Logical Volume Manager (LVM)

Florent Gluck - Florent.Gluck@hesge.ch

March 9, 2022

What is Logical Volume Manager?

- LVM = storage virtualization
- LVM is a **layer of abstraction over the physical storage**
- Project started in 1998 with LVM1, followed by LVM2 in 2001 (Linux 2.6)
- LVM2 heavily uses the Linux device mapper (DM) kernel driver

MD, DM, and LVM

- **Linux MD (Multiple Device) kernel driver**
  - provides virtual devices created from physical devices
  - foundation of software RAID 0,1,4,5,6 (`mdadm` tool)
- **Linux DM (Device Mapper) kernel driver**
  - virtualizes block devices
  - maps physical block devices onto higher-level virtual block devices
  - foundation of LVM, disk encryption, file system snapshots, etc.
- **LVM2 (Logical Volume Manager)**
  - uses DM to provide generic volume management from userspace
  - can be used on top of MD or DM devices

Why LVM?

. . . because

virtualization of mass storage is mandatory in data centers to **minimize system downtime** and **increase flexibility!**

LVM benefits

- **Thin provisionning** → create filesystems larger than available physical space
- **Abstraction layer hides details about physical storage**
  → storage can be modified **unknowingly** to applications
  - transparent aggregation of multiple physical devices
  - disks can be added/replaced at runtime (hot-swapping)
- **Data can be moved/re-arranged/resized at runtime**
  → provides flexibility
- **Atomic filesystem snapshots**, regardless of the underlying physical layout → allows for consistent backups

Advanced features

Features usually required by large storage farms:

- Clustered LVM (CLVM)
- High-Availability LVM (HA-LVM)
- Mirroring

LVM terminology

Physical Volume (PV)

- Physical storage, typically hard disk, partition, or something that looks like a disk, e.g. software RAID device

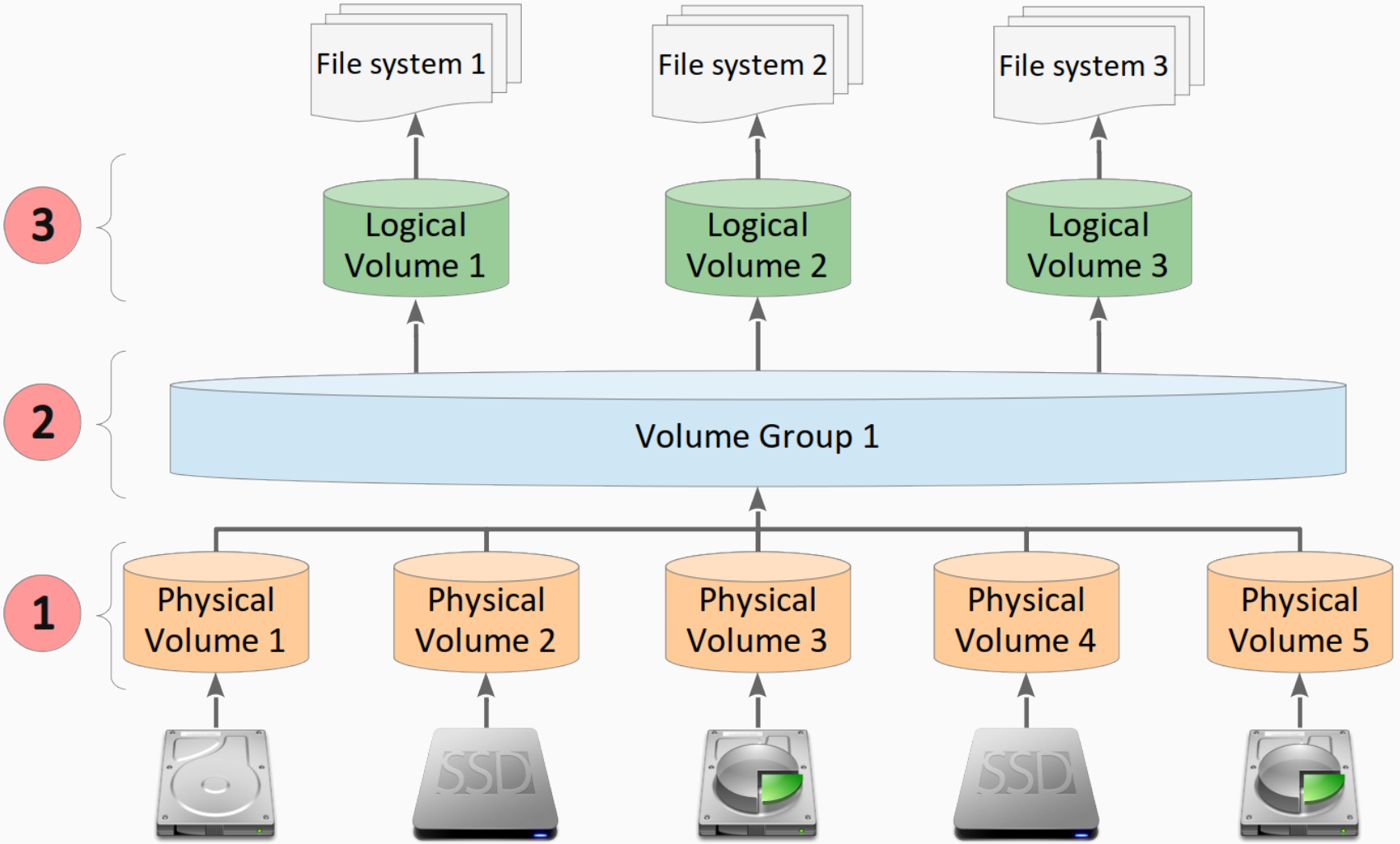
Volume Group (VG)

