Logical Volume Manager (LVM)

Florent Glück - florent.gluck@hesge.ch

March 31, 2025

ISC - HEPIA

What is Logical Volume Manager?

- LVM = Logical Volume Manager
- LVM is a **layer of abstraction over the physical storage** for Linux systems
- **LVM provides storage virtualization**
- Created in 1998 with LVM1, followed by LVM2 in 2001 (Linux kernel 2.6)
- LVM2 heavily uses the Linux device mapper (DM) kernel driver

Why LVM?

... because

Why LVM?

... because

... Virtualization of mass storage is mandatory in data centers to:

Why LVM?

... because

... Virtualization of mass storage is mandatory in data centers to:

- **increase flexibility**

Why LVM?

... because

... Virtualization of mass storage is mandatory in data centers to:

- **increase flexibility**
- **minimize system downtime!**

LVM features

- **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**
  - increases flexibility

LVM features

- **Atomic filesystem snapshots**
  - only way to provide **consistent backups** (unlike typical backups which are **not** consistent)
- **Block deduplication**[1]
  - maximizes storage efficiency
- Supports RAID 0, 1, 4, 5, 6, 10

[1] https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/deduplicating_and_compressing_logical_volumes_on_rhel/index

Advanced features

Features usually required by large storage farms:

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

LVM 3-layer model

Physical Volume (PV)

- Physical storage, typically a hard disk or a partition

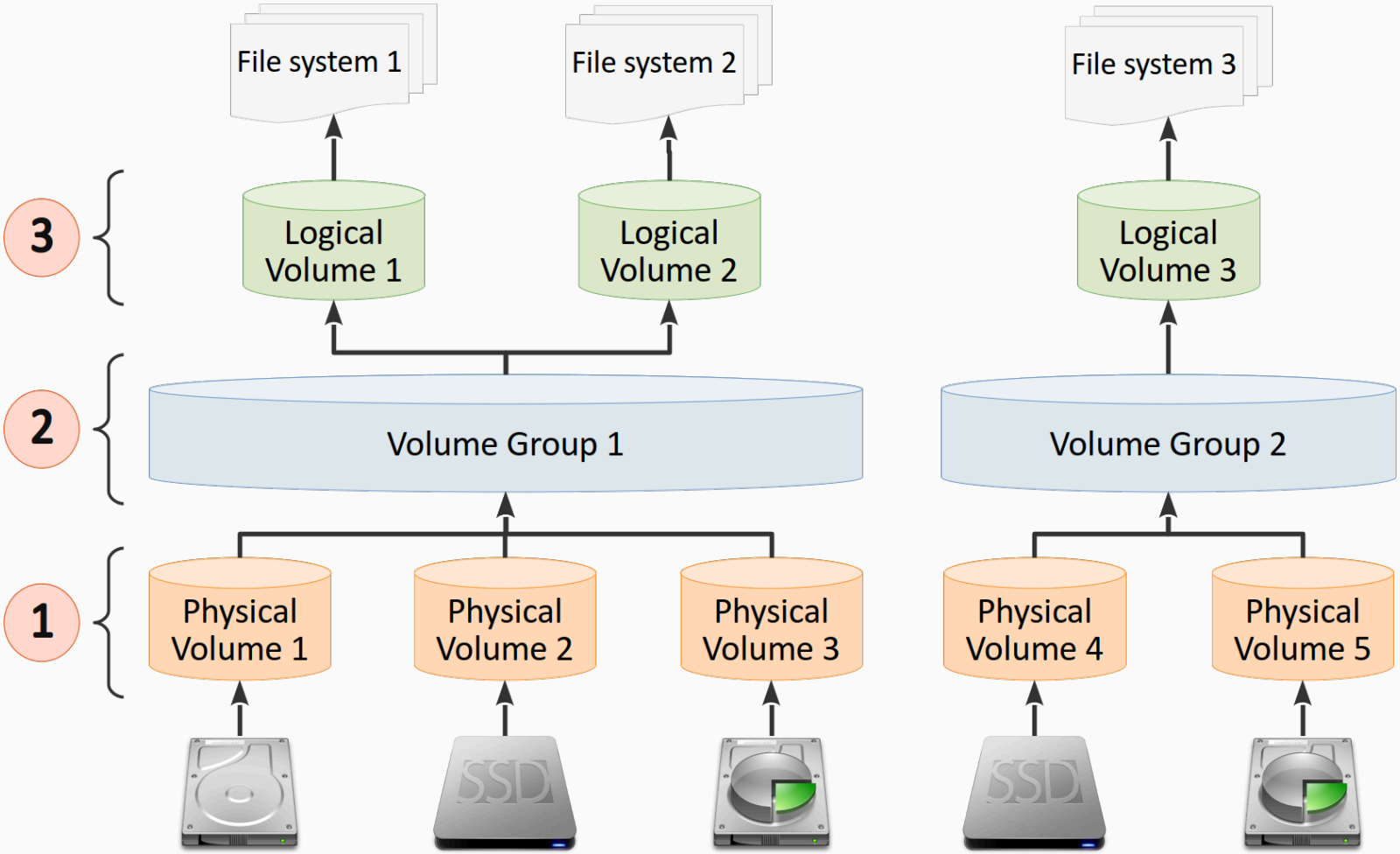
Volume Group (VG)

