Assignment No Elec 3-C4

1 PROBLEM STATEMENT

Execute at least three commands related to Storage organization of the cloud. Create using Python

2 OBJECTIVE

1. To study the different storage types

2. To study virtual pool.

3. To study different mounting methods.

3 THEORY

3.1 NAS

A Network Attached Storage (NAS) device is a storage device connected to a network that allows storage and retrieval of data from a centralised location for authorised network users and heterogeneous clients. NAS devices are flexible and scale-out, meaning that as you need additional storage, you can add on to what you have.

A NAS is like having a private cloud in the office. It's faster, less expensive and provides all the benefits of a public cloud onsite, giving you complete control.

NAS devices are perfect for small businesses because they are:

i. Simple to operate, a dedicated IT professional is often not required
ii. Lower cost
iii. Easy to use for back up of data, so it's always accessible when you need it
iv. Good at centralising data storage in a safe, reliable way

With a NAS device, data is continually accessible, making it easy for employees to collaborate, respond to customers in a timely fashion, and promptly follow up on sales or other issues because information is in one place. Because a NAS device is like a private cloud, data may be accessed remotely using a network connection, meaning employees can work anywhere and any time.

With more scattered storage arrangements, small businesses may suffer from:

i. Data being out of sync
ii. Reliability and accessibility issues if storage goes down (NAS devices can easily be used for back up, helping ensure your data is constantly at your fingertips when you need it)
iii. Delays in responding to customer service requests or sales queries

3.1.1 TYPICAL IMPLEMENTATION

COMMANDS AND STEPS:

i. fdisk -l
to display disks in you system like sda, sdb, vda etc
ii. fdisk /dev/sdb
use name of disk from which yu would like to use its space for Storage purpose.

Here we are using /dev/sdb

Follow the following process

->n(new)->t(type)->type (8e for Linux LVM) -> p(print)-> w(write) or q(quit without saving)

iii. partprobe
use to tell system that we have partition our storage system without restarting the machine.

iv. df -h
to check for device list

v. pvcreate /dev/sdb1
Select physical storage Device for LVM. Create physical volume.

vi. pvs / pvdisplay
display information about physically liked connected volumes

vii. vgcreate vg_name /dev/sdb1
create virtual storage Device. vg_name is any name to the storage.
/dev/sdb1 which we have created partition drom /dev/sdb

viii. vgs / vgdisplay
display information about virtually connected volume.

ix. lvcreate -L +1000M -T vg_name/mypool

used to create logical volume. +1000M is the storage space. It can be anything you want to give like +10G.
-T is for creating logical volume as Thin Pool, which is used for over-commitment of storage space.
Mypool is any name given to the storage pool

x. lvcreate -V +10G -T vg_name/mypool -n mybrick1
used to create virtual storage logical volume.
We create it from the storage pool mypool. Storage space is +10G as over-commitment.
mybrick1 is any name given to the logical volume.

xi. lvs / lvdisplay
display information about logical volume from storage pool.

xii. mkfs.xfs -i size=512 /dev/vg_name/mybrick1
format the created logical volume.

xiii. mkdir -p /brick/brick1

xiv. echo '/dev/vg_name/mybrick1 /brick/brick1 xfs rw,noatime,inode64,nouuid 1 2' >> /etc/fstab
to make the permanent entry in the fstab so that when machine restart storage will still be mounted.

xv. mount -a

xvi. df -h

4 MATHEMATICAL MODEL

Let S be the solution perspective of the Storage problem such that

S={s, e, i, o, f, DD, NDD, success, failure}

s = initial state
{initial storage devices space i.e. fdisk -l}

e = be the end state
{new volume created and mounted successfully i.e. lvs, df -h }

i = input of the system.
{Input is the name of storage from which we create new volume.

o = output of the system.
Output is the newly created volume with over-commitment.

