Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Linux LVM Components:-

1.Physical Volumes(PV)
2.Volume Groups(VG)
3.Logical Volumes(LV)

1.Physical Volume:- The physical volume is a block device such as partition or whole device(disk). To use the device as an LVM device, you must initialize the device as physical volume using **pvcreate** command. Initializing a block device as a PV, places a label near the start of the device. The LVM label is placed in the 2nd 512 bytes sector. The LVM label provides the correct identification and device order. The label is persistent and will not change across reboots.

The **LVM label** identifies the device as LVM physical volume with Unique Identifier called UUID. The LVM label also stores the size of the block device in bytes and it records where metadata will be stored on the device.

LVM meta data contains information about VG's on your system and all the PVs in the same VG will have identical copies of meta data.. By default PV will have one copy of metadata. It also has two identical copies on each PV. The first copy is stored at the start of the device immediate to the label. If there is a second copy, it is placed at the end of the device. If there is any issue with the 1st copy, such as accidental overwriting the area at the beginning, and a second copy of the metadata at the end of the device will allow you recover the Meta data.

UUID refers to Universal Unique Identifier.

2. Volume Group: - One or more PVs are grouped into volume group. This will be a pool of disk space. Within a VG, each PV is divided into fixed size units called Extents. An Extent is a smallest unit space that can be allocated within a PV. Extents are referred as Physical Extent (PE).

3. Logical Volume:- In LVM a volume group is divided into logical volumes. A logical volume is made of up of logical extents (LE) which intern mapped with PE's of PV. LE size is same as PE size. LVM creates a layer of abstraction over a physical storage which allows you to create a logical volume. Logical volume provides below advantages over using physical storage:

1) Flexible capacity
2) Resizable storage Pool
3) Online data relocation
4) Convenient device mapping
5) Device Striping
6) Mirroring Volumes
7) Volume Snapshot etc.,

There are **three** types of logical volumes. They are:

a. Linear Logical Volumes
b. Striped Logical Volumes
c. Mirrored Logical Volumes

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

a)Linear Logical Volume:- Linear LV aggregates space from one or more physical volumes into one volume. For example, a VG with 3 PVs, each 3GB can concatenate to 30GB and an LV with size 90GB can be created.

b)Striped LV:- Striping enhances performance by chopping data into small pieces(stipe) and each piece of data is writing into different PVs in round robin fashion. With striping the input and output operations on PVs are done parallel and hence its performance is more.

b)Mirrored Logical volume:- Mirrored LV maintains identical copies on different devices(PVs), where the data is written in one device is also written in the second device as well. The approach provides fault tolerance if any one device fails. When one leg of mirror fails the LV become a linear LV and can still accessible. You can have multiple mirrors for single LV.

When LVM mirror is used, LVM divides the device being copied into region, that they are typically 512KB in size. LVM maintains a small log where it uses to keep track of which regions are in sync with mirrors. This log can be kept on a disk (persistent across reboots). From RHEL 6.3 onwards, LVM supports RAID logical volumes.

Creating Linear Logical Volume:-

lvcreate -L SizeOfLV VgName
or
lvcreate -L SizeOfLV -n Lvname Vgname
or
lvcreate -L sizeofLV -n LvName Vgname PVname1 pvname2 ...
or
lvcreate -l NumberofLEs -n Lvname Vgname pvname1 pvname2 ...
or
lvccreate -l PerCent%VG -n LVname VGname
ex:-
lvcreate -n myloglv -l 50%VG logsvg
In the above example, 50% of VG space is allocated to myloglv.
or
lvcreate -l Percent%FREE -n lvname vgname
ex:-lvcreate -n mydatalv01 -l 100%FREE datavg
The abvoe example will create an LV with all left over free space in the VG.
or
Specify the range as below example:
lvcreate -l 100 -n dlv01 datavg /dev/sdb:0-49 /dev/sdc:51-100
The above example creates a linear logical volume 0-49 PEs from /dev/sdb and continuous by 51-100 PE's from /dev/sdc

Change Pv allocatable attribute:-

pvchage -x n /dev/sdc --> set allocatable attribute to No. You can use any PEs from this PV to create or extend logical volume.
pvchage -x y /dev/sdc --> set allocatable attribute to Yes. This is default attribute and allows PEs that can be used to create or extend LV.

