IST Lab 02: LOGICAL VOLUMES AND SNAPSHOTS

Authors: Kevin Auberson and Paul Gillet

Group: L1GrG Note: L2GrH

Date: March 7, 2024

TASK 1: CREATE PHYSICAL VOLUMES

> 1.Reset your external disk. Using parted remove all partitions, or simply write a new partition table.

```
$ sudo parted
GNU Parted 3.4
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) select /dev/sdb
Using /dev/sdb
(parted) print
Model: Generic Flash Disk (scsi)
Disk /dev/sdb: 31.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
 1      1049kB  15.7GB  15.7GB  primary  fat32        lba
 2      15.7GB  31.5GB  15.7GB  primary  ext4

(parted) rm 1
(parted) rm 2
```

> 2.Create four partitions with these characteristics: primary, 25 MB size, type ext4.

```
(parted) mkpart primary ext4 0MB 25MB
(parted) mkpart primary ext4 25MB 50MB
(parted) mkpart primary ext4 50MB 75MB
```

```
(parted) mkpart primary ext4 75MB 100MB
(parted) p
Model: Generic Flash Disk (scsi)
Disk /dev/sdb: 31.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
 1      512B    25.0MB  25.0MB  primary
 2      25.2MB  50.3MB  25.2MB  primary
 3      50.3MB  75.5MB  25.2MB  primary
 4      75.5MB  99.6MB  24.1MB  primary
```

> 3.List the available LVM commands. They belong to the Debian package lvm2 which should already be installed. Use dpkg with the -L option to list the content of the package. The commands are all located in the /sbin directory. Use grep to filter and sort to sort alphabetically.

```
$ dpkg -L lvm2 | grep /sbin | sort
/sbin
/sbin/fsadm
/sbin/lvchange
/sbin/lvconvert
/sbin/lvcreate
/sbin/lvdisplay
/sbin/lvextend
/sbin/lvm
/sbin/lvmconfig
/sbin/lvmdiskscan
/sbin/lvmdump
/sbin/lvmpolld
/sbin/lvmsadc
/sbin/lvmsar
/sbin/lvreduce
/sbin/lvremove
/sbin/lvrename
/sbin/lvresize
/sbin/lvs
/sbin/lvscan
```

```
/sbin/pvchange
/sbin/pvck
/sbin/pvcreate
/sbin/pvdisplay
/sbin/pvmove
/sbin/pvremove
/sbin/pvresize
/sbin/pvs
/sbin/pvscan
/sbin/vgcfgbackup
/sbin/vgcfgrestore
/sbin/vgchange
/sbin/vgck
/sbin/vgconvert
/sbin/vgcreate
/sbin/vgdisplay
/sbin/vgexport
/sbin/vgextend
/sbin/vgimport
/sbin/vgimportclone
/sbin/vgmerge
/sbin/vgmknodes
/sbin/vgreduce
/sbin/vgremove
/sbin/vgrename
/sbin/vgs
/sbin/vgscan
/sbin/vgsplit
```

> 4.List all partitions that could potentially host a Physical Volume by using pvs with the --all option.

```
$ sudo pvs --all
  PV          VG Fmt Attr PSize PFree
  /dev/loop1         ---     0     0
  /dev/loop10        ---     0     0
  /dev/loop11        ---     0     0
  /dev/loop13        ---     0     0
  /dev/loop2         ---     0     0
  /dev/loop3         ---     0     0
```

```
  /dev/loop4         ---     0     0
  /dev/loop5         ---     0     0
  /dev/loop6         ---     0     0
  /dev/loop7         ---     0     0
  /dev/loop8         ---     0     0
  /dev/loop9         ---     0     0
  /dev/sda2          ---     0     0
  /dev/sda3          ---     0     0
  /dev/sdb1          ---     0     0
  /dev/sdb2          ---     0     0
  /dev/sdb3          ---     0     0
  /dev/sdb4          ---     0     0
```

