Backup Volume Management

Scenario

Your organization has introduced a new backup strategy. You have received two new 5 GiB disks. However, since no additional physical disks are attached to your current system, you will simulate these disks using loopback devices and files.

Your tasks include:

- Configuring the loop devices for use with LVM (Logical Volume Manager)
- Creating logical volumes for different backup purposes
- Mounting them persistently
- Demonstrating how to grow and shrink the logical volumes safely

1. Create Virtual Disks (Loop Devices)

```
mkdir -p ~/lvm-lab
cd ~/lvm-lab
fallocate -l 5G disk1.img
fallocate -l 5G disk2.img
```

Command 1: mkdir -p ~/lvm-lab

Component	Description
mkdir	Command to create a new directory.
-p	Ensures that parent directories are created as needed; avoids error if the directory already exists.
~	Represents the current user's home directory.
/lvm-lab	The name of the directory to create within the home directory.

Command 2: cd ~/lvm-lab

Component	Description
cd	Command to change the current working directory.
~	Refers to the user's home directory.
/lvm-lab	The target directory to navigate into.

Command 3: fallocate -l 5G disk1.img

Component	Description
fallocate	Preallocates space for a file on disk without filling it with data; used to quickly create disk images.
-l	Specifies the length (size) of the file to allocate.
5G	File size of 5 Gigabytes.
disk1.img	Name of the file to create (disk image).

Command 4: fallocate -l 5G disk2.img

Component	Description
fallocate	Same as above.
-l	Same as above.
5G	Same file size (5 Gigabytes).
disk2.img	Second disk image file name.

```
tiago@Linux:~$ mkdir -p ~/lvm-lab
tiago@Linux:~$ cd ~/lvm-lab
tiago@Linux:~/lvm-lab$ fallocate -l 5G disk1.img
tiago@Linux:~/lvm-lab$ fallocate -l 5G disk2.img
```

Check the created files:
ls -lh disk*.img

Command: ls -lh disk*.img

Component	Description
ls	Lists directory contents (files and directories).
-l	Uses long listing format; shows detailed file information such as permissions, owner, size, and modification date.
-h	"Human-readable" option; displays file sizes in KB, MB, GB instead of bytes. Requires `-l` to be useful.
disk*.img	Shell wildcard pattern: matches all files that start with `disk`, followed by any characters (`*`), and ending in `.img`.

```
tiago@Linux:~/lvm-lab$ ls -lh disk*.img
====================================================================
-rw-rw-r-- 1 tiago-paquete tiago-paquete 5.0G May  3 10:13 disk1.img
-rw-rw-r-- 1 tiago-paquete tiago-paquete 5.0G May  3 10:13 disk2.img
====================================================================
```

2. Attach Loop Devices

```
sudo losetup -fP disk1.img
sudo losetup -fP disk2.img
```

Command 1: sudo losetup -fP disk1.img

Component	Description
sudo	Executes the command with superuser (root) privileges; required for loop device operations.
losetup	Command to set up and control loop devices (virtual block devices backed by regular files).
-f	Automatically finds the first available (unused) loop device.
-P	Forces kernel to scan for partitions on the associated loop device after setup.
disk1.img	The disk image file to associate with a loop device (e.g., `/dev/loop0`).

Command 2: sudo losetup -fP disk2.img

Component	Description
sudo	Same as above.
losetup	Same as above.
-f	Same as above.
-P	Same as above.
disk2.img	Second disk image file to associate with another available loop device (e.g., `/dev/loop1`).

```
tiago@Linux:~/lvm-lab$ sudo losetup -fP disk1.img
=========================================================================
[sudo] password for tiago-paquete:
=========================================================================
tiago@Linux:~/lvm-lab$ sudo losetup -fP disk2.img
```

Check attached loop devices:

```
sudo losetup -a | grep -E 'disk(1|2)\.img'
```

Command: sudo losetup -a | grep -E 'disk(1|2)\.img'

Component	Description
sudo	Runs the command with superuser (root) privileges. Required for many system-level operations, including `losetup`.
losetup	Utility used to configure and manage loopback devices (virtual block devices).
-a	Option for `losetup` to list **all loop devices** currently set up, along with their associated backing files.
|	Sends the output of the command on the left (`losetup -a`) as the input to the command on the right (`grep`).
grep	A command-line utility for searching text using patterns (regular expressions).
-E	Enables extended regular expression (ERE) syntax in `grep`, allowing more complex patterns like `(1|2)` without needing escape characters.
disk(1|2)\.img'	The extended regular expression used to match strings that contain either `disk1.img` or `disk2.img`.
\.img	Matches the literal string `.img`. The backslash (`\`) is used to escape the dot so it is treated as a literal period instead of a wildcard.

```
tiago@Linux:~/lvm-lab$ sudo losetup -a | grep -E 'disk(1|2)\.img'
==========================================================================
/dev/loop15: [66309]:1977986 (/home/tiago-paquete/lvm-lab/disk1.img)
/dev/loop16: [66309]:1977987 (/home/tiago-paquete/lvm-lab/disk2.img)
==========================================================================
```