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

VgActivate:- When you create a VG, by default, it is activated ie all LVs in that VG will be available to use. However, you can make VG to inactivate state so that all LVs in that VG are not available to use.

vgchange -a y Vgname : To activate VG (ie make all LVs in the VG available)
vgchange -a n vgname : To inactivate VG (ie make all LVs in the VG are not available to use).

Note:- If Vg contains any LVs that are opened (ie mounted), you cannot inactivate the VG. You need close all LVs (ie umount all FS in that VG) to inactivate the VG.

Renaming the Volumegroup:-

Syntax:- vgrename OldVGName NewVGname
or
vgrename /dev/OldVGName /dev/NewVGName

Setting up Max LVs for a VG:-

vgchange -l MaxNo.OfLvs VGname
ex:-
vgchange -l 128 datavg.
The above example allows datavg to have maximum 128 Logical volumes.

Setting up Max PVs for a VG:-

vgchange -p MaxNo.OfPVs VGName
ex:-
vgchange -p 16 datavg
The above command will allow maximum 16 PVs in the vg called datavg.'
Note: for LVM version 2, by default the max value 0, which means, there is restriction on maximum limit. However, for LVM version 1, you can have maximum 255 PVs.

VG resizable:-
Syntax:- vgchange -x n Vgname
or
vgchange -x y vgName
where -x, --resizeable {y|n}, Enables or disables the extension/reduction of this volume group with/by physical volumes.

Changing UUID of the VG:-
Syntax:- vgchange -u Vgname
ex:- vgchange -u datavg.
Make sure that the VG is in inactivate state (ie all LVs are not active in the VG) before you change UUID. You will get a random UUID assigned to the VG.

Vgmerge:-
Syntax:- vgmerge Vgname Vgname_tobe_merged.
Note:-
i) both VGs PE sizes must be same
ii) All LVs in the vg which is going be merged must be inactivated state
ex:- vgmerge datavg appvg

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

After the sucessfull execution of the above command all PVs and LVs that are there in appvg will be moved to datavg. So appvg will be eventually removed.

vgexport:-
vgexport command allows you to make the inactive VolumeGroupName(s) unknown to the system. You can then move all the Physical Volumes in that Volume Group to a different system for later vgimport.
Syntax:- vgexport Vgname

vgimport:-
vgimport allows you to make a Volume Group that was previously exported using vgexport known to the system again, perhaps after moving its Physical Volumes from a different machine.
Syntax:- vgimport VolumeGroupName

Renaming a Logical Volume:-
lvrename Oldname NEwName
Example: lvrename /dev/oravg/oralv01 /dev/oravg/mylv01

Resizing Logical Volume:-
i)Increase size of an LV:-
lvextend -L 12G /dev/Vgname/Lvname
the Lv will resized to 12GB
ii)
lvextend -L +1G /dev/vgname/Lvname
the lv will be added with 1Gb space to existing size.
iii)
lvextend -l +100%FREE /dev/myvg/testlv
all the left over space in the VG myvg is added to testlv.

You can reduce Lv size, and below are examples:-
Syntax:- lvreduce -l -n /dev/vgname/lvname
where -n is number of LEs
ex:-
lvreduce -l -3 /dev/datavg/lv01
the lv is decreased by 3 LEs.
Changing LV as readonly(Changing parameter to read only):-
lvchange -pr /dev/vgname/lvname
ex:-
lvchange -pr /dev/datavg/lv01
where p : permission
r : read only
rw : read and write
The above command will make lv as read only.