DD = deterministic data
Name and storage device capacity. NDD =

Non deterministic data

Success case

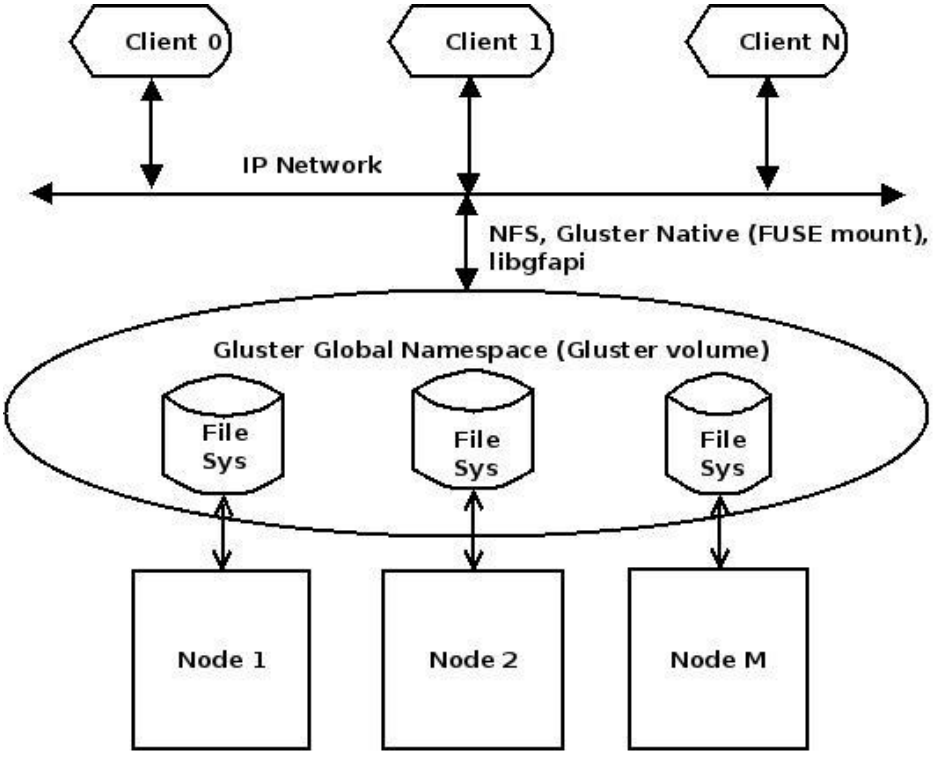
desired outcome generated. Successfully created storage system. All functions working properly.

Failure case = Desired outcome not generated or forced exit due to system error.

5 TEST CASES

case no.	Input	Expected o/p	Actual o/p
1	pvcreate /dev/sdb1	Successfully created physical volume created	Same as Expected
2	lvcreate -L +1000M -T vg_name/mypool	Created thin pool with size same as physical storage device.	Volume group "vg_name" has insufficient free space
3	lvcreate -V +10G -T vg_name/mypool -n mybrick1	Logical volume "mybrick1" created.	Logical volume "mybrick1" created.

5 DIAGRAM



6 IMPLEMENTATION DETAILS

[root@localhost ~]# fdisk /dev/sdb
Welcome to fdisk (util-linux 2.23.2).

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
Building a new DOS disklabel with disk identifier 0xa06ab671.

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
g create a new empty GPT partition table
G create an IRIX (SGI) partition table 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): n
Partition type:
p primary (0 primary, 0 extended, 4 free)
e extended
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-41943039, default 2048): Using
default value 2048
Last sector, +sectors or +size{K,M,G} (2048-41943039, default 41943039): +1000M Partition
1 of type Linux and of size 1000 MiB is set

Command (m for help): p

Disk /dev/sdb: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes I/O size
(minimum/optimal): 512 bytes / 512 bytes Disk label type:
dos
Disk identifier: 0xa06ab671

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		2048	2050047	1024000	83	Linux

Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): 8e
Changed type of partition 'Linux' to 'Linux LVM'

Command (m for help): p

Disk /dev/sdb: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes I/O size
(minimum/optimal): 512 bytes / 512 bytes Disk label type:
dos
Disk identifier: 0xa06ab671

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		2048	2050047	1024000	8e	Linux LVM