3. Partition the Disks for LVM

```
for d in /dev/loop15 /dev/loop16; do
  sudo parted --script "$d" mklabel gpt \
      mkpart primary 0% 100% \
      set 1 lvm on
done
```

1. for d in /dev/loop15 /dev/loop16; do ... done
Purpose: This is a for loop that iterates over two loop devices: /dev/loop15 and /dev/loop16.
d: A temporary variable representing the current device path in each iteration.
Action: For each device, the script executes a set of partitioning commands using parted.

2. sudo parted --script "$d" ...
sudo: Runs the command as root, required for disk operations.
parted: A powerful command-line tool used to create, destroy, resize, check, and copy disk partitions.
--script: Runs parted in non-interactive mode (no prompts), suitable for scripting.
So we're silently performing partitioning actions on each loop device.

3. mklabel gpt
Action: Creates a new **partition table** on the device.
gpt: Refers to the **GUID Partition Table**, a modern partitioning scheme that supports:
More than 4 partitions
Large disk sizes (>2TB)
UEFI boot compatibility
Result: This erases any existing partition table on the disk and replaces it with a GPT structure.

⚠️ Important: This will destroy all existing data on the disk.

4. mkpart primary 0% 100%
Action: Creates a new partition of type primary.
0%: Start at the beginning of the disk (0% of total disk space).
100%: End at the end of the disk (100% of total disk space).
Effect: The partition spans the **entire disk**.

🧠 In GPT, all partitions are technically "primary" — the term is used here for compatibility.

5. set 1 lvm on
Action: Sets a **flag** on the first partition (1) of the disk to indicate it's intended for **LVM (Logical Volume Manager)**.
LVM flag:
Signals to tools (like installers or pvcreate) that this partition is for use by LVM.
It does **not** create LVM metadata; it's just a **partition-level flag**.

```
tiago@Linux:~/lvm-lab$ for d in /dev/loop15 /dev/loop16; do
  sudo parted --script "$d" mklabel gpt \
      mkpart primary 0% 100% \
      set 1 lvm on
done
```

Validate the partitions:

lsblk -f | grep loop1[5-6]

Command: lsblk -f | grep loop1[5-6]

Component	Description
lsblk	Lists information about all available or the specified block devices.
-f	Option for `lsblk` that displays filesystem information (e.g., UUID, LABEL, TYPE, and MOUNTPOINT).
|	Pipe
grep	Searches for lines matching a given pattern in the input it receives.
loop1[5-6]	A regular expression pattern used by `grep` to match strings like `loop15` and `loop16`.
[5-6]	Regex character class that matches either the digit 5 or 6 (used in this case for matching loop15/16).

```
tiago@Linux:~/lvm-lab$ lsblk -f | grep loop1[5-6]
======================================================================
loop15
 └─loop15p1
loop16
 └─loop16p1
======================================================================
```

4. Create Physical Volumes (PVs)

sudo pvcreate /dev/loop15p1 /dev/loop16p1

Command: sudo pvcreate /dev/loop15p1 /dev/loop16p1

Component	Description
sudo	Runs the command with **superuser privileges**. Required for LVM operations.
pvcreate	LVM command to **initialize physical volumes** for use in volume groups (VGs).
/dev/loop15p1	**Device path** to the **first partition** of the loopback device `loop15`.
/dev/loop16p1	**Device path** to the **first partition** of the loopback device `loop16`.

```
tiago@Linux:~/lvm-lab$ sudo pvcreate /dev/loop15p1 /dev/loop16p1
==================================================================
  Physical volume "/dev/loop15p1" successfully created.
  Physical volume "/dev/loop16p1" successfully created
==================================================================
```

Check:
sudo pvs

Command: sudo pvs

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
pvs	**Physical Volumes Summary** – displays information about LVM (Logical Volume Manager) physical volumes on the system, such as their names, sizes, and associated volume groups. Part of the `lvm2` package.

```
tiago@Linux:~/lvm-lab$ sudo pvs
==================================================================
  PV            VG Fmt  Attr PSize  PFree
  /dev/loop15p1    lvm2 ---  <5.00g <5.00g
  /dev/loop16p1    lvm2 ---  <5.00g <5.00g
==================================================================
```