> 5.On the four partitions of your external disk, create four Physical Volumes using pvcreate. Add the -vv option so that it tells you in detail what it is doing. For the first partition copy the output of the command into the report, but copy only the lines about the partition that receives the Physical Volume and ignore the other messages.

```
$ sudo pvcreate /dev/sdb1 -vv

  /dev/sdb1: using cached size 48828 sectors
  /dev/sdb1: using cached size 48828 sectors
  /dev/sdb1: No lvm label detected
  /dev/sdb1: using cached size 48828 sectors
  /dev/sdb1: using cached size 48828 sectors
  /dev/sdb1: No lvm label detected
  Wiping signatures on new PV /dev/sdb1.
  /dev/sdb1: using cached size 48828 sectors
  devices/default_data_alignment not found in config: defaulting to 1
  Device /dev/sdb1: queue/minimum_io_size is 512 bytes.
  Device /dev/sdb1: queue/optimal_io_size is 0 bytes.
  Device /dev/sdb1: alignment_offset is 0 bytes.
  Set up physical volume for "/dev/sdb1" with 48828 available sectors.
  Scanning for labels to wipe from /dev/sdb1
  Zeroing start of device /dev/sdb1.
  Writing physical volume data to disk "/dev/sdb1".
  /dev/sdb1: Writing label to sector 1 with stored offset 32.
  Physical volume "/dev/sdb1" successfully created.
  Unlocking /run/lock/lvm/P_global
```

> 6. Display detailed information about the first Physical Volume using pvdisplay.

```
$ sudo pvdisplay /dev/sdb1
  "/dev/sdb1" is a new physical volume of "23.84 MiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb1
  VG Name
  PV Size               23.84 MiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               Oe2h5r-x90G-fVst-N6oE-boxD-vree-tSROhy
```

TASK 2: CREATE TWO VOLUME GROUPS

> 1.Create a first Volume Group lab-vg1 that contains only the first Physical Volume. Display the Physical Volume again with pvdisplay. What has changed?

```
$ sudo vgcreate lab-vg1 /dev/sdb1
Volume group "lab-vg1" successfully created

$ sudo pvdisplay /dev/sdb1
  --- Physical volume ---
  PV Name               /dev/sdb1
  VG Name               lab-vg1
  PV Size               23.84 MiB / not usable 3.84 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              5
  Free PE               5
  Allocated PE          0
  PV UUID               Oe2h5r-x90G-fVst-N6oE-boxD-vree-tSROhy
```

As we can see, the Volume Group was successfully created and the PV is now in the VG lab-vg1 also the PV is subdivided into 5 Physical Extends (PEs) with the same size of 4MB (default size)

> 2.Create a second Volume Group lab-vg2 that contains Physical Volumes 2 and 3.

```
$ sudo vgcreate lab-vg2 /dev/sdb2 /dev/sdb3
  Volume group "lab-vg2" successfully created
```

> 3.List all Volume Groups with vgs. Then list all Physical Volumes with pvs. What do you see?

```
$ sudo vgs
  VG      #PV #LV #SN Attr   VSize  VFree
  lab-vg1   1   0   0 wz--n- 20.00m 20.00m
  lab-vg2   2   0   0 wz--n- 40.00m 40.00m

$ sudo pvs
  PV         VG      Fmt  Attr PSize  PFree
  /dev/sdb1  lab-vg1 lvm2 a--  20.00m 20.00m
  /dev/sdb2  lab-vg2 lvm2 a--  20.00m 20.00m
  /dev/sdb3  lab-vg2 lvm2 a--  20.00m 20.00m
  /dev/sdb4          lvm2 ---  23.00m 23.00m
```

As we can see we have 2 VG containing 1 PV and 2 PV for the 2nd one. We can then see with the command pvs all our PVs with their VG. Only PV "/dev/sdb4" is not in a VG.

TASK 3: CREATE LOGICAL VOLUMES

> 1.On the Volume Group lab-vg1 create a Logical Volume of size 20 MB with the command lvcreate -L 20M lab-vg1.

```
$ sudo lvcreate -L 20M lab-vg1

 Logical volume "lvol0" created.
```

