind.service - RPC Bind
     Loaded: loaded (/usr/lib/systemd/system/rpcbind.service; enabled; preset: enab>
     Active: active (running) since Sat 2023-08-19 13:30:51 IST; 43min ago
TriggeredBy: ● rpcbind.socket
       Docs: man:rpcbind(8)
   Main PID: 18657 (rpcbind)
      Tasks: 1 (limit: 10677)
     Memory: 1.2M
        CPU: 31ms
     CGroup: /system.slice/rpcbind.service
             └─18657 /usr/bin/rpcbind -w -f

Aug 19 13:30:51 localhost.localdomain systemd[1]: Starting RPC Bind...
Aug 19 13:30:51 localhost.localdomain systemd[1]: Started RPC Bind.
```

```
[SERVER@localhost ~]$ cd
[SERVER@localhost ~]$ ls
SERVER
[SERVER@localhost ~]$ cd SERVER
[SERVER@localhost SERVER]$ ls
files
[SERVER@localhost SERVER]$ cd files
[SERVER@localhost files]$ pwd
/home/SERVER/SERVER/files
```

```
exports: Failed to stat /SERVER/files: No such file or directory
[SERVER@localhost files]$ sudo vi /etc/exports
```

```
vi /etc/exports                                    ~

/home/SERVER/SERVER/files    10.211.55.7(rw,sync,no_root_squash)
~
~
~
~
```

```
[SERVER@localhost files]$ sudo systemctl stop  firewalld.service
[SERVER@localhost files]$ sudo systemctl status  firewalld.service
○ firewalld.service - firewalld - dynamic firewall daemon
     Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: en>
     Active: inactive (dead) since Sat 2023-08-19 14:18:34 IST; 11s ago
   Duration: 1h 3min 53.949s
       Docs: man:firewalld(1)
    Process: 691 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (co>
   Main PID: 691 (code=exited, status=0/SUCCESS)
        CPU: 279ms

Aug 19 13:14:40 localhost systemd[1]: Starting firewalld - dynamic firewall daemon.>
Aug 19 13:14:40 localhost systemd[1]: Started firewalld - dynamic firewall daemon.
Aug 19 14:18:34 localhost.localdomain systemd[1]: Stopping firewalld - dynamic fire>
Aug 19 14:18:34 localhost.localdomain systemd[1]: firewalld.service: Deactivated su>
Aug 19 14:18:34 localhost.localdomain systemd[1]: Stopped firewalld - dynamic firew>
```

```
[SERVER@localhost files]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gro
    link/ether 00:1c:42:cd:4f:6f brd ff:ff:ff:ff:ff:ff
    inet 10.211.55.6/24 brd 10.211.55.255 scope global dynamic noprefixroute enp0
       valid_lft 1462sec preferred_lft 1462sec
    inet6 fdb2:2c26:f4e4:0:21c:42ff:fecd:4f6f/64 scope global dynamic noprefixrou
       valid_lft 2591911sec preferred_lft 604711sec
    inet6 fe80::21c:42ff:fecd:4f6f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

CLIENT configuration:

```
lusy@localhost.localdomain ~ (13m 1.23s)
su CLIENT

Password:
[CLIENT@localhost lusy]$ cd
[CLIENT@localhost ~]$ sudo yum install nfs-utils
[sudo] password for CLIENT:
Last metadata expiration check: 3:38:30 ago on Sat 19 Aug 2023 10:41:06 AM IST.
Package nfs-utils-1:2.5.4-20.el9.aarch64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

```
Complete!
[CLIENT@localhost ~]$ ls
mounted
[CLIENT@localhost ~]$ cd mounted
[CLIENT@localhost mounted]$ ls
mountedfiles
[CLIENT@localhost mounted]$ cd mountedfiles
[CLIENT@localhost mountedfiles]$ pwd
/home/CLIENT/mounted/mountedfiles
```

```
[CLIENT@localhost ~]$ sudo mount 10.211.55.6:/home/SERVER/SERVER/files /home/CLIENT/mounted/mountedfiles
```

```
[CLIENT@localhost ~]$ ls
mounted
[CLIENT@localhost ~]$ cd mounted
[CLIENT@localhost mounted]$ ls
mountedfiles
[CLIENT@localhost mounted]$ cd mountedfiles
[CLIENT@localhost mountedfiles]$ ls
1
[CLIENT@localhost mountedfiles]$ touch files1.txt
```

