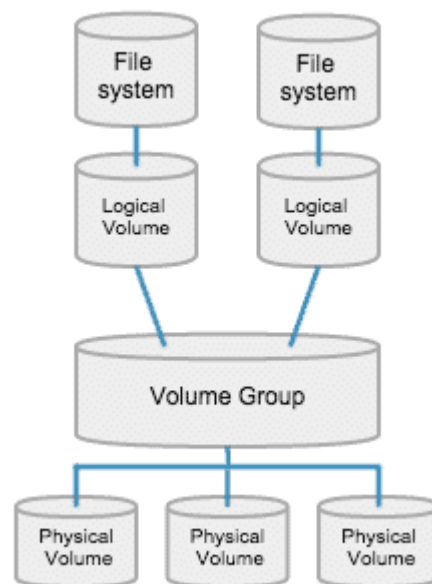


## LVM(Logical Volume Manager):

Lvm is used to increase the filesystem capacity by keeping filesystem 100% online which means without un-mounting. LVM can be used to gather existing storage devices into groups and allocate logical units from the combined space as needed.

### LVM structure:



There are 3 concepts that LVM manages:

- Physical Volumes
- Volume Groups
- Logical Volumes

### Physical Volume:

To use a partition or a full hard disk for LVM purpose we should convert those partitions or hard disks into physical volumes.

### Volume Group:

Volume Group is a virtual hard disk which is created by using physical volumes.

### Logical Volume:

Logical volume is a partition in the volume group.

### Creating physical volumes:

Use the below command and syntax to create physical volumes.

syntax: `pvccreate <partition-name> <partition-name2> e.t.c`

ex: `pvccreate /dev/sda1 /dev/sda2`

And use `pvs` and `pvdisplay` commands to list the physical volumes information.

### Creating Volume Group:

Use the below command and syntax to create volume group.

Syntax: `vgcreate <volume-group-name> <physical-volume1> <physical-volume2>`

Ex: `vgcreate myvg /dev/sda1 /dev/sda2`

Note: We cannot use one physical volume to create multiple volume groups.

Use `vgs` and `vgdisplay` commands to list volume group information.

### Creating Logical Volume:

Use the below command and syntax to create logical volumes.

Syntax: `lvcreate -L <size> -n <lv-name> <vg-name>`

Ex: `lvcreate -L 2G -n mylv myvg`

Use `lvs` and `lvdisplay` commands to list logical volumes information.

Once after creating the logical volume, format it with desired filesystem and mount it by following either of one way.

## Extending the filesystem capacity without unmounting:

Follow the below steps to extend the filesystem capacity

1. First extend the capacity of a logical volume by following below command and syntax.

```
# lvextend -L <+size> -n <Full logical volume name>
```

2. resize the filesystem capacity by using below command and syntax.

```
# resize2fs <file system name>
```

3. Use the command “df –h” to cross verify.

## Reducing the filesystem capacity:

Follow the below steps one by one to reduce the filesystem capacity.

1. Umount the filesystem

```
# umount <mount-point-name> or <filesystem-name>
```

2. Check the healthiness of the filesystem.

```
# e2fsck -f <filesystem-name>
```

3. Reduce the filesystem capacity.

```
# resize2fs <filesystem-name> <size>
```

4. Reduce the logical volume size.

```
# lvreduce -L <size> <full logical volume name>
```

5. Mount the filesystem again

```
# mount <filesystem-name> <mount-point>
```

6. Execute “df –h” to cross-verify.

## Extending the size of the volume group:

To extend the size of the volume group, use the below command and syntax.

```
# vgextend <volume-group-name> <physical-volume-name>
```

Use “vgs” or “vgdisplay” command to verify.

## Reducing the size of volume group:

To reduce the size of the volume group, use the below command and syntax.

```
#vgreduce <volume-group-name> <physical-volume-name>
```

Use “vgs” or “vgdisplay” command to verify.