> 2.Verify hat the new volume appears when you use lvs to list Logical Volumes. Also verify that it appears when you use lsblk to list the block devices. What is the name of the special file in /dev that represents the volume?

```
$ sudo lvs

  LV    VG      Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync
Convert
```

```
  lvol0 lab-vg1 -wi-a----- 20.00m

$ lsblk
NAME          MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0           7:0    0  63.9M  1 loop /snap/core20/2182
loop1           7:1    0     4K  1 loop /snap/bare/5
loop2           7:2    0  63.4M  1 loop /snap/core20/1974
loop3           7:3    0 266.6M  1 loop /snap/firefox/3836
loop4           7:4    0  73.9M  1 loop /snap/core22/858
loop5           7:5    0 237.2M  1 loop /snap/firefox/2987
loop6           7:6    0 349.7M  1 loop /snap/gnome-3-38-2004/143
loop7           7:7    0 485.5M  1 loop /snap/gnome-42-2204/120
loop8           7:8    0  12.3M  1 loop /snap/snap-store/959
loop9           7:9    0  53.3M  1 loop /snap/snapd/19457
loop10          7:10   0  40.4M  1 loop /snap/snapd/20671
loop11          7:11   0   497M  1 loop /snap/gnome-42-2204/141
loop12          7:12   0  91.7M  1 loop /snap/gtk-common-themes/1535
loop13          7:13   0   452K  1 loop /snap/snapd-desktop-integration/83
sda             8:0    0    25G  0 disk
├─sda1          8:1    0     1M  0 part
├─sda2          8:2    0   513M  0 part /boot/efi
└─sda3          8:3    0  24.5G  0 part /var/snap/firefox/common/host-hunspell
                                        /
sdb             8:16   1  29.3G  0 disk
├─sdb1          8:17   1  23.8M  0 part
│ └─lab--vg1-lvol0
│             252:0    0    20M  0 lvm
├─sdb2          8:18   1    24M  0 part
├─sdb3          8:19   1    24M  0 part
└─sdb4          8:20   1    23M  0 part
sr0            11:0    1  1024M  0 rom
```

The special file is stored in /dev/lab-vg1/lvol0. The directory lab-vg1 has been created when we created the logical volume in this group.

> 3. Create an ext4 file system on the volume. Mount the volume. Fill the file system with a 14 MB file using dd (Google it).

```
$ sudo mkfs.ext4 /dev/lab-vg1/lvol0

mke2fs 1.46.5 (30-Dec-2021)

Creating filesystem with 5120 4k blocks and 5120 inodes

Allocating group tables: done

Writing inode tables: done

Creating journal (1024 blocks): done

Writing superblocks and filesystem accounting information: done

$ sudo dd if=/dev/lab-vg1/lvol0 of=test bs=1M count=14
14+0 records in
14+0 records out
14680064 bytes (15 MB, 14 MiB) copied, 8.58476 s, 1.7 MB/s
```

4. On the Volume Group lab-vg2 create another Logical Volume of size 20 MB, create an ext4 file system on it and mount it. Create a file named foo that contains the text 111.

```
$ sudo lvcreate -L 20M lab-vg2

Logical volume "lvol0" created.

$ sudo mkdir /mnt/vg2-lvol0
$ sudo mkfs.ext4 /dev/lab-vg2/lvol0
mke2fs 1.46.5 (30-Dec-2021)

Creating filesystem with 5120 4k blocks and 5120 inodes

Allocating group tables: done

Writing inode tables: done

Creating journal (1024 blocks): done

Writing superblocks and filesystem accounting information: done
```

```
$ sudo mount /dev/lab-vg2/lvol0 /mnt/vg2-lvol0/

$ sudo sh -c 'echo "111" > foo'
```

TASK 4: GROW A FILE SYSTEM WHILE IT IS IN USE

> 1.Verify that the file system is indeed full (use df -h).

```
$ df -h
Filesystem                  Size  Used Avail Use% Mounted on
...
/dev/mapper/lab--vg1-lvol0   15M   15M     0 100% /mnt/lvol0
/dev/mapper/lab--vg2-lvol0   15M   32K   14M   1% /mnt/vg2-lvol0
```

> 2.Verify that the Volume Group is full (use vgs).

```
$ sudo vgs
  VG      #PV #LV #SN Attr   VSize  VFree
  lab-vg1   1   1   0 wz--n- 20.00m     0
  lab-vg2   2   1   0 wz--n- 40.00m 20.00m
```

> 3.Extend the Volume group using vgextend and verify with vgs.

```
$ sudo vgextend lab-vg1 /dev/sdb4
Volume group "lab-vg1" successfully extended
$ sudo vgs
  VG      #PV #LV #SN Attr   VSize  VFree
  lab-vg1   2   1   0 wz--n- 40.00m 20.00m
  lab-vg2   2   1   0 wz--n- 40.00m 20.00m
```

> 4.Extend the Logical Volume by an additional 20 MB using lvextend --size <new_size> <volume_group>/<logical_volume>.