```
[CLIENT@localhost mountedfiles]$ vi files1.txt
[CLIENT@localhost mountedfiles]$ ls
files1.txt
[CLIENT@localhost mountedfiles]$ cat files1.txt
WELCOME TO NFS
[CLIENT@localhost mountedfiles]$
```

Verification:

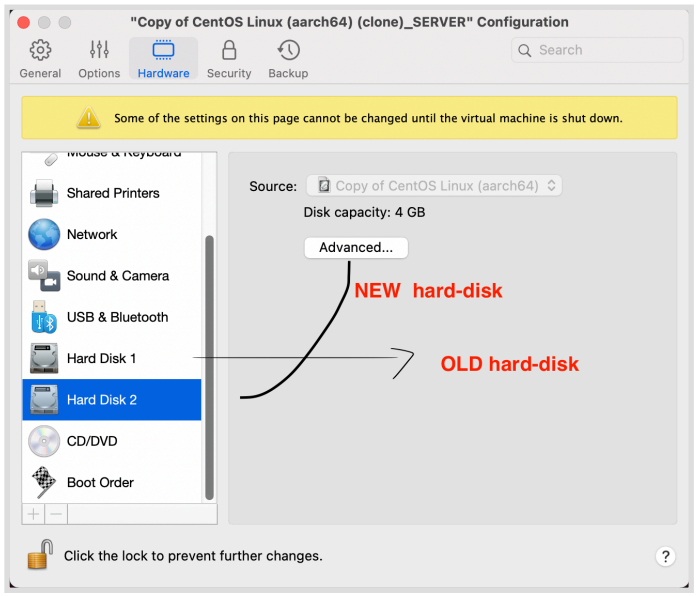
```
[SERVER@localhost SERVER]$ cd files
[SERVER@localhost files]$ ls
1  files1.txt
[SERVER@localhost files]$ cat files1.txt
WELCOME TO NFS
[SERVER@localhost files]$
```

==========

Assignment 3 - Add new Virtual hard disk Drive for the virtual box and partition the drive in linux then create file system and mount the file system.

Instructions

1. Add hew hard-disk I am using parallel desktop. I followed these instructions (https://kb.parallels.com/117649)



==========

1.Run "**Isblk**" command to check the partitions

```
lusy@localhost.localdomain ~ (0.03s)
lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda           8:0    0   64G  0 disk
├─sda1        8:1    0  600M  0 part /boot/efi
├─sda2        8:2    0    1G  0 part /boot
└─sda3        8:3    0 62.4G  0 part
  ├─cs-root 253:0    0 40.6G  0 lvm  /
  ├─cs-swap 253:1    0    2G  0 lvm  [SWAP]
  └─cs-home 253:2    0 19.8G  0 lvm  /home
sdb           8:16   0    4G  0 disk
sr0          11:0    1 1024M  0 rom
```

2.create a new partition using the command “fdisk/dev/sdb”

```
lusy@localhost.localdomain ~ (2m 19.95s)
sudo fdisk /dev/sdb
[sudo] password for lusy:

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.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x2b848c38.

Command (m for help):


Command (m for help): m
```

```
Help:

  DOS (MBR)
   a   toggle a bootable flag
   b   edit nested BSD disklabel
   c   toggle the dos compatibility flag

  Generic
   d   delete a partition
   F   list free unpartitioned space
   l   list known partition types
   n   add a new partition
   p   print the partition table
   t   change a partition type
   v   verify the partition table
   i   print information about a partition

  Misc
   m   print this menu
   u   change display/entry units
   x   extra functionality (experts only)

  Script
   I   load disk layout from sfdisk script file
   O   dump disk layout to sfdisk script file

  Save & Exit
   w   write table to disk and exit
   q   quit without saving changes

  Create a new label
   g   create a new empty GPT partition table
   G   create a new empty SGI (IRIX) partition table
   o   create a new empty DOS partition table
   s   create a new empty Sun partition table
```

3. **p** option will show the list of partitions

```
Command (m for help): p

Disk /dev/sdb: 4 GiB, 4294967296 bytes, 8388608 sectors
Disk model: Copy of CentOS L
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0x2b848c38
```

4. **n** option will create new partition

5.Create 1 primary partition

6. specify where the partition will begin and end by accepting the default values.

