LVM Best Practices

Efstathios Agrapidis

stathisagrapidis@gmail.com

IRC: efagra

What is LVM?

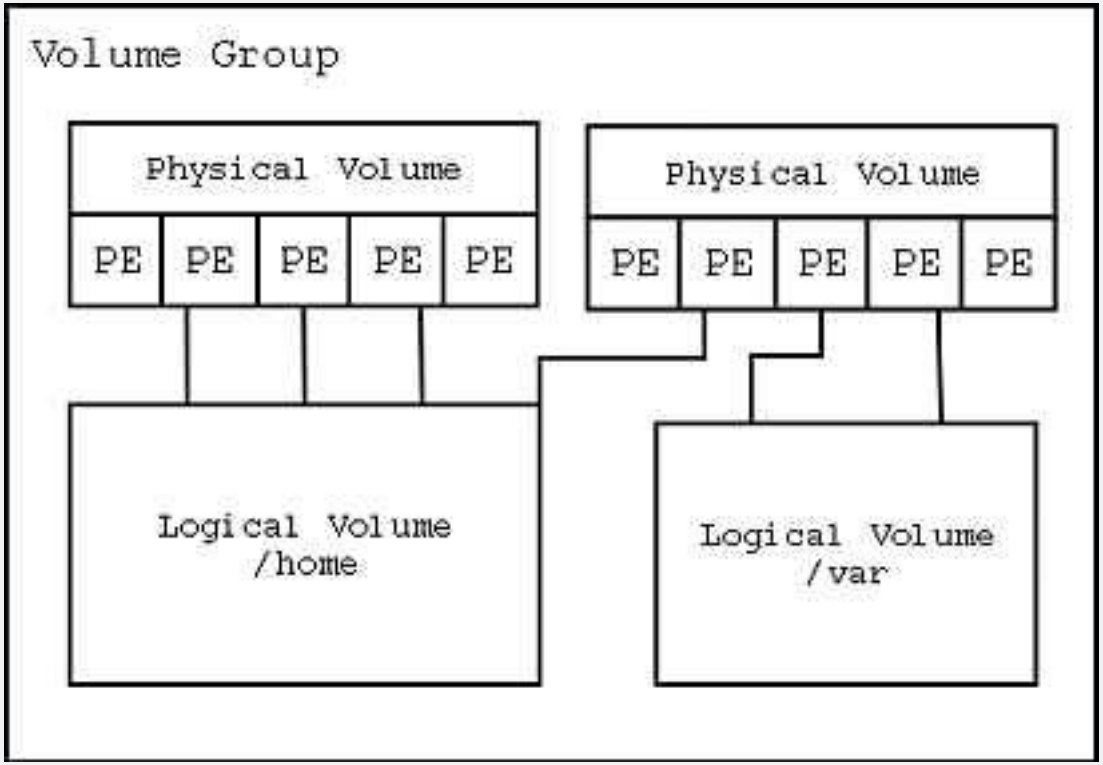
- Linux Volume Management
- Another way of partitioning the drive
- higher-level view of the disk storage
- Flexible partitioning schemes
- Live resizing of filesystems
- large installations containing many disks