4.4 La taille n'est pas étendue ici (de 20 à 20).
Il faut passer 40 pour 20 Mb supplémentaire (soit 40, soit +20). (0/1)

```
$ sudo lvextend --size 20M lab-vg1/lvol0
  New size (5 extents) matches existing size (5 extents).
```

> 5.Grow the file system while it is mounted using resize2fs and verify its new capacity with df -h

```
$ sudo resize2fs /dev/lab-vg1/lvol0

resize2fs 1.46.5 (30-Dec-2021)

The filesystem is already 5120 (4k) blocks long.  Nothing to do!

$ df -h

Filesystem                  Size  Used Avail Use% Mounted on

...

/dev/mapper/lab--vg2-lvol0   15M   32K   14M   1% /mnt/vg2-lvol0

/dev/mapper/lab--vg1-lvol0   15M   15M     0 100% /mnt/lvol0
```

Après la commande, il est indiqué que le système de fichier est déjà à sa taille maximum, donc rien de plus n'est fait. Oui, parce qu'elle n'a pas été étendue (cf au-dessus)!

The command output says that the filesystem is already up to it's maximum size, so there's nothing more to be done.

TASK 5: CREATE A SNAPSHOT

> 1.Create a snapshot volume using the --snapshot option of lvcreate. Use the --name option to give it the name snap. You also need to specify a size for the snapshot with the --size option. Remember that initially a snapshot does not consume any storage blocks as the data in the original volume and the snapshot volume is identical. It is only when the data in the two volumes starts deviating that storage blocks are needed. The size of the snapshot determines how many data blocks can be different

```
$ sudo lvcreate --snapshot --name snap --size 15M /dev/lab-vg2/lvol0
  Rounding up size to full physical extent 16.00 MiB
  Logical volume "snap" created.
```

> 2.Display an overview of all Logical Volumes using lvs. Which column shows the name of the original volume?

```
$ sudo lvs
  LV    VG      Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync
Convert
  lvol0 lab-vg1 -wi-a----- 20.00m
  lvol0 lab-vg2 owi-aos--- 20.00m
  snap  lab-vg2 swi-a-s--- 16.00m      lvol0  0.07
```

The sixth column "Origin" show us the name of the original volume.

> 3.Display the charactersticts of the snapshot volume using lvdisplay.

```
$ sudo lvdisplay

 ...

  --- Logical volume ---
  LV Path                /dev/lab-vg2/snap
  LV Name                snap
  VG Name                lab-vg2
  LV UUID                HPtKNz-6vZz-dedq-Bwya-el44-TpbX-ih3NNn
  LV Write Access        read/write
  LV Creation host, time kevin-VirtualBox, 2024-03-17 19:37:42 +0100
  LV snapshot status     active destination for lvol0
  LV Status              available
  # open                 0
  LV Size                20.00 MiB
  Current LE             5
  COW-table size         16.00 MiB
  COW-table LE           4
  Allocated to snapshot  0.07%
  Snapshot chunk size    4.00 KiB
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           252:4

 ...
```

> 3.1.What line shows the name of the original volume?

The "LV Path" shows the name of the original volume

> 3.2.What line shows the size of the original volume?

The ninth lines "LV Size" shows the size of the original volume.

> 3.3.What line shows the space allocated for the snapshot volume?

> 3.4.What does COW stand for?

COW means "Copy-On-Write", it is used when a snapshot is taken. Instead of copying all data, it copies only the metadata like pointers to data blocks. This means that the actial blocks are shared between the original volume and the snapshot. Whenever a block is modified in the original volume, the original data block is first copied to the snapshot, and then the modification is made. This allows to write only when changes are made to new blocks, the original data are preserved on both the snapshot and original volume.