```
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): 1
First sector (2048-8388607, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-8388607, default 8388607):

Created a new partition 1 of type 'Linux' and of size 4 GiB.

Command (m for help): w
```

7.**w** option will save the changes

(Note: Since this is the first partition we can start at the first available sector and since we want to use the entire disk we can specify the last sector as the end. Note that, if you wish to create multiple partitions - you can even specify the size of each partition by sectors, bytes, kilobytes or megabytes)

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

Verification:

```
lusy@localhost.localdomain ~ (0.03s)
lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda           8:0    0   64G  0 disk
├─sda1        8:1    0  600M  0 part /boot/efi
├─sda2        8:2    0    1G  0 part /boot
└─sda3        8:3    0 62.4G  0 part
  ├─cs-root 253:0    0 40.6G  0 lvm  /
  ├─cs-swap 253:1    0    2G  0 lvm  [SWAP]
  └─cs-home 253:2    0 19.8G  0 lvm  /home
sdb           8:16   0    4G  0 disk
└─sdb1        8:17   0    4G  0 part
sr0          11:0    1 1024M  0 rom
```

8.Create the file system on the partition using the command “/usr/sbin/mkfs.ext4-Ldata /dev/sdb1”

```
lusy@localhost.localdomain ~ (0.104s)
sudo /usr/sbin/mkfs.ext4 -L data /dev/sdb1
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 1048320 4k blocks and 262144 inodes
Filesystem UUID: 61ae6cb2-8cde-49c1-a176-115218a47ef9
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

9. Create the directory /data

```
lusy@localhost.localdomain ~ (0.06s)
sudo mkdir /data
```

10. Mount the file system using the command "mount LABEL=data /data”

```
lusy@localhost.localdomain ~ (0.11s)
sudo mount LABEL=data /data
```

11.Run the mount command with no arguments which shows all currently mounted filesvstems. You can use the below command as well.

```
lusy@localhost.localdomain ~ (0.027s)
mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4096k,nr_inodes=213567,mod
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relati
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=349380k,nr_inodes=819200,m
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,m
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)
efivarfs on /sys/firmware/efi/efivars type efivarfs (rw,nosuid,nodev,noexec,relat
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
/dev/mapper/cs-root on / type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,lo
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=29,pgrp=1,timeou
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabe
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,secla
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
none on /run/credentials/systemd-sysctl.service type ramfs (ro,nosuid,nodev,noexec
none on /run/credentials/systemd-tmpfiles-setup-dev.service type ramfs (ro,nosuid,
nfsd on /proc/fs/nfsd type nfsd (rw,relatime)
/dev/mapper/cs-home on /home type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=
/dev/sda2 on /boot type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize
/dev/sda1 on /boot/efi type vfat (rw,relatime,fmask=0077,dmask=0077,codepage=437,i
none on /run/credentials/systemd-tmpfiles-setup.service type ramfs (ro,nosuid,node
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=174688k
/dev/sdb1 on /data type ext4 (rw,relatime,seclabel)
```

12. Run "**lsblk -o NAME, FSTYPE, SIZE, MOUNTPOINT,LABEL** "

```
lusy@localhost.localdomain ~ (0.029s)
lsblk -o NAME,FSTYPE,SIZE,MOUNTPOINT,LABEL
NAME        FSTYPE       SIZE MOUNTPOINT LABEL
sda                       64G
├─sda1      vfat         600M /boot/efi
├─sda2      xfs            1G /boot
└─sda3      LVM2_member 62.4G
  ├─cs-root xfs         40.6G /
  ├─cs-swap swap           2G [SWAP]
  └─cs-home xfs         19.8G /home
sdb                       4G
└─sdb1      ext4             4G /data      data
sr0                    1024M
```

```
lusy@localhost.localdomain ~ (0.023s)
mount | grep '^/dev'
/dev/mapper/cs-root on / type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,
/dev/mapper/cs-home on /home type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=
/dev/sda2 on /boot type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquot
/dev/sda1 on /boot/efi type vfat (rw,relatime,fmask=0077,dmask=0077,codepage=437,iocharset=as
/dev/sdb1 on /data type ext4 (rw,relatime,seclabel)
```