PVmove command:-
Syntax:-
pvmoe sourcePV DestinPV
All the LVs in the source PV will be moved to destination PV.
or

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

pvmove -n Lvname SourcePV DestinPV
The given lvnmame is moved from SourcePV to Destination PV.
Note:- you need to make sure that the destination pv can accommodate migrated LVs. ie make sure you have enough free PEs exists in the destination PV.
Example:
pvmove /dev/sdb /dev/sdc
pvmove -n oralv01 /dev/sdd /dev/sdc
While pvmove is taking place, you will get to know how much percentage of data is being migrated. However you can run this command in the background by including -b flag as shown in the below example:
pvmove -b /dev/sdb /dev/sdc
You can also execute pvmove command on the fly (ie when the filesystem is mounted).

VGsplit command:-
if VG contains more than one PV and you want to split the VG into two, you can do it using vgsplit command. You need to make sure that there are no LVs that are being spread on PVs which you are going to split. Also, you need to make LVs to inactive state that are being split.
Syntax:-
vgsplit VgNameTobeSplit NewVG PVName
ex:- vgsplit datavg appVG /dev/sde
Note:- all the LVs that are there in /dev/sde are to be in inactivate state. So you may need to use lvchange -an /dev/Vgname/Lvname command against all the lv in the PV which is going to be split.

Attributes while listing Volume Groups:-
m : mirror
M : Mirrored without initial sync
r : read
w : write
z : resizable
p : Partial
s : Snapshot
S : Invalid Snapshot
V : Virtual
i : Mirror Image
I : Mirror Image without Sync
c : Under Construction
- : Simple Volume
a : Active
o : Opned
t : thinly provisioned device

Attribut postions (while list lvs), are in 9 positions:

```
- - - - - - - - -
1 2 3 4 5 6 7 8 9
```

Each portions given below:
Postion1:- Volume Type: (m) mirrored, (M) Mirrored without initial sysnc, (o) origin, (O) origin with mergin snapshot, (r) raid, (R) Raid without initial sync, (s) shnapshot, (S) Invalid Snapshot, (p) PVmove, (V) virtual Mirror or raid image, mirror or raid image out of sync, mirror (l) logdevice, under (c)construction, thing(v)olume, (t)hin pool, (T)hinpool data, raid or pool m(e)tadata.

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Positoion2: Permission:- (w)ritable, (r)eadonly, (R)eadonly activation of non-read only volume.

Pisition3: Allocation Policy:- (a)anywhere, (c)ontiguous, (i)nherited, (c)cling, (n)ormal.

Position4: Fixed (m)irror

Position5: State:- (a)ctive, (s)uspend, (I)nvalid snashot, invlid (S)uspended snapshot, snapshot (m)erge failed, Suspended snapshot (M)erge failed, Mapped device present with (i)nactive table.

Position6: Device (o)pen

Position7: Target Tupe: (m)irror, (r)aid, (s)napshot, (t)hin, (u)nknown, (v)irtual

Position8:- Newly allocated data blocks are overwritten with block of 2 errors before use

Position9:- Partial - One or more of the physical volumes that are being used by this LV missing from this system.

Snapshot Logical Volume:-
The LVM snapshot feature provide the ability to create virtual images of a device at a particular instant without causing a service interuption. When a change is made to the original device ie origin, after a snapshot is taken, the snap shot feature makes a copy of changed data area as it was prior to change, so that it can reconstruct the state of the device.

Because the snapshot only copies only the data areas that change after teh snapshot is created, the snapshot requires minimal amount of storage. Snapshot is not substitue for backup procedure.

Advantages:-
1. When you need to perform a backup of LV, without halting the system that is continuously updating the data.
2. You can execute fsc on a snapshot filesystem to check filesystem integrity and determine whether the original filesystem requires repair.
3. You can test applications by taking a snapshot by leaving the real data untouched.