> 4. Mount the snapshot volume. Using the file foo you created earlier verify that the two volumes behave like independent copies.

```
$ sudo mkdir snap
$ sudo mount /dev/lab-vg2/snap /mnt/snap
$ sudo sh -c 'echo "test" > foo'
$ cat foo
test
$ cat /mnt/vg2/foo
111
```

As we can see changes are not yet written

> 5.Make the data of the original volume change completely by using the dd command to write a new file of 14 MB size. Run df -h to see how it affects the fullness of the original volume and the snapshot. What do you see?

```
$ sudo dd if=/dev/lab-vg2/lvol0 of=newtest bs=1M count=14
14+0 records in
14+0 records out
14680064 bytes (15 MB, 14 MiB) copied, 14.4286 s, 1.0 MB/s

$ df -h
Filesystem                  Size  Used Avail Use% Mounted on
tmpfs                       794M  1.7M  792M   1% /run
```

```
/dev/sda3                    24G   13G   11G  55% /
tmpfs                       3.9G     0  3.9G   0% /dev/shm
tmpfs                       5.0M  4.0K  5.0M   1% /run/lock
/dev/sda2                   512M  6.1M  506M   2% /boot/efi
tmpfs                       794M   92K  794M   1% /run/user/1000
/dev/mapper/lab--vg2-lvol0   15M   15M     0 100% /mnt/vg2
/dev/mapper/lab--vg2-snap    15M   28K   14M   1% /mnt/snap
```

As we can see the original volume is full, however the snap volume is unaffected.

> 5.1.The way that you allocated it, is the snapshot volume able support a change of 14 MB of data?

The snapshot is able to support change of 14MB of data because it only stores the changes made since the snapshot was taken.

> 5.2.What happened? Why?

The snapshot volume remained unchanged beacause it only tracks the changes made to the original volume, rather than storing duplicate data.

> 6.Remove the broken snapshot volume.

```
$ sudo lvremove /dev/lab-vg2/snap
Do you really want to remove and DISCARD active logical volume lab-vg2/snap?
[y/n]: y
  Logical volume "snap" successfully removed
```

> 7.Redo the above, this time allocating sufficient space to the snapshot volume to support a complete change of data of the original volume.

```
$ sudo lvcreate --snapshot --name snap --size 20M /dev/lab-vg2/lvol0
  Logical volume "snap" created.

$ sudo lvs
  LV    VG      Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync
Convert
  lvol0 lab-vg1 -wi-ao---- 20.00m
  lvol0 lab-vg2 owi-aos--- 20.00m
  snap  lab-vg2 swi-a-s--- 20.00m      lvol0  0.06
```

```
$ sudo lvdisplay
...
  --- Logical volume ---
  LV Path                /dev/lab-vg2/snap
  LV Name                snap
  VG Name                lab-vg2
  LV UUID                XiiyxJ-dnH9-0fXz-KOVU-Uu4W-GpO1-Q4FgzD
  LV Write Access        read/write
  LV Creation host, time kevin-VirtualBox, 2024-03-21 09:41:20 +0100
  LV snapshot status     active destination for lvol0
  LV Status              available
  # open                 0
  LV Size                20.00 MiB
  Current LE             5
  COW-table size         20.00 MiB
  COW-table LE           5
  Allocated to snapshot  0.06%
  Snapshot chunk size    4.00 KiB
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           252:4
...

$ sudo mkdir /mnt/snap
$ sudo mount /dev/lab-vg2/snap /mnt/snap
$ sudo sh -c 'echo "test" > foo'
$ cat foo
111
$ cat /mnt/vg2-lvol0/foo
test

$ sudo dd if=/dev/lab-vg2/lvol0 of=newtest bs=1M count=14
14+0 records in
14+0 records out
14680064 bytes (15 MB, 14 MiB) copied, 15.7727 s, 931 kB/s

$ df -h
```

```
Filesystem                  Size  Used Avail Use% Mounted on
tmpfs                       196M  1.5M  195M   1% /run
/dev/sda3                    24G   12G   11G  53% /
tmpfs                       980M     0  980M   0% /dev/shm
tmpfs                       5.0M  4.0K  5.0M   1% /run/lock
/dev/sda2                   512M  6.1M  506M   2% /boot/efi
tmpfs                       196M   92K  196M   1% /run/user/1000
/dev/mapper/lab--vg2-lvol0   15M   15M     0 100% /mnt/vg2-lvol0
/dev/mapper/lab--vg2-snap    15M   28K   14M   1% /mnt/snap
```

