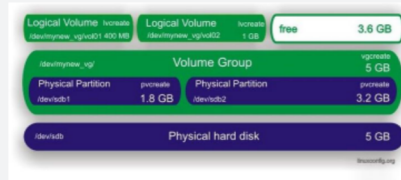


LVM - schema



Example of an LVM layout.

LVM - terminology

PE - physical extents	The smallest unit of disk space that can be managed by LVM. It's set when you create a volume group. It determines the "granularity" of disk space allocation. All physical volumes in a volume group will use the same extent size.
LE - logical extents	The same as PE but for Logical Volumes
PV - physical volume	Disks, partitions, RAID volumes
VG - volume group	Group of one or more physical volumes
LV - logical volume	Entity that contains information, store on physical volumes, grouped within a volume group

LVM - commands and components

lvm	pseudo shell for LVM
lvmconfig	prints LVM configuration from /etc/lvm/lvm.conf

LVM - commands and components (cont)

lvmdiskscan	prints devices that could be used as physical volumes
lvmdump	dumps LVM configuration to tar file
lvmtools	caching service for LVM metadata

LVM - important files

/etc/lvm/	LVM configuration directory
/etc/lvm/lvm.conf	LVM configuration file
/etc/lvm/archive/	Backup and archive of LVM configuration
/etc/lvm/profile	A set of selected customizable configuration settings that can be used to achieve certain characteristics in various environments or uses. Normally, the name of the profile should reflect that environment or use. An LVM profile overrides existing configuration.

LVM - Physical Volume

pvcreate	create PV
pvchange	change attributes of PV
pvs pvdisplay	display information about PV
pvmove	moves extents from one PV to another
pvremove	removes a PV
pvresize	resizes a PV
pvscan	scans for changes in PV configuration and size

LVM - Volume Group

vgs vgdisplay	displays VG information
vgmknodes	create the special files for VG devices in /dev
vgck	check VG consistency
vgcfsb-backup	make a backup of VG metadata
vgcfsr-restore	restore VG metadata from backup
vgrename	rename a VG
vgsplit	move PVs to a new or existing VG
vgconvert	change VG metadata format

LVM - Volume Group (cont)

vgimport	register exported VG in the system
vgexport	unregister VG from the system
vgimportclone	import a VG from cloned PVs
vgcreate	create VG from one or more PVs
vgremove	remove VG
vgreduce	remove PVs from VG
vgchange	change VG attributes
vgextend	extend VG with one or more PVs
vgmerge	merge VGs
vgscan	scan VGs for changes in metadata and size

LVM - Logical Volume

lvs lvdisplay	print LV metadata
lvchange	change attributes of LV
lvcreate	create a new LV
lvextend	extend size of LV
lvreduce	reduce size of LV (offline!)
lvresize	change size of LV