5. Create a Volume Group (VG)

```
sudo vgcreate backup /dev/loop15p1 /dev/loop16p1
```

Command: sudo vgcreate backup /dev/loop15p1 /dev/loop16p1

Component	Description
sudo	**Superuser Do** – runs the following command with administrative (root) privileges. Required because LVM operations affect system storage.
vgcreate	**Volume Group Create** – LVM command to create a new volume group using one or more physical volumes.
backup	**Volume Group Name** – the name to assign to the new volume group being created. You choose this name; here it is `backup`.
/dev/loop15p1	**First Physical Volume (Partition)** – refers to the first partition of a loopback device. This is treated as a physical volume by LVM.
/dev/loop16p1	**Second Physical Volume (Partition)** – another partition on a loopback device, also treated as a physical volume to include in the volume group.

```
tiago@Linux:~/lvm-lab$ sudo vgcreate backup /dev/loop15p1 /dev/loop16p1
===========================================================================
  Volume group "backup" successfully created
===========================================================================
```

Validate:

```
sudo vgdisplay backup
ls /etc/lvm/backup/
```

Command: sudo vgdisplay backup

Component	Description
sudo	**Superuser Do** – executes the following command with administrative (root) privileges. Necessary for accessing system-level LVM information.
vgdisplay	**Volume Group Display** – shows detailed information about a specified LVM volume group (e.g., size, physical volumes, logical volumes, etc.).
backup	**Volume Group Name** – this is the name of the volume group whose information is to be displayed. In this case, it refers to a VG named `backup`.

Command: ls /etc/lvm/backup/

Component	Description
ls	**List** – lists files and directories in the specified location.
/etc/lvm/backup/	**Directory Path** – a system directory that contains backup metadata files for LVM volume groups. These files are auto-generated by LVM to help recover metadata if needed.

```
tiago@Linux:~/lvm-lab$ sudo vgdisplay backup
==========================================================================
  --- Volume group ---
  VG Name               backup
  System ID
  Format                lvm2
  Metadata Areas        2
  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                2
  Act PV                2
  VG Size               9.99 GiB
  PE Size               4.00 MiB
  Total PE              2558
  Alloc PE / Size       0 / 0
  Free  PE / Size       2558 / 9.99 GiB
  VG UUID               9NcpwQ-tvLr-wsMT-tjD0-RFCw-rprE-wgoXAu
==========================================================================
```

```
tiago@Linux:~/lvm-lab$ ls /etc/lvm/backup/
==========================================================================
ls: cannot open directory '/etc/lvm/backup/': Permission denied
==========================================================================
```

```
tiago@Linux:~/lvm-lab$ sudo !!
==========================================================================
sudo ls /etc/lvm/backup/
backup
==========================================================================
```

6. Create Logical Volumes (LVs)

```
sudo lvcreate --name sysbk  --size 2G backup
sudo lvcreate --name databk --size 2G backup
```

Command: sudo lvcreate --name sysbk --size 2G backup

Component	Description
sudo	**Superuser Do** – runs the command as the root user for administrative access.
lvcreate	**Logical Volume Create** – used to create a new logical volume within an existing volume group.
--name	**Logical Volume Name Option** – followed by the desired name for the logical volume.
sysbk	**Logical Volume Name** – the name of the new logical volume being created (`sysbk`).
--size	**Size Option** – specifies the size of the logical volume to be created.
2G	**Size Value** – the size of the logical volume (`2 Gigabytes`).
backup	**Volume Group Name** – indicates the volume group (`backup`) in which the logical volume will be created.

Command: sudo lvcreate --name databk --size 2G backup

Component	Description
sudo	**Superuser Do** – executes the command with root privileges.
lvcreate	**Logical Volume Create** – initiates creation of a logical volume in LVM.
--name	**Logical Volume Name Option** – indicates the name of the new logical volume.
databk	**Logical Volume Name** – the new logical volume will be named `databk`.
--size	**Size Option** – sets the desired size of the logical volume.
2G	**Size Value** – assigns 2 Gigabytes to the logical volume.
backup	**Volume Group Name** – specifies the target volume group (`backup`) for the new logical volume.

```
tiago@Linux:~/lvm-lab$ sudo lvcreate --name sysbk --size 2G backup
=====================================================================
  Logical volume "sysbk" created.
=====================================================================

tiago@Linux:~/lvm-lab$ sudo lvcreate --name databk --size 2G backup
=====================================================================
  Logical volume "databk" created.
=====================================================================
```