- A pool of physical volumes presented as one administrative unit

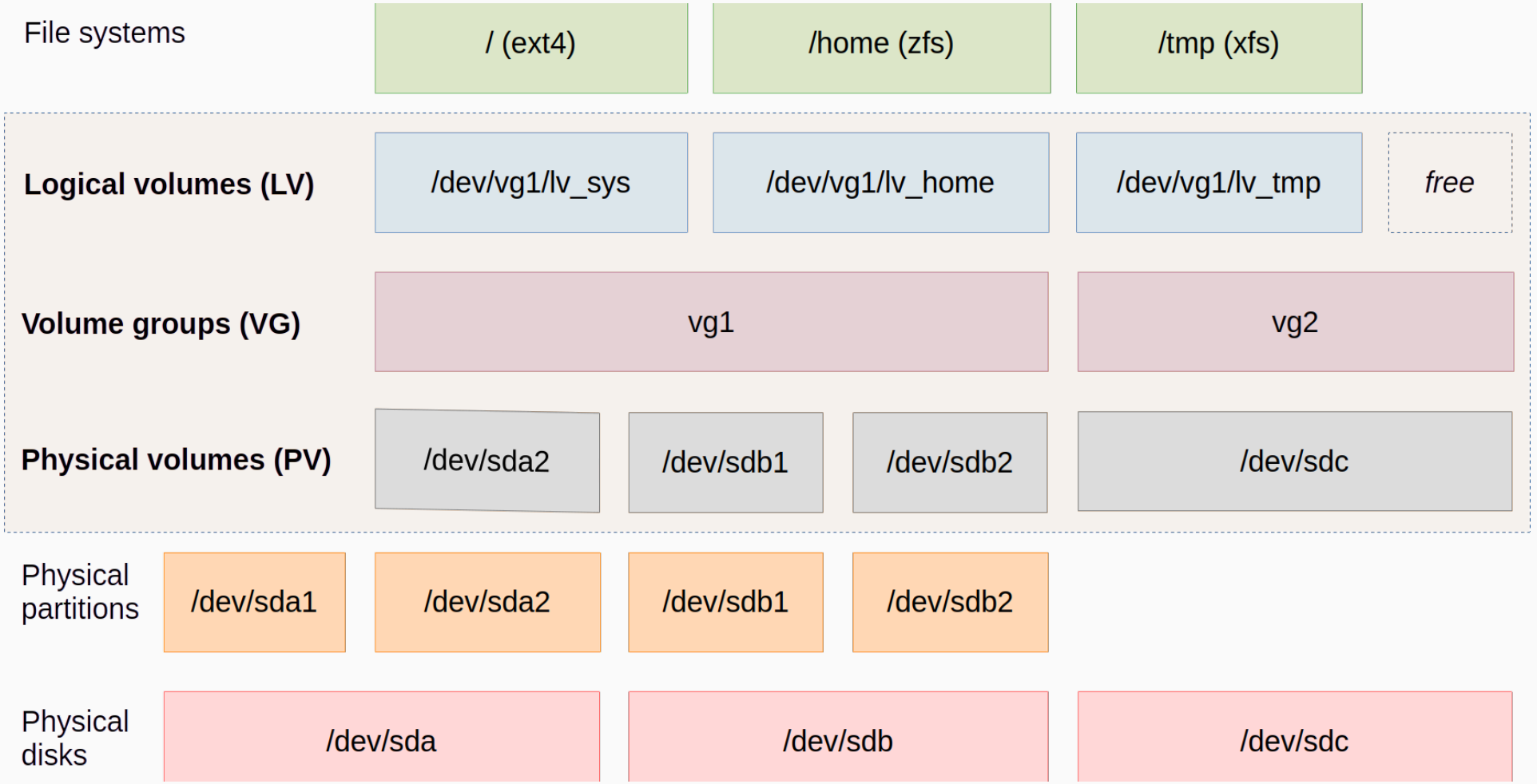
Logical Volume (LV)