Syntax to create snapshot logical volume:-

Using -s flag of lvcreate command as below
lvcreate -n SnapLVName -L SizeofLV /dev/Vg/LVNAmetoWhichSnapTobeCreated
Example:-
1.Create origanal LV (not snap)
lvcreate -n oralv -L 1G oravg /dev/sde
2. Create Filesystem
mkfs.ext4 /dev/oravg/oralv
3.create a mount point and mount it,
mkdir /fs01
mount /dev/oravg/oralv /fs01
Verify the same with df -h command.
4.Now create snapshot logical volume with reference to the above created /dev/oravg/oralv.

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Assume the snapshot lv name as snaplv01 and below is the command:
lvcreate -s -L 128M -n snaplv01 /dev/oravg/oralv
Where -s : create snapshot lv
verify with lvs -a /dev/oravg/oralv
lvs -a /dev/oravg/snaplv01

Merging snapshot logical volume:-
lvconvert command is used to merge snapshot into origin. When merge starts, the resulting logical volume will have origin name. Once the merge finishes, the snapshot is removed.
Syntax:-
lvconvert --merge /dev/Vgname/SnapLvname
ex:-
lvconvert --merge /dev/oravg/snaplv01

Creating Filesystem:-

mkfs -t typeOfFileSystem /dev/devname
where devname is a whole disk or partition or an lv.

example:-

mkfs -t ext4 /dev/sdb1

Mounting Filesystem:-

Synatx:-
mount /dev/devname /mntPoint
example:-
mount /dev/sdb1 /oracle
To see mounted filesystems:
df
or
df -h
where -h : human readable output
df -m
prints filesystem in MB units

How to remove disk from OS:- (VMware Virtual SCSI disk or Physical SCSI disk):

Step1:-
cd /sys/class/block
Step2:
use ls command identify the disk (seen as a directory)
Example:
dm-0 dm-1 dm-2 dm-3 dm-4 dm-5 dm-6 sda sda1 sda2 sdb sdc sdd sr0
Assume that I want to remove sdd disk. Then get into that directory.
cd sdd

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

cd device
Step 3.
Now You will find a file delete
use the below command:
echo 1 > delete
use lsblk and observe that the disk is removed from OS

How to scan the attached scsi disk from OS (VMware Virtual SCSI or Physical SCSI):

Step1:
cd /sys/class/scsi_host
Identify to which host the disk is attached (you find out form Virtual machine settings)
Assume it is host0.
Step2:
cd host0
Now observe that you have a file with name 'scan'
Use the below command to scan all devices attached to host0
echo "- - -">scan
Where
- - - refers to dummy c t l (Channel Target Lun).
- - -
Step3: verify with lsblk and observe that the scanned disk is available.

Thinly provisioned Logical Volumes:-

Thinly provisioned logical volumes allow you to create lv that are larger than the available extents. Using thin provisioning you can manage a storage pool of free space known as thin pool, which can be allocated to an arbitrary number of devices when needed by the application.
You can create devices that can be bounded to the thin pool for later allocation when an application actually writes to the LV. The thin pool can be extended dynamically when needed for cost-effective allocation of storage space.

Creating thin pool:-
To create a thin pool, we perform the following tasks:
1. Create a Volume Group with vgcreate command
   vgcreate -s 32M datavg /dev/sdb /dev/sdc /dev/sdd
2. Create a thin pool with lvcreate command
   lvcreate -L Size -T /dev/vgname/thinpoolname
   ex:-
   lvcreate -L 10G -T /dev/datavg/devthinPool
3. Create a thin logical volume using the above created thin pool. Use lvcreate command for the same.
   lvcreate -V VirtSizeOfLV -n ThinLVname -T /dev/VgName/ThinPoolName
   ex:-
   lvcreate -V 5G -n tlv01 -T /dev/datavg/devthinPool
4. Create FS on the above create thin lv and mount it.