LVM - Logical Volume (cont)	LVM - create LV (cont)	LVM - Snapshots (cont)	Useful storage commands (Bonus) (cont)
lvrename rename a LV lvconvert change layout of LVM lvscan scan LVs for metadata changes and size lvremove remove LV	lvcreate -n data_lv -l 100%FREE data_vg To create an LV you need a volume group.	modify snapshot_autoextend_threshold and snapshot_autoextend_percent in <code>lvm.conf</code> Enable automatic extension of snapshots by adjusting the values	mount <PATH_TO_LV> <MOUNTPOINT> Mount a filesystem in a given location mount -a Mount all filesystems listed in <code>/etc/fstab</code>
LVM - create PV	LVM - Snapshots	Snapshots contain differences from the point a given LV has been created - not the real data. If the size of a snapshot is exceeded it becomes useless and there is no way to restore data from it. Old state of an LV can be restored from a snapshot, but please remember to unmount the filesystem first.	Useful in many situations when you need to manipulate storage devices and layout.
pvcreate /dev/<disk_device> create a new PV from device To create a PV you need a usable storage device. Try <code>lvmdiskscan</code> command.	lvcreate -L <SIZE> -s -n <NAME_OF_SNAPSHOT> <PATH_TO_LV> Create a snapshot of an LV lvcreate -L 4G -s -n data_lv-snapshot /dev/mapper/data--vg--data--lv lvremove <PATH_TO_SNAPSHOT> Remove snapshot lvremove /dev/mapper/data--vg--data--lv	Useful storage commands (Bonus) lsblk -o name,mountpoint,label,size,uuid List block devices with useful information multipathd show maps status Print path status of multipath devices lvs -o +devices Print LVs, VGs and device paths vgs -o +lv_size,-lv_name Print information about LVs, VGs, sizes and attributes mkfs.ext4 <FS> Format a filesystem for EXT4 mkfs.xfs <FS> Format a filesystem for XFS	LVM - extend PV rescan-scsi-bus.sh echo 1 > /sys/block/<DEVICE>/device/rescan Rescan underlying storage to detect change in disk size partprobe Detect changes in partition size (if necessary) pvresize <PV> Resize PV to the maximum possible size of partition or disk/LUN pvresize /dev/sdb To resize a PV you'll need to know the underlying storage and know how to rescan its size. All steps can be performed online.
LVM - create VG	LVM - extend LV		
vgcreate <VG> <PV> create a volume groups that consists of one or more PVs vgcreate data_vg /dev/sdb /dev/sdc To create a VG you need one or more PVs.	lvextend -L <SIZE> <PATH_TO_LV> Extend snapshot lvextend -L +10G /dev/mapper/data--vg--data--lv lvconvert --merge <PATH_TO_SNAPSHOT> Restore data from snapshot lvconvert --merge /dev/mapper/data--vg--data--lv		
LVM - create LV			
lvcreate -n <LV_NAME> -L <SIZE> <VG> create a LV of a given size in a VG lvcreate -n data_lv -L 100G data_vg lvcreate -n <LV_NAME> -l 100%FREE <VG> create a LV that fills 100% of free space in VG			

LVM - Extend VG

vgextend Extend existing
<VG> VG with one or
<PV> more PVs
<PV>

```
vgextend data_vg /dev/sdb
/dev/sdc
```

To extend a VG you need a VG and one or more PVs that are not assigned to any PVs yet.

LVM - extend LV

lvextend -L extend logical
<SIZE> volume by given
/dev/m- size (add 10GB to
apper<- existing size)
PATH_T-
O_LV>

```
lvextend -L +10G /dev/mapper/-
data--vg_data--lv
```

lvextend -r extend logical
-L <SIZE> volume and its
/dev/m- filesystem by
apper<- given size (add
PATH_T- 10GB to existing
O_LV> size)

```
lvextend -r -L +10G /dev/mapp-
er/data--vg_data--lv
```

To extend an LV you need its path. Use **df** command.
In most cases you'll also want to extend the underlying filesystem, so use **-r** option to do it.

Extend filesystem (bonus)

umount <FS> Unmount
filesystem

```
umount /data
```

e2fsck -f Check and fix
<PATH_TO_- potential errors
LV>

Extend filesystem (bonus) (cont)

```
e2fsck -f /dev/mapper/data--vg_-
data--lv
```

resize2fs <PA- Reduce size
TH_TO_LV> of filesystem
<SIZE>

```
resize2fs /dev/mapper/data--v-
g_data--lv 10G
```

lvreduce -L <SI- Reduce size
ZE> <PATH_- of LV
TO_LV>

```
lvreduce -L 10G /dev/mapper/-
data--vg_data--lv
```

e2fsck -f <PA- Check for
TH_TO_LV> errors once
more

```
e2fsck -f /dev/mapper/data--vg_-
data--lv
```

mount <FS> Mount
filesystem

```
mount /data
```

Remember to make sure the size of both: FS (filesystem) and LV (logical volume) are the same and there are no errors. Reducing the size can ONLY be done OFFLINE. The filesystem must be unmounted. Not all filesystems can be reduced at all.