TASK 6: PROVISION A THIN VOLUME AND SNAPSHOT IT

> 1.Remove all Logical Volumes from Volume Group lab-vg2.

```
$ sudo umount /dev/lab-vg2/lvol0

$ sudo lvchange -an /dev/lab-vg2/*

$ sudo lvremove lab-vg2
Do you really want to remove and DISCARD logical volume lab-vg2/lvol0? [y/n]:
y
  Logical volume "lvol0" successfully removed
```

> 2.Follow the explanations in the Ubuntu manual on lvmthin to create
>
> - 2.1.a thin data Logical Volume called pool0 of 28 MB
> - 2.2.a thin metadata Logical Volume called pool0meta of 4 MB

```
$ sudo lvcreate -n pool0 -L 28M lab-vg2
WARNING: ext4 signature detected on /dev/lab-vg2/pool0 at offset 1080. Wipe
it? [y/n]: y
  Wiping ext4 signature on /dev/lab-vg2/pool0.
  Logical volume "pool0" created.

$ sudo lvcreate -n pool0meta -L 4M lab-vg2
  Logical volume "pool0meta" created.
```

> 3.Combine the two into a thin pool Logical Volume. List the Logical Volumes using lvs. Use the -a option to list also the hidden ones.

```
$ sudo lvconvert --type thin-pool --poolmetadata lab-vg2/pool0meta lab-
vg2/pool0
  Thin pool volume with chunk size 64.00 KiB can address at most 15.81 TiB of
data.
  WARNING: Converting lab-vg2/pool0 and lab-vg2/pool0meta to thin pool's data
and metadata volumes with metadata wiping.
  THIS WILL DESTROY CONTENT OF LOGICAL VOLUME (filesystem etc.)
Do you really want to convert lab-vg2/pool0 and lab-vg2/pool0meta? [y/n]: y
  Converted lab-vg2/pool0 and lab-vg2/pool0meta to thin pool.

$ sudo lvs -a
  LV              VG      Attr       LSize  Pool Origin Data%  Meta%  Move Log
Cpy%Sync Convert
  lvol0           lab-vg1 -wi-a----- 20.00m
  [lvol0_pmspare] lab-vg2 ewi-------  4.00m
  pool0           lab-vg2 twi-a-tz-- 28.00m             0.00   10.84
  [pool0_tdata]   lab-vg2 Twi-ao---- 28.00m
  [pool0_tmeta]   lab-vg2 ewi-ao----  4.00m
```

> 4.Create a thin Logical Volume from the thin pool named thin1 and give it a size of 80 MB, although the thin pool only has 28 MB capacity. What warnings to you see?

```
$ sudo lvcreate -n thin1 -V 80M --thinpool pool0 lab-vg2

  WARNING: Sum of all thin volume sizes (80.00 MiB) exceeds the size of thin
pool lab-vg2/pool0 and the size of whole volume group (40.00 MiB).

  WARNING: You have not turned on protection against thin pools running out of
space.

  WARNING: Set activation/thin_pool_autoextend_threshold below 100 to trigger
automatic extension of thin pools before they get full.

  Logical volume "thin1" created.
```

The following warnings means :

The first one : indicates that the total size of thin volumes requested (80.00 MiB) exceeds both the size of the thin pool ("pool0") and the size of the entire volume group ("lab-vg2")

The second one : means that there is no pretection to prevent thin pools from running out of space. It's then possible for it to become full, it might lead to data loss or disruption of operations that depend on the thin pool

The thrid one : advises setting the activation/thin_pool_autoextend_threshold parameter below 100 to trigger automatic extension of thin pools before they become full.