- A pool of physical volumes presented as one administrative unit

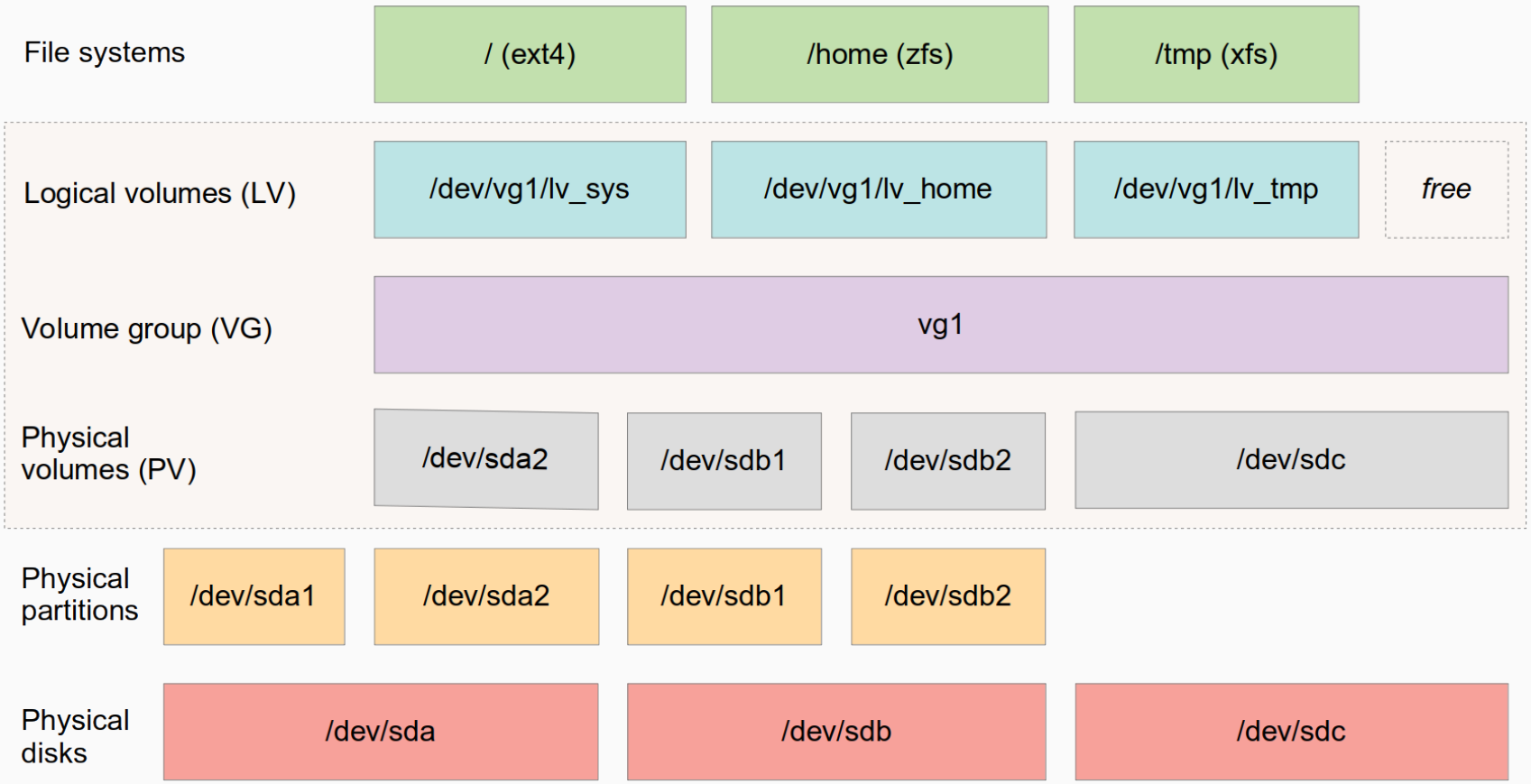
Logical Volume (LG)