Benefits on a Small System

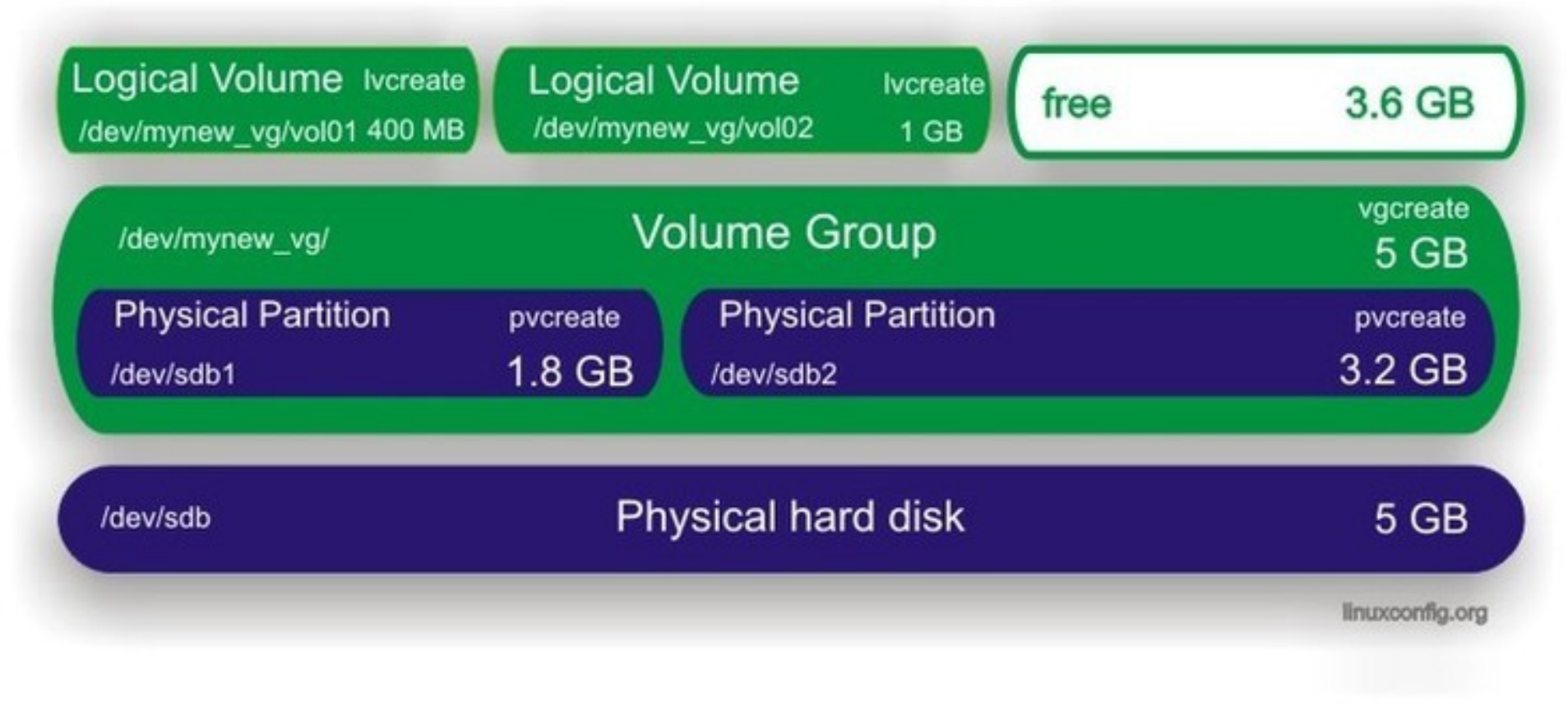
- How much space is going to be needed for system files and user files?
- Live resizing of filesystems
- Scenarios
  - Shrink a filesystem – Grow another one
  - Allocate minimum space – Expand acording to the live needs

Anatomy of LVM

- volume group (VG)
- physical volume (PV)
- logical volume (LV)
- physical extent (PE)
- logical extent (LE)
- Snapshots



An LVM example System



More details

- The Volume Group is the highest level abstraction used within the LVM. It gathers together a collection of Logical Volumes and Physical Volumes into one administrative unit.
- A physical volume is typically a hard disk, though it may well just be a device that 'looks' like a hard disk (eg. a software raid device).
- The equivalent of a disk partition in a non-LVM system. The LV is visible as a standard block device; as such the LV can contain a file system (eg. /home).
- Each physical volume is divided chunks of data, known as physical extents, these extents have the same size as the logical extents for the volume group.

More details (continue)

- Each logical volume is split into chunks of data, known as logical extents. The extent size is the same for all logical volumes in the volume group.
- mapping modes (linear/striped) – The general strategies for mapping logical extents onto physical extents:
  - Linear mapping will assign a range of PE's to an area of an LV in order eg., LE 1 - 99 map to PV1 and LE 100 - 347 map onto PV2.
  - Striped mapping will interleave the chunks of the logical extents across a number of physical volumes.
- Snapshot is an exact copy of a logical volume, frozen at some point in time. Typically this would be used when some batch processing, a backup for instance, needs to be performed on the logical volume, but you don't want to halt a live system that is changing the data.

Commands

- pvcreate /dev/sdb1
- vgcreate my_volume_group /dev/sda1 /dev/sdb1
- vgchange -a y my_volume_group
- vgchange -a n my_volume_group
- vgremove my_volume_group
- vgextend my_volume_group /dev/hdc1
- pvdisplay /dev/sda1
- pvmove /dev/sdb
- vgreduce my_volume_group /dev/sda1

Commands (continue)

- lvcreate -L1500 -ntestlv testvg
- umount /dev/myvg/homevol
- lvremove /dev/myvg/homevol
- lvextend -L12G /dev/myvg/homevol
- lvextend -L+1G /dev/myvg/homevol
- pvmove /dev/sdb /dev/sdc
- vgreduce testvg /dev/sdb

LVM Host System Setup

Multiple distro Installations Example

Snapshot Case

- we can back up a volume without having to worry about data being changed while the backup is going on
- lvcreate -L592M -s -n dbbackup /dev/testvg/databases
- mkdir /mnt/ops/dbbackup
- mount /dev/testvg/dbbackup /mnt/ops/dbbackup
- tar -cf /pathtofile.tar /mnt/ops/dbbackup
- umount /mnt/testvg/dbbackup
- lvremove /dev/testvg/dbbackup

Snapshot Live Example

Destruction Case Scenario

Recover a Raid 1 System

- mdadm --manage /dev/md0 --fail /dev/sdb1
- mdadm --manage /dev/md0 --remove /dev/sdb1
- mdadm --manage /dev/md0 --add /dev/sdb1

Join us on:
www.opensuse.org


openSUSE