Linux LVM: fundamentals

Physical volumes, volume groups, logical volumes, filesystems and lifecycle operations

The storage stack

Logical Volume Manager (LVM2) separates the physical placement of blocks from the logical block devices used by filesystems and applications. A physical volume (PV) is a disk, partition, or suitable block device initialised for LVM. One or more PVs form a volume group (VG), a pool of allocatable extents. Logical volumes (LVs) draw extents from that pool and appear as block devices such as /dev/vgdata/home. A filesystem, swap area, database or other consumer can use an LV.

This separation permits resizing and moving storage without changing the application-visible device name. It does not make the underlying data redundant. A linear LV spanning two disks can fail if either disk fails. Backups, redundancy and monitoring are separate requirements.

Extents are LVM's allocation units. A VG has a physical extent size, commonly 4 MiB, and logical volumes are allocated in whole extents. Display free extents before planning changes.

```
pvs -o pv_name,vg_name,pv_size,pv_free
vgs -o vg_name,vg_size,vg_free,vg_extent_size
lvs -a -o lv_name,vg_name,lv_size,segtype,devices
lsblk -f
```

Creating a volume group and logical volume

First identify the intended disk by serial number and existing signatures. The following example assumes /dev/sdb is an unused disk. These commands write metadata to that device and are not safe against a disk containing data you wish to retain. A partition is also a valid PV; using a whole disk is a design choice, not a requirement.

After creating the LV, format it with the intended filesystem, create a mount point and mount it. Use filesystem UUIDs or stable mapper paths in /etc/fstab. Validate the mount before depending on it at boot.

```
lsblk -o NAME,SIZE,MODEL,SERIAL,FSTYPE,MOUNTPOINTS
sudo wipefs -n /dev/sdb
sudo pvcreate /dev/sdb
sudo vgcreate vgdata /dev/sdb
sudo lvcreate -L 20G -n documents vgdata
sudo mkfs.ext4 /dev/vgdata/documents
sudo mkdir -p /srv/documents
sudo mount /dev/vgdata/documents /srv/documents
findmnt /srv/documents
sudo blkid /dev/vgdata/documents
```

Growing capacity

There are three separate operations to understand: add capacity to the VG, extend the LV, and grow the filesystem. If an existing PV has been expanded at the hypervisor or block-device layer, pvresize makes LVM recognise its new size. Alternatively, create another PV and add it to the VG with vgextend.

lvextend changes the logical block device; the filesystem may still be its old size. The -r option asks LVM to run fsadm and resize the filesystem as part of the operation, where supported. For ext4 this can commonly be done

while mounted; XFS grows while mounted using xfs_growfs. Confirm both LV and filesystem sizes afterward.

```
sudo pvresize /dev/sdb
sudo lvextend -r -L +10G /dev/vgdata/documents
lvs /dev/vgdata/documents
df -hT /srv/documents
```

Shrinking and moving data

Reduction is more hazardous than extension. Never reduce an LV below the actual filesystem size. ext4 may be reduced only while unmounted, after a filesystem check; XFS cannot be shrunk in place. Filesystem-specific procedures must be completed before lvreduce, or data loss is likely.

pvmove relocates allocated extents from one PV to other free extents in the same VG. Before removing a disk, ensure sufficient space elsewhere, run pvmove, confirm no extents remain, then vgreduce. A pvmove is not a backup and should not be treated as a substitute for one.

```
sudo pvmove /dev/sdb /dev/sdc
pvs -o pv_name,pv_used,pv_free
sudo vgreduce vgdata /dev/sdb
sudo pvremove /dev/sdb
```

Inspection, persistence and recovery

Use pvs, vgs and lvs for structured summaries; vgdisplay and lvdisplay provide additional detail. lvs -a reveals hidden/internal volumes, including RAID and thin-pool components. /etc/lvm/backup and /etc/lvm/archive normally hold text metadata backups, but not file data. vgcfgbackup can save VG metadata explicitly; vgcfgrestore is a recovery operation requiring a precise understanding of device state.

A healthy-looking LV does not guarantee a healthy filesystem or a functioning backup. Monitor VG free space, filesystem utilisation and physical device errors. Use lsblk, findmnt, smartctl (where available) and kernel logs to correlate failures.

```
sudo vgcfgbackup vgdata
lvs -a -o +devices,lv_health_status
findmnt -o SOURCE,TARGET,FSTYPE,OPTIONS
journalctl -k -p warning --since today
```

Reference scope: commands and defaults vary by distribution and software version. Consult the locally installed manual pages for exact syntax and behaviour.