- An exposed block device ($\sim$ equivalent of a disk partition)
- May be spanned, striped, mirrored, or a snapshot

LVM 3-layer model



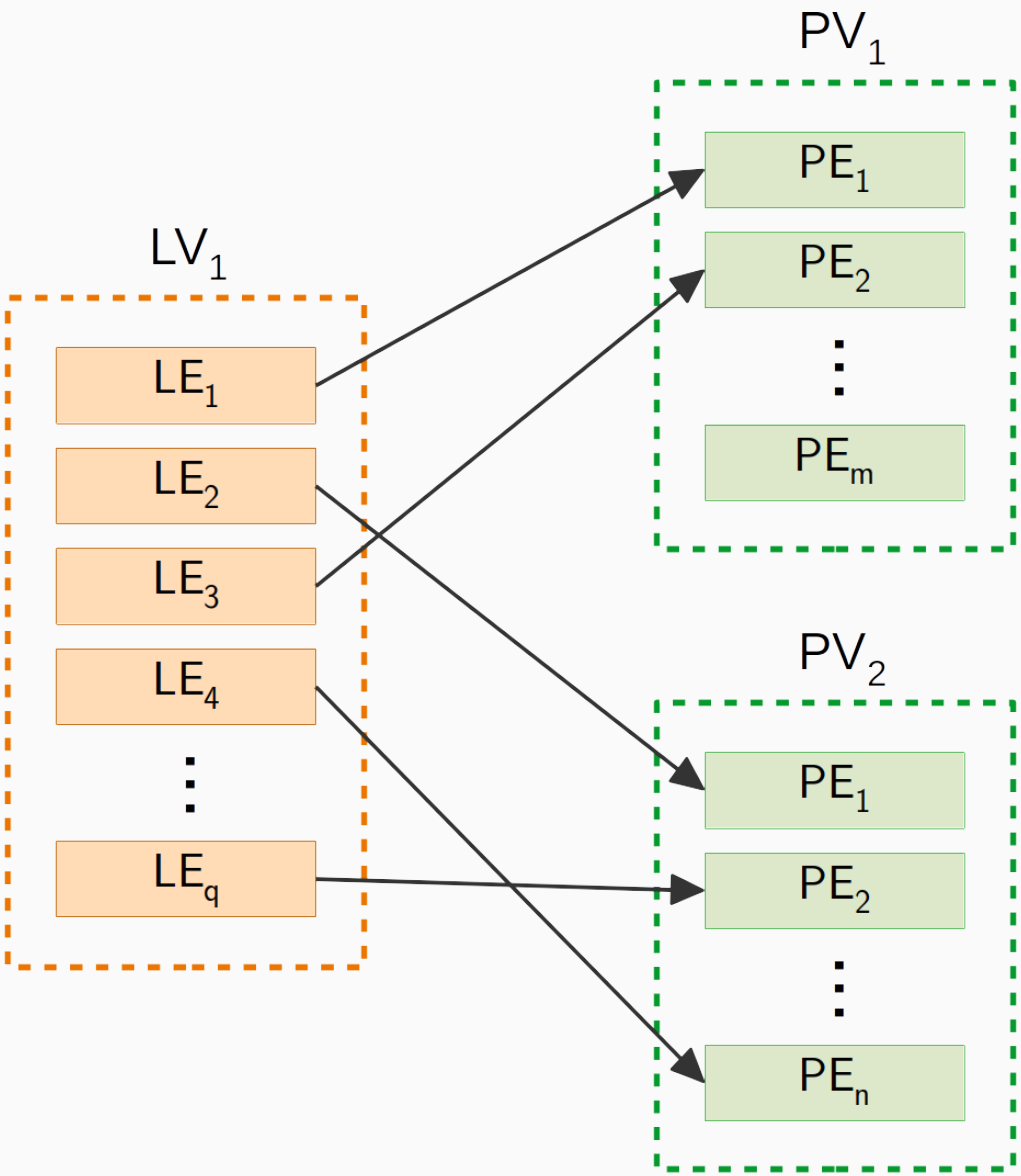
LVM example


File systems
/ (ext4)
/home (zfs)
/tmp (xfs)
Logical volumes (LV)
/dev/vg1/lv_sys
/dev/vg1/lv_home
/dev/vg1/lv_tmp
free
Volume group (VG)
vg1
Physical volumes (PV)
/dev/sda2
/dev/sdb1
/dev/sdb2
/dev/sdc
Physical partitions
/dev/sda1
/dev/sda2
/dev/sdb1
/dev/sdb2
Physical disks
/dev/sda
/dev/sdb
/dev/sdc

LVM data allocation unit

- LVM'b basic allocation unit is an **extent**
- An extent is a contiguous area of storage, represented by 2 numbers: (offset, length)
- However, LVM's extents are of unique size
  - thus, an LVM extent is simply a "large block"
- LVM manages physical extents (PE) and logical extents (LE)
  - physical volumes are divided into PE
  - volume groups are sets of PE
  - logical volumes are sets of LE
  - PE and LE have the same size
  - PE size displayed in `vgdisplay`

LE to PE mapping



LVM features

- VGs **resizable online** by absorbing new PVs or ejecting existing ones
- LVs **resizable online**
- LVs movable between PVs
- VGs can be split or merged as long as no LVs span the split
  - Useful when migrating whole LVs to or from offline storage
- **Atomic snapshots** (copy-on-write)
- **Thinly-provisioned** LVs (over-commit physical storage)
- Supports RAID 0, 1, 4, 5, 6, 10
- High availability (shared-storage cluster with shared PVs between hosts)

LVM usage

- Run `lvm`, then `help`
- Physical volume commands start with `pv*`
- Volume group commands start with `vg*`
- Logical volume commands start with `lv*`
- LVM stores its configuration files in `/etc/lvm/`