11. To automatically mount the file system at the time of boot we need the below entry to be added to the /etc/stab file

LABEL=data /data ext4 defaults 1 2

```
lusy@localhost.localdomain ~ (54.618s)
sudo vi /etc/fstab
```

```
#
# /etc/fstab
# Created by anaconda on Fri Aug  4 05:32:46 2023
#
# 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.
#
/dev/mapper/cs-root     /                       xfs     defaults        0 0
UUID=6a810383-8037-4807-b8bb-9a7349dbdf86 /boot                   xfs     defaults        0 0
UUID=D776-2717          /boot/efi               vfat    umask=0077,shortname=winnt 0 2
/dev/mapper/cs-home     /home                   xfs     defaults        0 0
/dev/mapper/cs-swap     none                    swap    defaults        0 0
LABEL=data    /data    ext4    defaults    1 2
```

```
lusy@localhost.localdomain ~ (0.026s)
lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda           8:0    0   64G  0 disk
├─sda1        8:1    0  600M  0 part /boot/efi
├─sda2        8:2    0    1G  0 part /boot
└─sda3        8:3    0 62.4G  0 part
  ├─cs-root 253:0    0 40.6G  0 lvm  /
  ├─cs-swap 253:1    0    2G  0 lvm  [SWAP]
  └─cs-home 253:2    0 19.8G  0 lvm  /home
sdb           8:16   0    4G  0 disk
└─sdb1        8:17   0    4G  0 part /data
sr0          11:0    1 1024M  0 rom
```

==========

Assignment 4 - Expanding a filesystem on the fly using Logical Volume Management.

Instructions

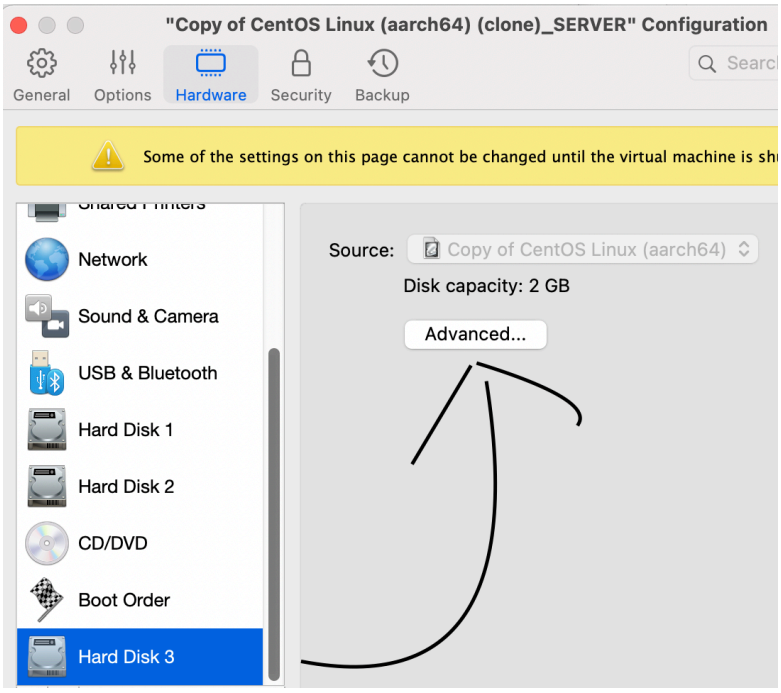
Note: install LVM using sudo yum install lvm2

Adding a new logical volume

```
lusy@localhost.localdomain ~ (5.316s)
sudo yum install lvm2
[sudo] password for lusy:
Last metadata expiration check: 0:11:25 ago on Sat 19 Aug 2023 11:27:08 PM IST.
Package lvm2-9:2.03.21-3.el9.aarch64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

==========

1.1 Install a new hard drive.



1.2 Use **Isblk** command to identify the new hard drive.