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

LVM example (1/2)



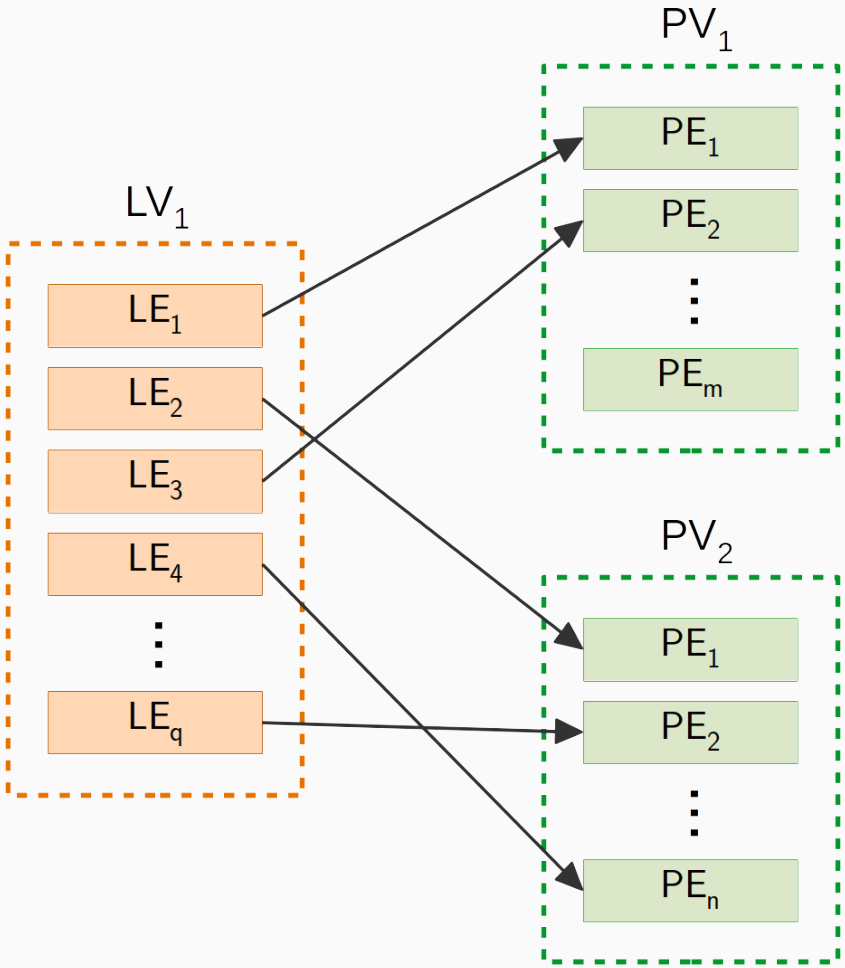
LVM example (2/2)



LVM data allocation unit

- LVM basic allocation unit is an **extent**
- Traditionally in filesystems, an extent is a contiguous area of storage represented by 2 numbers: (offset, length)
- **With LVM, an extent is simply a "large block"** (4MB by default)
- 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 `pvdisplay` and `vgdisplay`

LE to PE mapping



VG and LV features

- VGs are **resizable online**[1] by absorbing new PVs or ejecting existing ones
- LVs are **resizable online**
- LVs are 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
- **Thinly-provisioned** LVs (over-commit physical storage)

[1] **online** means while in use, by opposition to **offline**

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/`

How to create and mount a logical volume?