Check:
sudo lvs

Command: `sudo lvs`

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Needed for accessing system-level LVM data.
lvs	**Logical Volumes Summary** – displays a summarized list of all Logical Volumes in the system along with attributes like size, volume group, and status. It is part of the `lvm2` package.

```
tiago@Linux:~/lvm-lab$ sudo lvs
======================================================================
  LV     VG     Attr       LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  databk backup -wi-a----- 2.00g
  sysbk  backup -wi-a----- 2.00g
======================================================================
```

Output Breakdown

Column	Meaning
LV	**Logical Volume Name** – Name of the logical volume (e.g., `databk`, `sysbk`).
VG	**Volume Group Name** – The volume group the logical volume belongs to (e.g., `backup`).
Attr	**Attributes** – A 10-character code describing volume properties (explained below).
LSize	**Logical Size** – Size of the logical volume (e.g., `2.00g` = 2 gigabytes).
Pool	**Thin Pool** – If this LV is using a thin pool, the pool name appears here.
Origin	**Origin LV** – For snapshot volumes, this shows the origin LV (empty here).
Data%	**Data Usage** – Percent of the data area used if it's a thin volume (not applicable here).
Meta%	**Metadata Usage** – Percent of metadata used in a thin volume (also not used here).
Move	**Move Operation** – If data is being moved, this shows status (empty = no movement).
Log	**Mirror Log** – If the LV is mirrored, this shows the log volume (not used here).
Cpy%Sync	**Copy % Sync** – For mirrored or snapshot volumes, sync progress (empty = not syncing).
Convert	**Conversion** – Status of volume conversion (e.g., to thin provisioning) (empty = none).

Attr Field (-wi-a-----) Breakdown

This is a 10-character code representing various properties of the logical volume:

Char Pos	Char	Meaning
1	-	Volume type (- = linear)
2	w	Permissions (w = writeable)
3	i	Allocation policy (i = inherited)
4	-	Fixed minor number (rarely used)
5	a	Activation (a = active)
6–10	-----	Other rarely used flags (e.g., snapshot, thin)

7. Format the Logical Volumes

sudo mkfs.xfs /dev/backup/sysbk
sudo mkfs.ext4 /dev/backup/databk

Command: sudo mkfs.xfs /dev/backup/sysbk

Component	Description
sudo	**Superuser Do** – runs the command with elevated privileges needed for disk formatting.
mkfs.xfs	**Make Filesystem (XFS)** – creates a new XFS filesystem on the specified device. XFS is a high-performance journaling filesystem.
/dev/backup/sysbk	**Device Path** – refers to the target block device or partition where the XFS filesystem will be created. Typically, this is a logical volume or disk partition.

Command: sudo mkfs.ext4 /dev/backup/databk

Component	Description
sudo	**Superuser Do** – allows execution of privileged disk operations.
mkfs.ext4	**Make Filesystem (ext4)** – creates a new ext4 filesystem on the given device. ext4 is a widely used default Linux filesystem.
/dev/backup/databk	**Device Path** – the target device or partition to format with an ext4 filesystem, likely a logical volume.

```
tiago@Linux:~/lvm-lab$ mkfs.xfs /dev/backup/sysbk
======================================================================
mkfs.xfs: cannot open /dev/backup/sysbk: Permission denied
======================================================================

tiago@Linux:~/lvm-lab$ sudo !!
======================================================================
sudo mkfs.xfs /dev/backup/sysbk
[sudo] password for tiago-paquete:
meta-data=/dev/backup/sysbk      isize=512    agcount=4, agsize=131072 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=1
         =                       reflink=1    bigtime=1 inobtcount=1 nrext64=0
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=16384, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
Discarding blocks…Done.
======================================================================
```

Key output details:

- Metadata and data layout was initialized (e.g., agcount=4, blocks=524288).
- Advanced features like reflink, rmapbt, crc, and bigtime are enabled – indicating support for modern filesystem capabilities.
- Internal journaling is used (log=internal log).
- The device was discarded (wiped) before formatting (Discarding blocks...Done.).

```
tiago@Linux:~/lvm-lab$ mkfs.ext4 /dev/backup/databk
=========================================================================
mke2fs 1.47.0 (5-Feb-2023)
mkfs.ext4: Permission denied while trying to determine filesystem size
=========================================================================
```

```
tiago@Linux:~/lvm-lab$ sudo !!
=========================================================================
sudo mkfs.ext4 /dev/backup/databk
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: 5f4b3f4e-9690-42dd-b967-17924d4703f6
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

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

Key output details:

- 524,288 blocks and 131,072 inodes created.
- Filesystem UUID assigned for unique identification.
- Superblock backups were placed at specific block locations for recovery.
- Journaling was created (16,384 blocks), and inode/superblock writing completed.

Confirm:

```
sudo blkid /dev/backup/*
```

Command: sudo blkid /dev/backup/*

Component	Description
sudo	**Superuser Do** – allows execution of commands that require access to device metadata.
blkid	**Block ID** – displays information about block devices, including UUIDs, filesystem type, and labels.
/dev/backup/*	**Wildcard Device Path** – matches all devices under `/dev/backup/` (e.g., `sysbk`, `databk`), allowing `blkid` to return info on each.