## To delete the LVM structure completely, follow the below steps.

1. Umount all logical volumes.
2. Delete all logical volumes

```
# lvremove <full-logical-volume-name>
```

3. Delete the volume group

```
# vgremove <volume-group-name>
```

4. Remove lvm labelling on physical partitions.

```
# pvremove <physical-volume-names>
```

Due to several reason a volume group may become corrupted after unintended user intervention. LVM will not be aware of this situation unless the alternate is already on the volume group, and may allow to add the alternate path to a different volume group. Two of the useful commands provided as part of LVM are `vgcfgbackup` and `vgcfgrestore`. These commands allow you to **back up the metadata of a volume group**. This may not seem like much, but there are a few scenarios where having a backup of the volume group metadata can save some time and maybe sanity as well.

By default, the metadata backup is stored in the `/etc/lvm/backup` file and the metadata archives are stored in the `/etc/lvm/archive` file. At the top of each file it will tell you the time/date when the file was generated so chances are you'll have a copy of the older metadata as it was before the irreversible operations.

## 1. `vgcfgbackup`

Whenever you do any lvm activities like lvcreate, remove, disk add... should be updated in the respective volume group conf file under `/etc/lvm/` directory. If disk fails we can restore the LVM configuration onto the disk using this backup configuration file. `vgcfgbackup` command takes the configuration metadata from the lvm header of the disk and save to a default file location `/etc/lvm/backup/vg_name`. By default, `vgcfgbackup` runs automatically each time, the lvm command changes the lvm configuration. It reads the LVM header details from system area of the disk and copies it to file. This file helps you to restore configuration on newly added disk in place of old disk which may have got corrupted or failed.

Note that a metadata backup does not back up the user and system data contained in the logical volumes. It is recommended to have this backup taken after every LVM level change. By default all LVM commands altering LVM details are designed to take this backup automatically hence manually running command is not necessary.

```
# vgcfgbackup [-f path_newfilename ] vg_name
```

- **vg\_name:** the path name of a volume group
- **-f path\_newfilename:** save the configuration using the new filename specified at the path specified. If **-f** is omitted, the default file name is in the form of `/etc/lvm/backup/vg_name` where `vg_name` is the base name of `vg_name`. If you don't specified a new path, it will save the new file in the current directory.

Example:

```
# vgcfgbackup /dev/vg-01
```

Volume group "vg-01" successfully backed up.

We can see the message our backup is a success. Let's see the backup folder

```
# ls -l /etc/lvm/backup/
```

```
total 8

-rw-----. 1 root root 1715 Apr 25 16:50 centos

-rw-----. 1 root root 4089 May 4 01:39 vg-01
```

The default filename is similar to our volume group name. Now, let's see what happen when we use `-f` option

```
# vgcfgbackup -f vg-01_bak /dev/vg-01

Volume group "vg-01" successfully backed up.
```

It is a success. Let's check our file `vg-01_bak` in backup directory

```
# ls /etc/lvm/backup/

centos vg-01
```

Only the previous file exists. We don't see our file. Now let's check in the current directory

```
# ls -l

total 56392

-rwxrwxrwx. 1 admin admin 1975 Apr 25 17:01 manifest.txt

-rwxrwxrwx. 1 admin admin 2287 Apr 25 17:01 run_upgrader.sh

-rw-----. 1 root root 4074 May 4 02:05 vg-01_bak
```

You can see our file on the last line. Now let's retry the same command but we will specify the default backup folder.

```
# vgcfgbackup -f /etc/lvm/backup/vg-01_bak /dev/vg-01
```

Volume group "vg-01" successfully backed up.

Now let's check.

```
# ls -l /etc/lvm/backup/  
  
total 12  
  
-rw-----. 1 root root 1715 Apr 25 16:50 centos  
  
-rw-----. 1 root root 4089 May 4 01:39 vg-01  
  
-rw-----. 1 root root 4090 May 4 02:26 vg-01_bak
```

In many cases, lvm configuration gets corrupted. lvm could also be corrupted on some disk failures and on reboot OS fails to boot up. So it is important to have the lvm configuration backup.