1. Create a physical volume: `pvcreate`
   - create it on a disk, e.g. `/dev/sda`
   - or on a partition, e.g. `/dev/sda2`
2. Create a volume group: `vgcreate`
3. Create a logical volume: `lvcreate`
4. Create a file systems in a logical volumes: `mkfs.ext4`, `mkfs.xfs`, etc.
5. Mount the file systems: `mount ...`

Physical volume management: example

- To create 3 physical volumes: on disk `sda`, disk `sdb` and first partition of disk `sdc`:

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

`pvcreate` only "labels" each physical disk

- To list physical volumes:

```
pvs
pvdisplay
pvscan
```

Volume group management: example

- To create volume group `vg1` over 2 physical volumes (`sda` and `sdb`):

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

- To list volume groups:

```
vgs
vgdisplay
vgscan
```

Logical volume management: examples

- To create logical volume `vol1` in volume group `vg1`, using 8GB of `vg1`:

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

  then create an ext4 filesystem into it:

```
mkfs.ext4 /dev/vg1/vol1
```

- To create logical volume `vol2` in volume group `vg1`, using 50% of `vg1`:

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

- To list logical volumes, use either:

```
lvs      lvdisplay      lvscan
```

Extend storage space: examples

- To add physical volume `sdc1` to volume groupe `vg1`:

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

- To set the new size of logical volume `vol1` to 42 extends:

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

- To extend logical volume `vol1` to 100% of the volume group:

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

- To extend the filesystem (e.g., ext4) to the size of logical volume:

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

How to replace a physical disk?

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

```
pvmove /dev/sdb
```

3. Remove the old physical volume `sdb` from the volume group `vg1`:

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

4. Remove label from the physical volume:

```
pvremove /dev/sdb
```

The need for snapshots

- Snapshots allow to create **atomic backups**
- Withouth LVM $\longrightarrow$ impossible to perform atomic backups on filesystems that do not support native snapshots (such as ext4[1])
- Snapshot are performed at the block layer level $\longrightarrow$ they are **filesystem independent!**

[1] More advanced filesystems such as btrfs and zfs feature snapshots

Snapshot mechanism

- A snapshots **atomically** saves the state of a logical volume
- The state is saved in a “snapshot” logical volume
  - **keeps track of changes** made to the original logical volume’s data
  - it is a frozen, **read only image** of the original logical volume
- The snapshot logical volume can be **merged back** into the original logical volume **to roll back the data**

Snapshot: under the hood

- The snapshot logical volume must be able to **store** the original blocks of **all files that have changed** during the snapshot's lifetime
- A snapshot logical volume is **frozen** at the moment of creation, but the original logical volume can be used and will be changed
- The snapshot logical volume must have **enough space to keep the changes** on the original logical volume during the lifetime of the snapshot

Snapshot behavior: example

- Let A be the original logical volume and S the snapshot of A
- S records the changes to A (since S was created)
- When accessing **S** (via `mount`), we see A's **original content**, i.e. before S was created
  - the **atomic state of A prior to S can then be backup'ed!**
- When accessing **A**, we see its **current content**
- 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

Snapshot use-cases

- **Atomic backup** of a logical volume **without** taking the volume offline
- **System upgrade** (likely to succeed)
  - create snapshot before the upgrade
  - if everything goes well $\longrightarrow$ remove the snapshot
  - if upgrade fails $\longrightarrow$ 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 $\longrightarrow$ discard the snapshot

Snapshot usage

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

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

  - means the snapshot volume can store up to 10G of changes

- Good idea to check how full the snapshot volume is with `lvs` (column `Data%`)

- Restore the state of `vol1` before the snapshot

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

  - ⚠️ `vol1` and `snap` **must not** be mounted!

Resources

- Manual pages
  ```
  man lvm
  ```
- LVM HOWTO
  https://tldp.org/HOWTO/LVM-HOWTO/
- Beginners guide to how LVM works in Linux (architecture)
  https://www.golinuxcloud.com/overview-lvm-in-linux/