> 5.Create an ext4 file system on thin1. Mount the file system. How much capacity does df -h see in the file system?

```
$ sudo mkfs.ext4 /dev/lab-vg2/thin1
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 20480 4k blocks and 20480 inodes

Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

$ df -h
Filesystem                  Size  Used Avail Use% Mounted on
tmpfs                       196M  1.5M  195M   1% /run
/dev/sda3                    24G   12G   11G  53% /
tmpfs                       980M     0  980M   0% /dev/shm
tmpfs                       5.0M  4.0K  5.0M   1% /run/lock
/dev/sda2                   512M  6.1M  506M   2% /boot/efi
tmpfs                       196M   92K  196M   1% /run/user/1000
/dev/mapper/lab--vg2-thin1   71M   24K   66M   1% /mnt/thin
```

On df -h see 71M capacity for this file system.

> 6. Do experiments: Fill the file system with a bit of data by using dd to write files and verify that it behaves normally. Then write more and more data until you cross the size of the thin pool and see what happens. You can see LVM's log messages by using the dmesg command, they appear as device-mapper. What do you observe?

```
$ sudo dd if=/dev/zero of=/mnt/lab-vg2/thin1/testfile bs=1M count=5
5+0 records in
5+0 records out
5242880 bytes (5.2 MB, 5.0 MiB) copied, 0.0058668 s, 894 MB/s
```

```
$ sudo dd if=/dev/zero of=/mnt/lab-vg2/thin1/testfile2 bs=1M count=10
10+0 records in
10+0 records out
10485760 bytes (10 MB, 10 MiB) copied, 0.0120071 s, 873 MB/s

$ sudo dmesg | grep device-mapper

[    0.527830] device-mapper: core: CONFIG_IMA_DISABLE_HTABLE is disabled.
Duplicate IMA measurements will not be recorded in the IMA log.

[    0.527839] device-mapper: uevent: version 1.0.3

[    0.528026] device-mapper: ioctl: 4.48.0-ioctl (2023-03-01) initialised:
dm-devel@redhat.com

[ 2793.954229] device-mapper: thin: Data device (dm-2) discard unsupported:
Disabling discard passdown.

[ 2826.556547] device-mapper: thin: Data device (dm-2) discard unsupported:
Disabling discard passdown.

[ 4334.735831] device-mapper: thin: Data device (dm-2) discard unsupported:
Disabling discard passdown.

[ 4335.085779] device-mapper: thin: Data device (dm-2) discard unsupported:
Disabling discard passdown.

[ 5121.494862] device-mapper: thin: 252:3: reached low water mark for data
device: sending event.

[ 5121.516711] device-mapper: thin: 252:3: switching pool to out-of-data-space
(queue IO) mode

[ 5182.426473] device-mapper: thin: 252:3: switching pool to out-of-data-space
(error IO) mode
```

device-mapper: thin: Data device (dm-2) discard unsupported: Disabling discard passdown : This message indicates that the thin provisioned pool doesn't support the discard operation, which is a mechanism for releasing unused data blocks.

device-mapper: thin: reached low water mark for data device: sending event : This message indicate that available space is running low.

device-mapper: thin: switching pool to out-of-data-space (queue IO) mode: This message indicates that the thin provisioned pool is switching to out-of-data-space mode, which implies that it's queuing IO operations.

These messages gives us informations on how the pool initially handles writes normally, but as it nears capacity, it starts queuing IO operations and eventually enters an error state, resulting in failed IO operations.

TASK 7: SCENARIO

> You are a data engineer in a company. You need to set up the backup system for your production system, which runs a large database. Backups are important in case of unexpected incidents or human error. As the volume of data is significant, the backup process's duration can be fairly long, typically between 30 minutes and 1 hour.
> Design a solution for this backup system using snapshots, knowing that:
>
> - The backup should be performed without interrupting the database operations.
> - The backup files need to be physically distant and sent to another datacenter located a few kilometers away.
>
> Describe a potential solution and explain your thought process

A simple solution we can implement is by using LVM like we did in this lab to create snapshots of the DB. Those snapshots shall be scheduled to be made during off peak hours to minimize impact on performance of the DB.

Then transfert over SSH those snapshots to another datacenter located kilometers away to ensure data integrity during the transit.

Using monitoring tools we can ensure that the creation process is done successfully and track data transfer.

Using the 3-2-1 rule that is a data protection strategy that recommends having three copies of your data, stored on two different types of media, with one copy kept off-site.

We can also install an appliance of NetBackup to manage the backup.

(4/6): Good! Mising aspect: Initial full backup, delta changes and incremental backup