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

Calling ioctl() to re-read partition table. Syncing disks.
[root@localhost ~]# partprobe

[root@localhost ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/rhel-root   50G  3.1G   47G   7% /
devtmpfs               1.9G     0  1.9G   0% /dev
tmpfs                  1.9G  148K  1.9G   1% /dev/shm
tmpfs                  1.9G  8.9M  1.9G   1% /run
tmpfs                  1.9G     0  1.9G   0% /sys/fs/cgroup
  /dev/sda1             497M  124M  373M  25% /boot
  /dev/mapper/rhel-home   46G   67M   46G   1% /home

  [root@localhost ~]# pvcreate /dev/sdb1
    Physical volume "/dev/sdb1" successfully created

  [root@localhost ~]# pvs
    PV         VG   Fmt  Attr PSize    PFree
    /dev/sda2  rhel lvm2 a--    99.51g   64.00m
    /dev/sdb1       lvm2 ---  1000.00m 1000.00m

  [root@localhost ~]#  vgcreate vg_name /dev/sdb1
    Volume group "vg_name" successfully created

  [root@localhost ~]# vgs
    VG      #PV #LV #SN Attr   VSize   VFree
    rhel      1   3   0 wz--n-  99.51g  64.00m
    vg_name   1   0   0 wz--n- 996.00m 996.00m

  [root@localhost ~]# lvcreate -L +1000M -T vg_name/mypool
    Volume group "vg_name" has insufficient free space (248 extents): 250 required.

  [root@localhost ~]# lvcreate -L +800M -T vg_name/mypool
    Logical volume "mypool" created.

  [root@localhost ~]# lvcreate -V +10G -T vg_name/mypool -n mybrick1
    Logical volume "mybrick1" created.

  [root@localhost ~]# lvs
    LV       VG      Attr       LSize   Pool   Origin Data%  Meta%  Move Log Cpy%Sync Convert
    home     rhel    -wi-ao----  45.57g
    root     rhel    -wi-ao----  50.00g
    swap     rhel    -wi-ao----   3.88g
    lvol0    vg_name -wi-------
    4.00m
    mybrick1 vg_name Vwi-a-tz--  10.00g mypool        0.00
    mypool   vg_name twi-aotz-- 800.00m              0.00   0.98

  [root@localhost ~]#  mkfs.xfs -i size=512 /dev/vg_name/mybrick1
  meta-data=/dev/vg_name/mybrick1  isize=512    agcount=16, agsize=163824 blks
               =                       sectsz=512   attr=2,        projid32          bit=1
               =                       crc=0        finobt=0
  data         =                       bsize=4096                  blocks=2621184   imaxpct=25
               =                       sunit=16     swidth=16 blks
  naming    =version 2                 bsize=4096   ascii-ci=0 ftype=0
  log       =internal log              bsize=4096   blocks=2560, version=2
            =                          sectsz=512   sunit=16 blks, lazy-count=1
  realtime =none                       extsz=4096   blocks=0, rtextents=0

  [root@localhost ~]# mkdir -p /brick/brick1
```

```
[root@localhost ~]# echo '/dev/vg_name/mybrick1   /brick/brick1  xfs  rw,noatime,inode64,nouuid 1 2' >> /etc/fstab

[root@localhost ~]# mount -a

[root@localhost ~]# df -h
Filesystem                      Size  Used Avail       Use% Mounted on
/dev/mapper/rhel-root             50G  3.1G   47G   7% / dev
tmpfs                              1.9G    0     1.9G 0%     /dev
tmpfs                             1.9G  148K 1.9G 1%    /dev/shm
tmpfs                             1.9G  8.9M 1.9G 1%    /run
tmpfs                             1.9G    0      1.9G 0%    /sys/fs/cgroup
/dev/sda1                         497M 124M 373M25%  /boot
/dev/mapper/rhel-home            46G  67M  46G    1%    /home
/dev/mapper/vg_name-mybrick1    10G  33M  10G  1% /brick/brick1
```

8 CONCLUSION

We have successfully implemented Storage organization in linux system. With the help of this assignment we are able to create virtualized storage.