```
tiago@Linux:~/lvm-lab$ sudo blkid /dev/backup/*
=========================================================================
/dev/backup/databk: UUID="5f4b3f4e-9690-42dd-b967-17924d4703f6" BLOCK_SIZE="4096"
TYPE="ext4"
/dev/backup/sysbk: UUID="ccf71e26-0567-46f3-b283-6a483046e6da" BLOCK_SIZE="512"
TYPE="xfs"
=========================================================================
```

8. Create Mount Points and Mount the LVs

```
sudo mkdir -p /backup/sysbk /backup/databk
echo '/dev/backup/sysbk  /backup/sysbk  xfs   defaults  0 2' | sudo tee -a /etc/fstab
echo '/dev/backup/databk /backup/databk ext4  defaults  0 2' | sudo tee -a /etc/fstab
sudo mount -a
```

Command: sudo mkdir -p /backup/sysbk /backup/databk

Component	Description
sudo	**Superuser Do** – allows execution of commands that require access to restricted directories or system-level modifications.
mkdir	**Make Directory** – creates one or more directories.
-p	**Parents** – ensures that parent directories are created as needed and suppresses errors if directories already exist.
/backup/sysbk	**Directory Path** – the full path of the directory for mounting the `sysbk` volume.
/backup/databk	**Directory Path** – the full path of the directory for mounting the `databk` volume.

Command: echo '/dev/backup/sysbk /backup/sysbk xfs defaults 0 2' | sudo tee -a /etc/fstab

Component	Description
echo	**Echo** – prints the given string to standard output.
/dev/backup/sysbk /backup/sysbk xfs defaults 0 2'	**fstab Entry** – a line representing the filesystem mount configuration: /dev/backup/sysbk (device), /backup/sysbk (mount point), xfs (filesystem), defaults (options), 0 (dump), 2 (fsck order).
sudo	**Superuser Do** – required to append content to a system file like `/etc/fstab`.
tee	**Tee** – reads from standard input and writes to both standard output and a file.
-a	**Append** – appends the input to the file instead of overwriting it.
/etc/fstab	**fstab File** – configuration file used to define how and where filesystems are mounted at boot.

Command: echo '/dev/backup/databk /backup/databk ext4 defaults 0 2' | sudo tee -a /etc/fstab

Component	Description
echo	**Echo** – prints the string to standard output.
/dev/backup/databk /backup/databk ext4 defaults 0 2'	**fstab Entry** – entry to mount the `databk` device with `ext4` filesystem using default options.
sudo	**Superuser Do** – runs the `tee` command as root.
tee	**Tee** – appends the string to the `/etc/fstab` file.
-a	**Append** – appends to the file without deleting its current contents.
/etc/fstab	**fstab File** – defines static mount points for filesystems.

Command: sudo mount -a

Component	Description
sudo	**Superuser Do** – required to perform mount operations system-wide.
mount	**Mount** – attaches a filesystem to the directory tree at a specified mount point.
-a	**All** – mounts all filesystems defined in `/etc/fstab` that aren't already mounted.

```
tiago@Linux:~/lvm-lab$ sudo mkdir -p /backup/sysbk /backup/databk
tiago@Linux:~/lvm-lab$ echo '/dev/backup/sysbk /backup/sysbk xfs defaults 0 2' | sudo tee -a /etc/fstab
========================================================================
dev/backup/sysbk /backup/sysbk xfs defaults 0 2f
========================================================================

tiago@Linux:~/lvm-lab$ echo '/dev/backup/databk /backup/databk ext4 defaults 0 2' | sudo tee -a /etc/fstab
========================================================================
/dev/backup/databk /backup/databk ext4 defaults 0 2
========================================================================

tiago@Linux:~/lvm-lab$ sudo mount -a
```

Verify:

```
df -h | grep /backup
ls /mnt/loop
```

Command: df -h | grep /backup

Component	Description
df	**Disk Free** – displays disk space usage for mounted filesystems.
-h	**Human-readable** – shows sizes in KB, MB, GB etc. for easier interpretation.
grep	**Global Regular Expression Print** – filters and shows only lines that match the specified string.
/backup	**Search Pattern** – filters the output to show only mount points under `/backup`.

