**Logical Volume Manager**

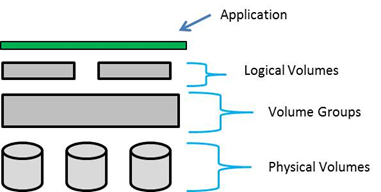
LVM is a logical volume manager for the Linux kernel that manages disk drives and similar mass-storage devices. Using a logical volume manager will give more flexibility to increase/reduce the file system in a more effective way and no loss of data.

### Advantages

LVM gives you more flexibility than just using normal hard drive partitions:

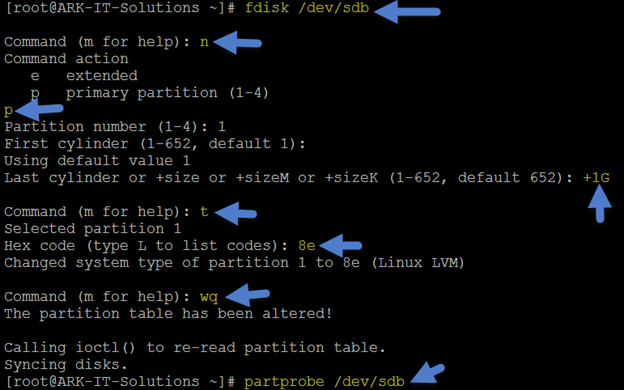
* Use any number of disks as one big disk.
* Have logical volumes stretched over several disks?
* Create small logical volumes and resize them "dynamically" as they get filled up.
* Resize logical volumes regardless of their order on disk. It does not depend on the position of the LV within VG, there is no need to ensure surrounding available space.
* Resize/create/delete logical and physical volumes online. File systems on them still need to be resized, but some (such as ext4) support online resizing.
* Online/live migration of LV being used by services to different disks without having to restart services.
* Snapshots allow you to back up a frozen copy of the file system while keeping service downtime to a minimum.

Support for various device-mapper targets, including transparent file system encryption and caching of frequently used data.



To make LVM’s we will first create physical volumes, we will combine all the PV’s into the volume group and top of the VG’s we will create LVM’s.

Let’s create Partitions



[root@ARK-IT-Solutions ~]# fdisk /dev/sdb

Command (m for help): n

Command action

e extended

p primary partition (1-4)

p

Partition number (1-4): 1

First cylinder (1-652, default 1):

Using default value 1

Last cylinder or +size or +sizeM or +sizeK (1-652, default 652): +1G

Command (m for help): t

Selected partition 1

Hex code (type L to list codes): 8e

Changed system type of partition 1 to 8e (Linux LVM)

Command (m for help): wq

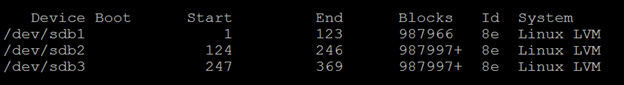
The partition table has been altered!

Calling ioctl() to re-read partition table.

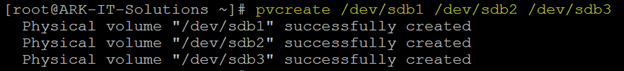
Syncing disks.

[root@ARK-IT-Solutions ~]# partprobe /dev/sdb

Repeat the above step for remain partition creations.



Creating Physical volumes

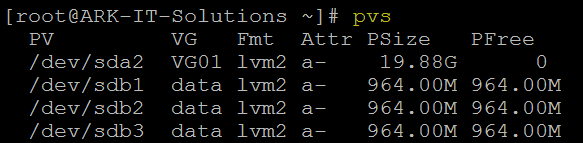


# pvcreate /dev/sdb1 /dev/sdb2 /dev/sdb3 – to create PV’s

# pvscan - to see if any PV’s are there

# pvdisplay - to see PV properties, attributes of a physical volume

# pvs - to see the PV’s information, produces formatted output about PV’s

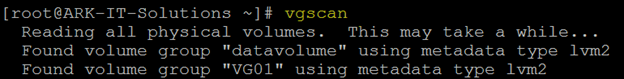


# vgcreate /dev/sdb1 /dev/sdb2 /dev/sdb3

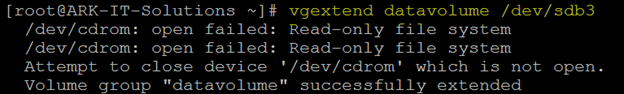


# vgdisplay - It will display all VG properties

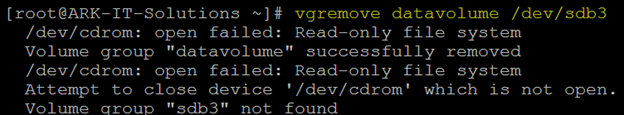
# vgscan - It will scan for all existing volume groups and rebuild caches



# vgextend <pv name> - allows you to add one or more initialized physical volumes to an existing volume group to extend it in size.



# vgremove <vg name> - to delete VG

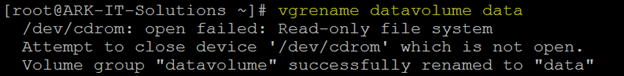


# vgreduce <vg name> <pv name> - to remove PV from VG



Note: Volume Group reducing will lead to data loss, we have to take a backup of complete VG then do the above step. Do not practice above step in any production environment.

#vgrename <old vg name> <new vg name> - to rename VG name



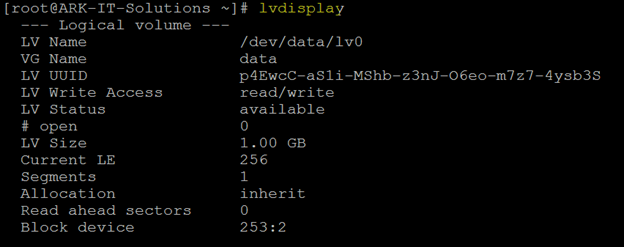
#vgmerge <vg1> <vg2> - To merge two VG’s as one group



# lvcreate –n <lv name> -L +<size> <vg name> - create a logical volume in an existing VG



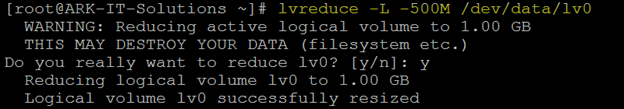
# lvdisplay - allows you to see the attributes of a logical volume like size, read/write status, snapshot information, etc.



# lvextend –L +500M <lv path> - to extend the LV space



# lvreduce –L -500M <lv path> - to reduce the LV space



Note: reducing the logical volume space will lead to data loss, before reducing the lv space we have to take lv snapshot to restore back to the same state. Do not practice this command until unless it’s required.