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Create more lvs that goes beyond 10G in total size and observe that thin lvs can consume more than the space allocated. However, if lvs actually writes the data, you might need to extend thin pool dynamically with lvextend command. Else, write operations will fail and might cause you application to go down.

Changing vgresize attribute:-
vgchange -x n vgname
vgs
observe there is no z in the attributes (which mean no resize is possible)
You can not extend or reduce the vg when it is set as non-resizable

vgchange -x y vgname
here -x y: resizable: Yes

Note: create a vg and extend the vg with pv. Do this with -xy and -xn attribes and observe the behaviour.

Changing UUID of vg or LV:
vgchange -u vgname : Change UUID of the vg
lvchange -u /dev/vg/lvname :Change UUID of the LV
Verify with vgdisplay and lvdisplay command.

Changing PE size of the VG:-
vgchange -s PESize VGName

Chage Max PVs per VG:-
vgchange -p MaxPvsCount vgname
ex:-
vgchange -p 32 datavg
observe the max pv count using vgdisplay command
To set max pvs to unlimited
vgchange -p 0 vgname
Note:- One PE size can be as max as 2TB

Increasing and decreasing logical volume:-
Increasing logical volume:-
lvextend -l NumberOfLEs /dev/vgname/lvname

example:
Assume that /dev/datavg/datalv is an LV with 10 LEs in size. Make this to have 25LE's, ie, add 15LEs in addition.

lvextend -l 25 /dev/datavg/datalv
or
lvextend -l +15 /dev/datavg/datalv
or
Assuming PE size is 16MB, a total a total of 10LEs for /dev/datavg/dlv01 as below(160MB)
lvcreate -l 10 -n dlv01 datavg /dev/sdb
Now add another 5 LEs (ie another 80MB, a total of 240MB). Here is the command:

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

lvextend -l +5 /dev/datavg/dlv01
or
lvextend -l 15 /dev/datavg/dlv01
or
lvextend -L 240M /dev/datavg/dlv01
or
lvexnted -L +80M /dev/datavg/dlv01

Decreasing the size of logical volume:-
Use lvreduce command to shrink the logical volume. Provide negative number with -l or -L for the same
Syntax:-
lvreduce -L -1G /dev/vg/lvname
or
lvreduce -l -10 /dev/vg/lvname
where -l referes to number of LEs. In the above, reduce 10 LEs on existing number of LEs
or
lvreduce -l 10 /dev/vg/lvname
the above example, will make /dev/vg/lvname to contain only 10 LEs if LV size is > 10 LEs. If the existing LV is already less than 10 LEs, it can not be reduced.
Examples:-

lvreduce -L -80M /dev/datavg/dlv01 (lv is reduced by 80MB on existing size)
lvreduce -L 240M /dev/datavg/dlv01 (LV size will become 240MB if it is > 240MB)
lvreduce -l 15 /dev/datavg/dlv01 (lv size will become 15 if it is > 15 Les)
lvreduce -l -5 /dev/datavg/dlv01 (lv size is reduce by 5 les on existing size)

Increasing filesystem(ext based):-
Steps:
LVM based filesystems can be increased dynamically (on the fly) - no need to unmount the filesystem. When the filesystem in use, you can still increase the filesystem.
Assume that the existing filesystem /oracle is 256MB and increase the same by 1G.
Assume the lvname is /dev/oravg/oralv
1. Increase the LV
  lvextend -L +1G /dev/oravg/oralv
Note that extending the lv will not increase the Filesystem. Use resize2fs command for the same.
2. verify if the lv is increased with lvdisplay command.
  use resize2fs command to crease the filesystem
  resize2fs /dev/oravg/oralv +1G
  verify with df -h and observe the FS is increased.