```
~ (0.437s)
ssh lusy@10.211.55.6

lusy@localhost.localdomain ~ (0.034s)
lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda           8:0    0   64G  0 disk
├─sda1        8:1    0  600M  0 part /boot/efi
├─sda2        8:2    0    1G  0 part /boot
└─sda3        8:3    0 62.4G  0 part
  ├─cs-root 253:0    0 40.6G  0 lvm  /
  ├─cs-swap 253:1    0    2G  0 lvm  [SWAP]
  └─cs-home 253:2    0 19.8G  0 lvm  /home
sdb           8:16   0    4G  0 disk
└─sdb1        8:17   0    4G  0 part /data
sdc           8:32   0    2G  0 disk
sr0          11:0    1 1024M  0 rom
```

1.3 Create Physical Volume from the hard drive using "**pvcreate /dev/sdc**". Assuming **/dev/sde** is the new hard drive created.

```
lusy@localhost.localdomain ~ (0.18s)
sudo pvcreate /dev/sdc
  Physical volume "/dev/sdc" successfully created.
```

1.4 You can list all the PV with the command: **pvscan**.

```
lusy@localhost.localdomain ~ (0.142s)
sudo pvscan
  PV /dev/sda3   VG cs              lvm2 [62.41 GiB / 0    free]
  PV /dev/sdc                       lvm2 [2.00 GiB]
  Total: 2 [64.41 GiB] / in use: 1 [62.41 GiB] / in no VG: 1 [2.00 GiB]
```

1.5 If you want to display more information about any specific PV, let's say **/dev/sc** run "**pvdisplay/ dev/sdc**"

```
lusy@localhost.localdomain ~ (0.144s)
sudo pvdisplay /dev/sdc
  --- Physical volume ---
  PV Name               /dev/sdc
  VG Name               share
  PV Size               2.00 GiB / not usable 4.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              511
  Free PE               230
  Allocated PE          281
  PV UUID               nGfLYr-7dAY-i27U-BWZ3-cHJO-eF6e-6PBOwr
```

1.6 Create Volume Group named "**share**" using **sudo vgcreate share / dev/sdc**.

```
lusy@localhost.localdomain ~ (0.182s)
sudo vgcreate share /dev/sdc
  Volume group "share" successfully created
```

1.7 you can list all the VGs with the command: **vgscan**

```
lusy@localhost.localdomain ~ (0.166s)
sudo vgscan
  Found volume group "share" using metadata type lvm2
  Found volume group "cs" using metadata type lvm2
```

1.8 You can display more information about any specific VG, such as share with the command: **vgdisplay share**.

```
lusy@localhost.localdomain ~ (0.178s)
sudo vgdisplay share
  --- Volume group ---
  VG Name               share
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               <2.00 GiB
  PE Size               4.00 MiB
  Total PE              511
  Alloc PE / Size       0 / 0
  Free  PE / Size       511 / <2.00 GiB
  VG UUID               s7ss9d-mi9g-JPXs-XX14-t2oH-qbOX-p7uGmQ
```

1.9 Create Logical Volume: To create a 100MB LV mylv from VG share use the command "**sudo lvcreate --size 100M --name mylv share**"

```
lusy@localhost.localdomain ~ (0.213s)
sudo lvcreate --size 100M --name mylv share
  Logical volume "mylv" created.
```

1.10 create another LV mylv1 of size 1GB from VG share with the command:" **sudo Ivcreate -size 1G --name mylv1 share**"

```
lusy@localhost.localdomain ~ (0.154s)
sudo lvcreate --size 1G --name mylv1 share
  Logical volume "mylv1" created.
```

1.11 list all the LVs with the commands: **Ivscan or Ivs.**

```
lusy@localhost.localdomain ~ (0.165s)
sudo lvscan
  ACTIVE            '/dev/share/mylv' [100.00 MiB] inherit
  ACTIVE            '/dev/share/mylv1' [1.00 GiB] inherit
  ACTIVE            '/dev/cs/swap' [2.04 GiB] inherit
  ACTIVE            '/dev/cs/home' [19.80 GiB] inherit
  ACTIVE            '/dev/cs/root' [40.56 GiB] inherit
```

1.12 You can also display more information about any specific LV with the command: "**sudo IvdisplayVG_NAME/LV_NAME**"/.

1.13 The LVs are available as /dev/VG NAME/LV NAME. You should be able to see teh 2 Vs created so far.

