

LVM Cheatsheet v1.4

The following is a command summary for the basic operation of Linux LVM including Physical Disks (PVs), Volume Groups (VGs) and Logical Volumes (LVs). It is assumed that the reader is familiar with the concepts of such a system.

Command Cheatsheet

Physical Volume Tools

pvdisplay [<code></dev/XXX></code>]	- Display all PV details, or optionally by device
pvscan	- Scan all disks for physical volumes
pvs	- Terse display of PV information
pvcreate <code>/dev/hdb1</code>	- Initialise a disk or partition for use by LVM
pvmove <code>/dev/sdb1</code>	- Move data off PV (Eg. prior to removal).
pvmove <code>/dev/sdb1</code> <code>/dev/sdf1</code>	- Move data off PV onto a new one.
pvremove <code>/dev/sdb1</code>	- Remove configuration for unused PV
pvck <code>/dev/sdb1</code>	- Check physical volume metadata
pvresize <code>/dev/sdb1</code>	- Resize PV to match changes in underlying storage (some caveats, read docs first).

Volume Group Tools

vgdisplay [<code><vgname></code>]	- Display all VG details, or optionally by VG
vgscan	- Scan all disks for VGs and rebuild caches
vgs	- Display terse information on volume groups.
vgchange <code><options></code> <code><vgname></code>	- Change attributes of a volume group - Eg. <code>vgchange -a y/n</code> (activate/deactivate)
vgextend <code><vgname></code> <code>/dev/hdc1</code>	- Add PVs to a volume group
vgreduce <code><vgname></code> <code>/dev/hda1</code>	- Remove PVs from a volume group
vgcreate <code><newvg></code> <code>/dev/sdc</code> <code>/dev/sdd</code>	- Create new volume group from SDC & SDD

vgexport <vgname>	- Prepare VG for move to another machine.
vgimport <vgname>	- Import previously exported VG
vgmerge <dest_vgname> <source_vgname>	- Merge two existing volume groups - (Note <source_vgname> must be inactive)
vgsplit {Too many options to list}	- Split a VG in two, by moving one or more PV - to a new VG (Study docs before proceeding!)
vgremove <vgname>	- Remove unused VGs from the system
vgck <vgname>	- Check volume group metadata
vgconvert <vgname>	- Convert metadata type (Eg. LVM1 to LVM2) - Note: You should never need to do this!

Logical Volume Tools

lvdisplay [<lvname>]	- Display all LV details, or optionally by LV
lvscan	- Scan (all disks) for Logical Volumes
lvs	- Display terse information on logical volumes
lvchange	- Change attributes of a logical volume - Eg. lvchange -a y/n (activate/deactivate)
lvcreate -l<extents> -n<newlv> <vgname> lvcreate -L<size> -n<newlv> <vgname>	- Create LV 'newlv' in VG 'vgname' - Create LV 'newlv' in VG 'vgname' - Note: <size> in K, M or G format (-L1.8G)
lvextend -L12G myvol lvextend -L+1G myvol	- Extend LV to 12Gig (-l to specify in extents) - Extend LV by 1G (-l to specify in extents)
lvreduce -L12G myvol lvreduce -L+1G myvol	- Reduce LV to 12Gig (-l to specify in extents) - Reduce LV by 1G (-l to specify in extents) - Use Extreme Care, Eg. -- resizefs option
lvremove <lvname>	- Remove logical volume
lvrename <vgname> <lvname> <lvname>	- Rename logical volume
lvresize <lvname>	- Resize LV (Note: many options, study docs)
lvconvert [options] <lvname>	- Change logical volume type - Eg. From Linear to Mirrored volume - Note: Study documentation before proceeding!

Logical Volume Examples

- lvcreate -L1500M -ntestlv testvg** - Create a 1500MB linear LV named 'testlv'
- lvcreate -i2 -I4 -l100 -ntestlv testvg** - Create a LV 100 extents in size with
- 2 stripes and stripe size 4 KB
- lvcreate -L1500M -ntestlv testvg [/dev/sdg](#)** - An LV can be allocated from a specific
- PV or PVs in the specified VG if needed

LV's can be referred to by their /dev/ path name, as well as by name. (Eg. /dev/mapper/<lvname>)

To create an LV that uses the entire VG, use something like **vgdisplay testvg | grep "Total PE"**, then take the "Total PE" value and proceed to populate a command like the following:-
lvcreate -l <Total PE> testvg -n mylv.

Remember to first unmount before you can remove an LV.

For Snapshot - Size is the max amount of change which we reserve space for.

lvcreate -L592M -s -n dbbackup databases

Always "lvremove" after use to recover space.

Worked Example

Create Volume

First Initialise the volumes to use.

```
# pvcreate /dev/sdb1  
# pvcreate /dev/sdc1
```

Create volume group with these physical volumes in.

```
# vgcreate Test_Vol /dev/sdb1/dev/sdc1
```

Activate Volume Group in Kernel

```
# vgchange -a y Test_Vol
```

Create Logical Volume inside the new Volume Group

```
# lvcreate -L500 -nTest_Disk Test_Vol
```

Write filesystem to Logical Volume

```
# mkfs.ext4 /dev/mapper/Test_Vol-Test_Disk
```

Mount filesystem so that it becomes accessible (or place in fstab)

```
# mount /dev/mapper/Test_Vol-Test_Disk /mnt
```

Grow Logical Volume by 1 gigabyte

```
# lvextend -L+1G /dev/mapper/Test_Vol-Test_Disk
```

Extend Filesystem into new, larger LV

```
# resize2fs /dev/mapper/Test_Vol-Test_Disk
```

Show that the new space is available

```
# df -h
```

Ensure no data is stored on /dev/sdb1

```
# pvmove /dev/sdb1
```

Remove /dev/sdb1 from volume

```
# vgreduce Test_Vol /dev/sdb1
```

After testing is complete, we may wish to remove volume

Deactivate and remove volume group

```
# vgchange -a n Test_Vol  
# vgremove Test_Vol
```

Additional Thoughts

LVM and TRIM on SSD

Using TRIM via LVM on striped or mirrored LV's does work, but at the time of writing, it does not work if LVM is nested on top of MDRAID, so please bear this in mind when reading the example below..

```
# pvcreate --metadatasize 250k /dev/sdg2
```

```
# pvcreate --metadatasize 250k /dev/sdh2
```

```
# vgcreate vgssd /dev/sdg2 /dev/sdh2
```

```
# vgsdisplay vgssd | grep Total  
Total PE 42806
```

(you'll need to adjust --extents according to the size of your device, this one is for 90GB)

```
# lvcreate -i2 -l128 -n lvssd vgssd --extents 42806
```

```
# mkfs.ext4 -v -m 0.1 -E stride=32,stripe-width=64 /dev/vgssd/lvssd
```

Create EXT4 filesystem:

```
# mkfs.ext4 -v -m 0.1 -E stride=32,stripe-width=64 /dev/vgssd/lvssd
```

Turn off periodic fsck:

```
# tune2fs -c 0 /dev/vgssd/lvssd
```

Set elevator to 'deadline', and increase queue size. I put this into my [/etc/rc.local](#):-

```
echo deadline >/sys/block/sdg/queue/scheduler  
echo deadline >/sys/block/sdh/queue/scheduler  
find /sys/devices -name nr_requests -exec sh -c 'echo 4096 > {}' \;
```

Add noatime and discard (required for TRIM support) mount options to [/etc/fstab](#)