Steps to create and mount logical volumes

0. (Create partitions: `fdisk`)
1. Create physical volumes: `pvcreate`
2. Define volume groups: `vgcreate`
3. Create logical volumes: `lvcreate`
4. Create file systems in logical volumes: `mkfs.ext4`, `mkfs.xyz`
5. Mount file systems: `mount ...`

Physical volume management

- Create 3 physical volumes

```
pvcreate /dev/sda /dev/sdb /dev/sdc1
```

  This operation simply “labels” each physical disk

- List physical volumes

```
pvs
pvdisplay
pvscan
```

Volume group management

- Create `vg1` volume group over 2 physical volumes

```
vgcreate vg1 /dev/sda /dev/sdb
```

- List volume groups

```
vgs
vgdisplay
vgscan
```

Logical volume management

- Create `vol1` logical volume, of size 8G, in volume group `vg1`

```
lvcreate -n vol1 -L 8G vg1
```

vgcreate vg1 /dev/sdb /dev/sdc

```
lvcreate -n vol2 -l 50%VG vg1
```

- List volume groups

```
lvs
lvdisplay
lvscan
```

Extend storage space

- Add a new physical volume to volume groupe `vg1`

```
vgextend vg1 /dev/sdc1
```

- Set the new size of the logical volume to 42 extends

```
lvextend -l 42 /dev/vg1/vol1
```

- Extend the logical volume to 100% of the volume group

```
lvextend -l 100%VG /dev/vg1/vol1
```

- Extend the filesystem (e.g., ext4):

```
resize2fs /dev/vg1/vol1
```

Replace a physical disk

- Add the new disk to the volume group with `pvcreate` and `vgextend`
- Move extents from the old physical volume (`sdb` here) to physical volume(s) in the same volume group

```
pvmove /dev/sdb
```

- Remove the old physical volume from the volume group

```
vgreduce vg1 /dev/sdb
```

- Remove the physical volume label from the physical disk

```
pvremove /dev/sdb
```

Snapshots

- A snapshots **atomically** saves the state of a logical volume
- Snapshot performed at the block layer level $\rightarrow$ filesystem independent
- Use Copy-On-Write (COW)
  - requires a new LV to save "changes"
  - up to the user to choose the new volume size
  - size must be enough to store the changes (10% of original's LV is often recommended)
- Allows to create **atomic backups**
  - a task impossible to perform on a filesystem that doesn't support native shapshots, such as ext4

Snapshots behavior

- Let A be the original volume and S the snapshot volume of A
- S stores the "changes" after the snapshot was performed
  - changes are not the new data, but A's data before S
- When accessing S (via `mount`), we see A's original content, i.e. before S was taken
- When accessing A, we see its current content
- The atomic state of A prior to S can then be backup'ed
- A snapshot's content can be merged back to restore the state pre-snapshot
  - however: requires to umount the original volume (A) before applying the merge

Snapshots use-cases

- **Atomic backup** of a logical volume **without** taking the volume offline
- **System upgrade** (likely to succeed)
  - snapshot before the upgrade
  - if everything goes well → remove the snapshot
  - if upgrade fails → revert (merge) the snapshot
- **Discardable changes** for temporary use
  - create a snapshot of the system
  - mount the snapshot (say in `/snap`)
  - let user use `/snap`
  - when user is finished → discard the snapshot

Snapshots usage

- Create snapshot `snap` of size 10G from logical volume `vol1`

```
lvcreate -s -n snap -L 10G /dev/vg1/vol1
```

- Good idea to check how full the snapshot volume is with `lvs` (column `Data%`)
- Restore the state of `vol1` before the snapshot (`vol1` and `snap` must not be mounted)

```
umount /dev/vga1/vol1
lvconvert --merge vg1/snap
```

Resources

- Manual pages

  `man lvm`

- LVM HOWTO

  http://tldp.org/HOWTO/LVM-HOWTO/

- Red Hat Enterprise Linux 7 LVM Administrator Guide