```
lusy@localhost.localdomain ~ (0.186s)
sudo lvdisplay  /dev/share
  --- Logical volume ---
  LV Path                /dev/share/mylv
  LV Name                mylv
  VG Name                share
  LV UUID                YIgZ5G-JKuR-eEQq-noXi-YEGX-cRkj-B7ZMEd
  LV Write Access        read/write
  LV Creation host, time localhost.localdomain, 2023-08-19 23:45:50 +0530
  LV Status              available
  # open                 1
  LV Size                100.00 MiB
  Current LE             25
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:3
```

```
  --- Logical volume ---
  LV Path                /dev/share/mylv1
  LV Name                mylv1
  VG Name                share
  LV UUID                VRz6ED-9GQx-tW0g-bsbx-Pj6L-8MiR-WG830n
  LV Write Access        read/write
  LV Creation host, time localhost.localdomain, 2023-08-19 23:46:01 +0530
  LV Status              available
  # open                 1
  LV Size                1.00 GiB
  Current LE             256
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:4
```

1.14 Format the Logical Volumes

mkfs-text4 /dev/share/mylv

```
lusy@localhost.localdomain ~ (0.079s)
sudo mkfs.ext4 /dev/share/mylv
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 102400 1k blocks and 25584 inodes
Filesystem UUID: 94424546-8000-46ef-8479-bc341640d663
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

mkfs-text4 /dev/share/mylv1

```
lusy@localhost.localdomain ~ (0.078s)
sudo mkfs.ext4 /dev/share/mylv1
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: aaf93c14-e998-4890-a994-0449712576e9
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376

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

1.15 create mount points where you want to mount these LVs

1.16 add an appropriate entry to the **/etc/fstab** file, and mount the filesystem.

sudo mkdir -pv /lvmdata

```
lusy@localhost.localdomain ~ (0.045s)
sudo mkdir -pv /lvmdata/mylv
mkdir: created directory '/lvmdata'
mkdir: created directory '/lvmdata/mylv'
```

```
lusy@localhost.localdomain ~ (0.059s)
sudo mkdir -pv /lvmdata/mylv1
mkdir: created directory '/lvmdata/mylv1'
```

```
lusy@localhost.localdomain ~ (51.98s)
sudo vi /etc/fstab
```

sudo vi /etc/fstab

```
#
# /etc/fstab
# Created by anaconda on Fri Aug  4 05:32:46 2023
#
# 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.
#
/dev/mapper/cs-root     /                       xfs     defaults        0 0
UUID=6a810383-8037-4807-b8bb-9a7349dbdf86 /boot                   xfs     defaults
UUID=D776-2717          /boot/efi               vfat    umask=0077,shortname=winnt 0 2
/dev/mapper/cs-home     /home                   xfs     defaults        0 0
/dev/mapper/cs-swap     none                    swap    defaults        0 0
LABEL=data    /data    ext4    defaults    1 2

/dev/share/mylv   /lvmdata/mylv   ext4   defaults   0   0
/dev/share/mylv1  /lvmdata/mylv1  ext4   defaults   0   0
```

New mount points

```
lusy@localhost.localdomain ~ (0.065s)
sudo mount -a

mount: (hint) your fstab has been modified, but systemd still uses
       the old version; use 'systemctl daemon-reload' to reload.
```

```
lusy@localhost.localdomain ~ (7.6s)
systemctl daemon-reload

==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ====
Authentication is required to reload the systemd state.
Authenticating as: root
Password:
==== AUTHENTICATION COMPLETE ====
```

```
lusy@localhost.localdomain ~ (0.03s)
lsblk

NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda             8:0    0   64G  0 disk
├─sda1          8:1    0  600M  0 part /boot/efi
├─sda2          8:2    0    1G  0 part /boot
└─sda3          8:3    0 62.4G  0 part
  ├─cs-root   253:0    0 40.6G  0 lvm  /
  ├─cs-swap   253:1    0    2G  0 lvm  [SWAP]
  └─cs-home   253:2    0 19.8G  0 lvm  /home
sdb             8:16   0    4G  0 disk
└─sdb1          8:17   0    4G  0 part /data
sdc             8:32   0    2G  0 disk
├─share-mylv  253:3    0  100M  0 lvm  /lvmdata/mylv
└─share-mylv1 253:4    0    1G  0 lvm  /lvmdata/mylv1
sr0            11:0    1 1024M  0 rom
```