```
tiago@Linux:~/lvm-lab$ df -h | grep /backup
========================================================================
/dev/mapper/backup-databk  2.0G   24K  1.8G   1% /backup/databk
========================================================================

tiago@Linux:~/lvm-lab$ ls /mnt/loop
========================================================================
data  system
========================================================================
```

9. Grow the databk LV from 2 GiB to 3 GiB

```
sudo lvextend -L 3G /dev/backup/databk
sudo resize2fs /dev/backup/databk
```

LVM Layer (lvextend): Adds capacity at the logical block level — the volume is physically bigger now.
Filesystem Layer (resize2fs): The filesystem inside still "thinks" it's 2 GiB unless it’s told to stretch out and occupy the new 3 GiB volume space.

Command: sudo lvextend -L 3G /dev/backup/databk

Component	Description
sudo	**Superuser Do** – executes the command with root privileges. Needed for LVM operations.
lvextend	**Logical Volume Extend** – command used to increase the size of an existing logical volume.
-L	**Size option** – specifies the new absolute size of the logical volume.
3G	**New size** – the logical volume will be resized to 3 gigabytes.
/dev/backup/databk	**Logical Volume Path** – the full device path to the logical volume that is being resized. `backup` is the Volume Group (VG), and `databk` is the Logical Volume (LV).

Command: sudo resize2fs /dev/backup/databk

Component	Description
sudo	**Superuser Do** – runs the command with elevated privileges. Required to modify filesystems.
resize2fs	**Resize ext2/3/4 Filesystem** – adjusts the size of an ext-based filesystem to match the size of the logical volume.
/dev/backup/databk	**Logical Volume Path** – the full path to the logical volume whose filesystem is being resized. Must be mounted or checked depending on usage.

```
tiago@Linux:~/lvm-lab$ sudo lvextend -L 3G /dev/backup/databk
=======================================================================
[sudo] password for tiago-paquete:
  Size of logical volume backup/databk changed from 2.00 GiB (512 extents) to 3.00 GiB (768 extents).
  Logical volume backup/databk successfully resized.
=======================================================================

tiago@Linux:~/lvm-lab$ sudo resize2fs /dev/backup/databk
=======================================================================
resize2fs 1.47.0 (5-Feb-2023)
Filesystem at /dev/backup/databk is mounted on /backup/databk; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/backup/databk is now 786432 (4k) blocks long.
=======================================================================
```

Check:
sudo lvdisplay /dev/backup/databk
df -h /backup/databk

Command: sudo lvdisplay /dev/backup/databk

Component	Description
sudo	**Superuser Do** – runs the command with root permissions. Necessary for querying LVM metadata.
lvdisplay	**Logical Volume Display** – shows detailed information about logical volumes.
/dev/backup/databk	**Logical Volume Path** – specific path of the logical volume whose details you want to view.

Command: df -h /backup/databk

Component	Description
df	**Disk Free** – reports file system disk space usage.
-h	**Human-readable** – shows sizes in easily readable format (e.g., KB, MB, GB).
/backup/databk	**Mount Point** – the path to the mounted filesystem (directory) whose space usage is being checked. This should correspond to the mounted logical volume.

```
tiago@Linux:~/lvm-lab$ sudo lvdisplay /dev/backup/databk
=========================================================================
  --- Logical volume ---
  LV Path                /dev/backup/databk
  LV Name                databk
  VG Name                backup
  LV UUID                TtSiHG-SI7b-7qSV-LIAr-uUVY-1Pgd-VGSDSy
  LV Write Access        read/write
  LV Creation host, time tiago-paquete-Linux, 2025-05-03 11:30:02 +0200
  LV Status              available
  # open                 1
  LV Size                3.00 GiB
  Current LE             768
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           252:1
=========================================================================
```

```
tiago@Linux:~/lvm-lab$ df -h /backup/databk
=========================================================================
Filesystem                 Size  Used Avail Use% Mounted on
/dev/mapper/backup-databk  2.9G   24K  2.8G   1% /backup/databk
=========================================================================
```

10. Shrink the databk LV to 1 GiB