Decreasing the filesystem(ext based):-
Steps:-
You need unmount the filesystem to decrease the filesystem.
After umount, you need to run fsck command before you apply lvreduce command.
Assume that you have 1.5GB of Filesystem and decided to reduce 512MB of space. You have data available 256M, so that decreasing 512M will left 1GB space to fit the 256MB of data.
Assume lvname is /dev/oravg/oralv. Here are the steps:

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

1.df -h to verify existing lv and its mount and FS size
lvdisplay /dev/oravg/oralv
umount /dev/oravg/oralv or umount /oracle
2.run e2fsck command against the filesystem
e2fsck -f /dev/oravg/oralv
3.Reduce the filesystem by 512MB, So the resulting FS size will be 1GB
resize2fs /dev/oravg/oralv 1G
4.reduce the lv
lvreduce -L -512M /dev/oravg/oralv (result size will be 1GB LV)
5.Mount the filesystem
mount /dev/oravg/oralv /oracle
6.Verify with df -h to confirm if the FS is resized.
Note: Decreasing filesytem is destruction activity. Make sure that you have backup before performing any changes.
If fsck throws any errors, do not proceed with decreasing the FS.

Creating a Swap Space:-
This can be done in 3 ways:
1. By using disk partition with Hex Code: 82
2. By using a logical volume
3. By using a File

1.Creating swap space using the disk partition:-

1. Create a partitiona with fdisk or gdisk and change its partition type by choosing hex code 82.
2. Now create swap space on that partition
mkswap /dev/DevPartition
ex:-
mkswap /dev/sdf1
3. Now activate the created swap
syntax:
swapon /dev/SwapDevName
swapon /dev/sdf1
4. Verify with
swapon -s or
free -h commands
Note: To make swap to be persistent across reboots, you need to introduce the stanza in /etc/fstab as below:
vi /etc/fstab
/dev/sdf1 swap swap defalts 0 0
or
UUID=XXXXXXX swap swap defaults 0 0

2.Creating swap space using the LVM logical volume:-
Create a logical volume and use mkswap on that LV as show in the below example:

lvcreate -n swaplv -L 1G swapvg /dev/sdb
mkswap /dev/swapvg/swaplv
swapon /dev/swapvg/swaplv

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Verify with free -m or free -h or swapon -s
vi /etc/fstab
/dev/mapper/swavg-swaplv swap swap defaults 0 0
Tha above will make swap activation persistent across reboots.

To list swap spaces:-
swapon -s
To list swap usage:-
swapon -s
or free -m
or free -h etc.,

To activate swap space:-
swapon /dev/vg/swaplvname
or
swapon /dev/swapPartitionName
or
swapon /path/ToSwapFile

To deactivate swap space:-
swapoff /dev/vg/swaplvname
or
swapoff /dev/swapPartitionName
or
swapoff /path/toSwapFile

To activate swap from /etc/fstab:
use swapon -a command
where -a is used to traverse the /etc/fstab file and activates all swapspaces found

3.Using a file in the filesystem as a swap file to create a swapspace:-
1. Create a file of size 512M (assume i want to create swap sapce of 512MB).
Syntax:-

  fallocate -l size /path/to/File
example:-
  fallocate -l 512M /sfs01/swapFile01
or
  dd if=/dev/zero of=/sfs01/swapFile01 size=1MB count=512
2.
Create swapspace on the above created file
  mkswap /sfs01/swapFile01
3.
Activate the swap
  swapon /sfs01/swapFile01
4.
Add entry /etc/fstab to make it persistent

Increasing SWAP space:-

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

1. Deactivate swap sapce
   swapoff -v /dev/vgname/lvname
2. Increase LV size (assume you want extend by another 10 PEs)
   lvextend -l +10 /dev/vgname/lvname
3. Recreate the swap with mkswap
   mkswap /dev/vgname/lvname
4. Activate the same using swapon and introduce the entry in /etc/fstab for persistent activation
   swapon /dev/vgname/lvname
   and edit /etc/fstab to add the entry if it doesnt exists
   verify with swapon -s or fee -m etc.,