## 2. vgcfgrestore

The vgcfgrestore command restores [LVM volume group configuration](#) from the archive to all the physical volumes in the volume groups. Restore cannot be performed if the volume group is activated in shared mode. Make sure you make a current copy of the existing working configuration using the vgcfgbackup command with the `-f` flag to specify a different file for the output so that you don't alter any files that are in the `/etc/lvm/backup` or `/etc/lvm/archive` folders. Make sure you diff the current configuration with the configuration you wish to restore to verify that the only changes you're about to apply are to revert the erroneous operations.

```
vgcfgrestore [-f|--file <filename>] [-t|--test] [-v] vg_name
```

- **vg\_name:** the name of the volume group corresponding

- `-f filename:` indicates the name of LVM metadata backup file which specifies a metadata backup or archive file to be used for restoring the `vg_name`
- `--test:` can be used to check the error of the command which can occur
- `-v:` to have the detail of what currently happens

In the example below, we will simulate the deletion of a logical volume and we will try to restore it.

**NB:** The following example can be **dangerous** and **destructive** so be very careful and if possible have a full backup. Please make sure to try it on a test environment

*step1:* Remove a logical volume

```
# lvremove /dev/vg-01/lv_linear
```

```
Do you really want to remove active logical volume lv_linear? [y/n]: y
```

```
Logical volume "lv_linear" successfully removed
```

Check

```
# lvs
```

```
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
```

```
root centos -wi-ao---- 17.47g
```

```
swap centos -wi-ao---- 2.00g
```

```
lv_mirror vg-01 rwi-a-r--- 200.00m 100.00
```

```
lv_stripe vg-01 -wi-a----- 1.00g
```

*step 2:* now look for the archive file at `/etc/lvm/archive/vgname_(xxxxx).vg`. Just looking the date that you have removed the logical volume. In our case it's the last file of the list



```
# ls -l /etc/lvm/archive/
```

```
total 28
```

```
-rw-----. 1 root root 1716 Apr 25 16:50 centos_00000-34432512.vg
```

```
-rw-----. 1 root root 1562 May 1 16:02 vg-01_00002-1413920221.vg
```

```
-rw-----. 1 root root 2009 May 1 16:07 vg-01_00003-718049637.vg
```

```
-rw-----. 1 root root 4097 May 4 03:51 vg-01_00004-306926489.vg
```

*step 3:* verify the content of the file to check the time/data when the file was generated and the description which indicates action done before

```
# head /etc/lvm/archive/vg-01_00004-306926489.vg
```

```
# Generated by LVM2 version 2.02.130(2)-RHEL7 (2015-10-14): Thu May 4 03:51:35 2017
```

```
contents = "Text Format Volume Group"
```

```
version = 1
```

```
description = "Created *before* executing 'lvremove /dev/vg-01/lv_linear'"
```

```
creation_host = "centos7-srv" # Linux centos7-srv 3.10.0-514.16.1.el7.x86_64 #1
```

```
creation_time = 1493866295 # Thu May 4 03:51:35 2017
```

You can see the description which indicates that the file was created before executing lvremove command.

*step 4:* Now recover it. You can test before recovering with `--test` option

```
# vgcfgrestore vg-01 -v -f /etc/lvm/archive/vg-01_00004-306926489.vg

Found same device /dev/sdb with same pvid 3vXbKwkq63cBiydWgQfd2aaBeYo6b5I

Found same device /dev/sdc with same pvid E9wih1232IjADxwkSwRC9kVV3VPJbD5O

Restored volume group vg-01
```

*step 5:* Now we can check it

```
<# lvscan

ACTIVE '/dev/centos/swap' [2.00 GiB] inherit

ACTIVE '/dev/centos/root' [17.47 GiB] inherit

inactive '/dev/vg-01/lv_linear' [1.00 GiB] inherit

ACTIVE '/dev/vg-01/lv_stripe' [1.00 GiB] inherit

ACTIVE '/dev/vg-01/lv_mirror' [200.00 MiB] inherit
```

You can see that our logical is not active in the third line. We will activate it and check again

```
# lvchange -ay /dev/vg-01/lv_linear

# lvscan | grep lv_linear

ACTIVE '/dev/vg-01/lv_linear' [1.00 GiB] inherit
```