```
sudo umount /backup/databk
sudo e2fsck -f /dev/backup/databk
sudo resize2fs /dev/backup/databk 1G
sudo lvreduce -L 1G /dev/backup/databk
sudo mount /backup/databk
sudo systemctl daemon-reload
```

Why we Must Unmount to Reduce but Not to Increase a Filesystem:

Resizing (Increasing) Without Unmounting

- Filesystem Growth is Safe While Mounted: Most modern filesystems (like ext4) allow online resizing (growing)because adding more space doesn't risk corrupting data already in use.
- The OS just extends what's available: The filesystem's structures can usually be extended dynamically without interrupting access.

Resizing (Reducing) Requires Unmounting

- Shrinking is Risky While Mounted: When shrinking, the system must rearrange or remove data blocks and metadata structures to fit into a smaller space.
- Data Loss Risk: If the filesystem is mounted and actively used, this can lead to data corruption or loss.
- Unmounting Ensures Integrity: By unmounting, you're ensuring no files are being accessed or written during the shrink operation, allowing tools like resize2fs to safely move data and reduce size.

Command: sudo umount /backup/databk

Component	Description
sudo	**Superuser Do** – runs the command with elevated (root) privileges.
umount	**Unmount** – detaches a mounted filesystem from the directory tree.
/backup/databk	**Mount Point** – the directory where the filesystem is currently mounted and will be unmounted.

Command: sudo e2fsck -f /dev/backup/databk

Component	Description
sudo	**Superuser Do** – runs the command with elevated privileges.
e2fsck	**Ext2 Filesystem Check** – checks a Linux ext2/ext3/ext4 filesystem for errors.
-f	**Force Check** – forces a check even if the filesystem appears clean.
/dev/backup/databk	**Block Device Path** – the logical volume or partition to be checked.

Command: sudo resize2fs /dev/backup/databk 1G

Component	Description
sudo	**Superuser Do** – runs the command with elevated privileges.
resize2fs	**Resize ext2/3/4 Filesystem** – resizes the filesystem on the specified device.
/dev/backup/databk	**Device Path** – the logical volume or partition whose filesystem will be resized.
1G	**Target Size** – the new size for the filesystem (1 gigabyte).

Command: sudo lvreduce -L 1G /dev/backup/databk

Component	Description
sudo	**Superuser Do** – runs the command with elevated privileges.
lvreduce	**Logical Volume Reduce** – reduces the size of a logical volume.
-L	**Size Option** – specifies the target size to reduce to.
1G	**New Size** – the size to reduce the volume to (1 gigabyte).
/dev/backup/databk	**Logical Volume Path** – the LVM logical volume being reduced.

Command: sudo mount /backup/databk

Component	Description
sudo	**Superuser Do** – runs the command with elevated privileges.
mount	**Mount Filesystem** – attaches a filesystem to the directory tree.
/backup/databk	**Mount Point** – the directory where the filesystem will be mounted.

Command: sudo systemctl daemon-reload

Component	Description
sudo	**Superuser Do** – executes the command that follows with root privileges. Needed because `systemctl`controls system-level services.
systemctl	**Systemd Control Tool** – used to examine and control the `systemd` system and service manager.
daemon-reload	**Reload Daemon Configuration** – tells `systemd` to reload its configuration files. This is necessary after making changes to unit files (e.g., `.service` files) so that `systemd` is aware of the new or updated definitions.

```
tiago@Linux:~/lvm-lab$ sudo umount /backup/databk
tiago@Linux:~/lvm-lab$ sudo e2fsck -f /dev/backup/databk
==========================================================================
e2fsck 1.47.0 (5-Feb-2023)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/backup/databk: 11/196608 files (0.0% non-contiguous), 30268/786432 blocks
==========================================================================

tiago@Linux:~/lvm-lab$ sudo resize2fs /dev/backup/databk 1G
==========================================================================
resize2fs 1.47.0 (5-Feb-2023)
Resizing the filesystem on /dev/backup/databk to 262144 (4k) blocks.
The filesystem on /dev/backup/databk is now 262144 (4k) blocks long.
==========================================================================
tiago@Linux:~/lvm-lab$ sudo lvreduce -L 1G /dev/backup/databk
=======================================================================
  WARNING: Reducing active logical volume to 1.00 GiB.
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce backup/databk? [y/n]: y
  Size of logical volume backup/databk changed from 3.00 GiB (768 extents) to 1.00 GiB (256
extents).
  Logical volume backup/databk successfully resized.
=======================================================================
```

```
tiago@Linux:~/lvm-lab$ sudo mount /backup/databk
=======================================================================
mount: (hint) your fstab has been modified, but systemd still uses
       the old version; use 'systemctl daemon-reload' to reload.
=======================================================================
```

```
tiago@Linux:~/lvm-lab$ sudo systemctl daemon-reload
```

Confirm:
df -h /backup/databk
sudo vgs

Command: df -h /backup/databk

Component	Description
df	**Disk Free** – shows disk space usage of filesystems.
-h	**Human-Readable** – shows sizes in readable units (e.g., KB, MB, GB).
/backup/databk	**Path** – displays disk usage for the specified mount point or directory.

Command: sudo vgs

Component	Description
sudo	**Superuser Do** – runs the command with elevated privileges.
vgs	**Volume Groups Summary** – shows summary information about LVM volume groups.