Reducing SWAP space:-

1. Deactivate the logical volume
   swaoff /dev/vgname/lvname
2. reduce the lv (assume reduce by 10 LEs in this example)
   lvreducce -l -10 /dev/vgname/lvname
3. Create swapspace on the device
   mkswap /dev/vgname/lvname
4. use swapon -va if the line is already introduced in /etc/fstab. Else use swapon /dev/vg/lvname
   verify with swapon -s or free -m or free -h

Note:- Swap info can also be found in /proc/swaps
Syntax:-
cat /proc/swaps

Removing swap space:-

1. Deactivate the swap by using swapoff command
2. Remove entry from /etc/fstab if exists
3. Remove LV with lvremove command or if it a file then use rm filename or if it is a partition use fdisk or gdisk to remove.
4. verify with free -m or free -g or free -h or swapon -s command
5. cat /etc/swaps and observe that the removed swap name was not listed.

Setting up a priority for the swap:-

The possible priorities are 0 - 32767. Higher number indicates higher priority.
Syntax:-
swapon -p Priority /dev/vg/lvname
ex:-
swapon -p 5 /dev/vg/swaplv
wehre -p: priority
To introduce the priority in /etc/fstab, below is the format:
/dev/vg/swaplv swap swap defaults,pri=5 0 0

LVM Commands Examples:-

#lvcreate -L 50 MB new_vg
lvcreate -v -L 50 MB new_vg
lvcreate -vvvv -L 50 MB new_vg
dd if= /d ev/zero of=PhysicalVolume bs=512 count=1

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

```
# pvcreate /dev/sdd /dev/sde /dev/sdf
# pvcreate /dev/hdb1
# lvmdiskscan
# pvdisplay
# pvscan
# pvchange -x n /dev/sdk1
# pvremove /dev/ram15
# vgcreate vg1 /dev/sdd1 /dev/sde1
# vgextend vg 1 /d ev/sd f1
# vgdisplay new_vg
# vgscan
# vgreduce my_volume_group /dev/hda1
# vgchange -l 128 /dev/vg00
# vgchange -a n my_volume_group
# vgremove officevg
# vgsplit bigvg smallvg /dev/ram15
# vgmerge -v databases my_vg
# vgrename /dev/vg02 /dev/my_volume_group
# vgrename vg02 my_volume_group
# lvcreate -L 10 G vg 1
# lvcreate -L 150 0 -n testl v testvg
# lvcreate -L 50 G -n g fsl v vg 0
# lvcreate -l 6 0 %VG -n myl v testvg
# lvcreate -l 10 0 %FR EE -n yo url v testvg
# vgdisplay testvg | g rep "Total PE"
# lvcreate -l 10 230 testvg -n mylv
# lvcreate -L 1500 -ntestlv testvg /dev/sdg1
# lvcreate -l 100 -n testlv testvg /dev/sda1:0-24 /dev/sdb1:50-124
# lvcreate -l 100 -n testlv testvg /d ev/sda1:0-25:100 -
# lvcreate -L 50G -i 2 -I 64 -n gfslv vg0
# lvcreate -l 100 -i 2 -nstripelv testvg /dev/sda1:0-49 /dev/sdb1:50-99
# lvcreate -L 50G -m1 -n mirrorlv vg0
# lvcreate -m1 -L 2T -R 2 -n mirror vol_group
# lvcreate -L 12MB -m1 --mirrorlog core -n ondiskmirvol bigvg
# lvcreate -L 500 M -m1 -n mirrorlv -alloc anywhere vg0
# lvcreate -L 12MB -m1 --mirrorlog mirrored -n twologvol bigvg
# lvcreate -L 500M -m1 -n mirrorlv vg0 /dev/sda1 /dev/sdb1 /dev/sdc1
# lvcreate -L 500M -m1 -n mirrorlv vg0 /dev/sda1:0-499 /dev/sdb1:0-499 /dev/sdc1:0
```