```
tiago@Linux:~/lvm-lab$ df -h /backup/databk
=======================================================================
Filesystem                 Size  Used Avail Use% Mounted on
/dev/mapper/backup-databk  939M   24K  876M   1% /backup/databk
=======================================================================
```

```
tiago@Linux:~/lvm-lab$ sudo vgs
=======================================================================
  VG     #PV #LV #SN Attr   VSize VFree
  backup   2   2   0 wz--n- 9.99g 6.99g
=======================================================================
```

Validation and Inspection Commands

```
sudo lsblk
sudo pvs
sudo vgs
sudo lvs
sudo blkid
cat /etc/fstab
sudo losetup -a
ls /dev/mapper/
```

Component	Description
sudo	**Superuser Do** – Runs the command that follows with elevated (root) privileges. Used when a command requires administrative access.
lsblk	**List Block Devices** – Displays information about all available or specified block devices, excluding RAM disks.
pvs	**Physical Volumes Summary** – Shows info about physical volumes used in LVM, such as name, size, and volume group association.
vgs	**Volume Groups Summary** – Lists summary info about volume groups in LVM, like size and number of logical/physical volumes.
lvs	**Logical Volumes Summary** – Displays info about logical volumes, including their size and the volume group they belong to.
blkid	**Block ID Utility** – Prints block device attributes such as UUID, filesystem type, and label.
cat	**Concatenate and Display** – Reads files sequentially and writes their contents to standard output. Used here to display file content.
/etc/fstab	**Filesystem Table** – Configuration file that contains static information about filesystems and their mount points.
losetup	**Loop Setup** – Configures and displays loopback devices, used to mount files as block devices.
-a	**All Loop Devices** – Option with `losetup` to list all active loop devices and their backing files.
ls	**List** – Lists files and directories in the specified path.
/dev/mapper/	**Device Mapper Directory** – Contains device mapper entries, typically used in LVM for logical volume device nodes.

Optional Cleanup (If You Want to Remove Everything Later)

```
sudo umount /backup/sysbk /backup/databk
sudo lvremove -y /dev/backup/sysbk /dev/backup/databk
sudo vgremove -y backup
sudo pvremove /dev/loop15p1 /dev/loop16p1
sudo losetup -d /dev/loop15 /dev/loop16
rm -rf ~/lvm-lab
```

Component	Description
sudo	**Superuser Do** – Runs the command that follows with administrative privileges.
umount	**Unmount** – Detaches the specified filesystems from the filesystem hierarchy.
/backup/sysbk	Mount point or directory representing a mounted filesystem for system backups.
/backup/databk	Mount point or directory representing a mounted filesystem for data backups.
lvremove	**Logical Volume Remove** – Deletes specified logical volumes in LVM.
-y	**Assume Yes** – Automatically confirms the removal operation without user prompt.
/dev/backup/sysbk	Path to the logical volume `sysbk` inside the volume group `backup`.
/dev/backup/databk	Path to the logical volume `databk` inside the volume group `backup`.
vgremove	**Volume Group Remove** – Deletes the specified volume group after its logical volumes are removed.
backup	Name of the volume group to be removed.
pvremove	**Physical Volume Remove** – Wipes LVM metadata from the specified physical volumes.
/dev/loop15p1	Partition 1 on the loop device `/dev/loop15`, used as a physical volume.
/dev/loop16p1	Partition 1 on the loop device `/dev/loop16`, used as a physical volume.
losetup	**Loop Device Setup** – Manages loopback devices, which map files to block devices.
-d	**Detach** – Option to delete/disconnect the specified loop device mappings.
/dev/loop15	Loopback device representing a virtual block device.
/dev/loop16	Another loopback device.
rm	**Remove** – Deletes files or directories.
-r	**Recursive** – Deletes directories and their contents recursively.
-f	**Force** – Ignores nonexistent files and does not prompt before removal.
~/lvm-lab	Path to the `lvm-lab` directory in the user's home directory.

Summary of Learning Outcomes

- You simulated disk hardware using loop devices.
- You implemented the full LVM chain: Physical Volume → Volume Group → Logical Volume.
- You used XFS for fast and grow-only backups, and ext4 for flexibility (grow/shrink).
- You applied best practices for resizing:
  - Growing is online and safe.
  - Shrinking requires a safety check and offline operations.
  - You made the configuration persistent using /etc/fstab.