5 easy steps to recover LVM2 partition, PV, VG, LVM metdata in Linux

WRITTEN BY - ADMIN

Updated On November 11, 2023

Table of Contents

In this article we will learn

- How to recover LVM2 partition (Restore deleted LVM)
- How to restore PV (Physical Volume) in Linux
- How to restore VG (Volume Group) in Linux
- How to restore LVM metadata in Linux

Earlier we had a situation wherein the LVM metadata from one of our CentOS 8 node was missing. Due to this all the logical volumes, volume groups and physical volumes mapped to that LVM metadata was not visible on the Linux server. So we had to restore LVM metadata from the backup using vgcfgrestore. I will share the steps to reproduce the scenario i.e. manually delete the LVM metadata and then steps to recover LVM2 partition, restore PV, restore VG and restore LVM metadata in Linux using vgcfgrestore.

vgcfgbackup can be used to manually create LVM backups, as these backups are very helpful and can also be used in LVM Disaster Recovery.

ALSO READ:

Step by Step Guide to perform LVM backup and restore using LVM snapshot (RHEL/CentOS 7/8)
How to boot a Linux host using LVM snapshot with BOOM Utility to verify the Snapshot content (CentOS/RHEL 8)

Prepare Lab Environment

Before we go ahead with the steps to recover LVM2 partition in Linux, we must first prepare Lab Environment with logical volumes. Next **we will manually delete lvm metadata** to reproduce the issue scenario.

I have created a Virtual Machine with CentOS 8 OS using Oracle VirtualBox which is installed on a Linux server. Next I added an additional virtual disk to this VM which is mapped to `/dev/sdb`.

Still installing Linux manually?

I would recommend to configure **one click installation** using Network PXE Boot Server. Using PXE server you can install Oracle Virtual Machines or KVM based Virtual Machines or any type of physical server **without any manual intervention** saving time and effort.

ALSO READ:

Create & Manage Striped Logical Volume Linux [Step-by-Step]

Create Physical Volume

The first step is to create physical volume using `pvcreate`

bash

```
[root@centos-8 ~]# pvcreate /dev/sdb
  Physical volume "/dev/sdb" successfully created.
```

Create Volume Group

Next create a new Volume Group, we will name this VG as `test_vg`.

bash

```
[root@centos-8 ~]# vgcreate test_vg /dev/sdb
  Volume group "test_vg" successfully created
```

List the available volume groups using `vgs`. I currently have two volume groups wherein `rhel` volume group contains my system LVM2 partitions

bash

```
[root@centos-8 ~]# vgs
  VG      #PV #LV #SN Attr   VSize   VFree
  rhel      1   2   0 wz--n- <14.50g     0
  test_vg   1   0   0 wz--n-  <8.00g <8.00g  <-- new VG
```

Create Logical Volume

Create a new logical volume `test_lv1` under our new volume group `test_vg`

bash

```
[root@centos-8 ~]# lvcreate -L 1G -n test_lv1 test_vg
  Logical volume "test_lv1" created.
```

Create File System on the Logical Volume

Create ext4 file system on this new logical volume

bash

```
[root@centos-8 ~]# mkfs.ext4 /dev/mapper/test_vg-test_lv1
mke2fs 1.44.6 (5-Mar-2019)
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: c2d6eff5-f32f-40d4-88a5-a4ffd82ff45a
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

List the available volume groups along with the mapped storage device. Here as you see `test_vg` is mapped to `/dev/sdb`

bash

```
[root@centos-8 ~]# vgs -o+devices
  VG      #PV #LV #SN Attr   VSize   VFree  Devices
  rhel      1   2   0 wz--n- <14.50g     0  /dev/sda2(0)
  rhel      1   2   0 wz--n- <14.50g     0  /dev/sda2(239)
  test_vg   1   1   0 wz--n-  <8.00g <7.00g /dev/sdb(0)
```

Similarly you can see the new logical volume `test_lv1` is mapped to `/dev/sdb` device

bash

```
[root@centos-8 ~]# lvs -o+devices
  LV       VG      Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert Devices
  root     rhel    -wi-ao----  13.56g                                                     /dev/sda2(239)
  swap     rhel    -wi-ao---- 956.00m                                                     /dev/sda2(0)
  test_lv1 test_vg -wi-a-----   1.00g                                                     /dev/sdb(0)  <-- new Logical Volume
```

Add some data to Logical Volume

We will put some data into our logical volume to make sure there are no data loss after we recover LVM2 partition, restore PV and restore VG using LVM metadata in the next steps.

bash

```
[root@centos-8 ~]# mkdir /test
[root@centos-8 ~]# mount /dev/mapper/test_vg-test_lv1 /test/
```

Create a dummy file and note down the md5sum value of this file

bash

```
[root@centos-8 ~]# touch /test/file
[root@centos-8 ~]# md5sum /test/file
d41d8cd98f00b204e9800998ecf8427e  /test/file
```

Next un-mount the logical volume

bash

```
[root@centos-8 ~]# umount /test/
```

ALSO READ:

Beginners guide to how LVM works in Linux (architecture)

How to manually delete LVM metadata in Linux?

To manually delete LVM metadata in Linux you can use various tools such as `wipefs`, `dd` etc. `wipefs` can erase filesystem, raid or partition-table signatures (magic strings) from the specified device to make the signatures invisible for libblkid. **wipefs does not erase the filesystem itself nor any other data from the device**.

WARNING:

Execute this command wisely and is not recommended to be executed in production environments as it will delete all the file system signature of the device.

In this example we will use wipefs to delete LVM metadata from `/dev/sdb` device. Since the device in question `/dev/sdb` is in use by Volume Group, we have to use `-f` to forcefully wipe the LVM metadata

bash

```
[root@centos-8 ~]# wipefs --all --backup -f /dev/sdb
/dev/sdb: 8 bytes were erased at offset 0x00000218 (LVM2_member): 4c 56 4d 32 20 30 30 31
```

We have used `--backup` so that before deleting the LVM metadata, **wipefs will create a backup** of the ext4 signature containing LVM metadata under the home folder of the user who is executing the command. Since we used root user, our LVM metadata backup is stored under root user's home folder.

bash

```
[root@centos-8 ~]# ls -l /root/wipefs-sdb-0x00000218.bak
```

```
-rw------- 1 root root 8 Apr  5 13:45 /root/wipefs-sdb-0x00000218.bak
```

HINT:

To restore lvm metadata stored in the file system signature from the backup we can use `dd if=~/wipefs-sdb-0x00000218.bak of=/dev/sdb seek=$((0x00000218)) bs=1 conv=notrunc`

Next you can verify that all the logical volumes, volume groups and physical volume part of `/dev/sdb` is missing from the Linux server

bash

```
[root@centos-8 ~]# lvs -o+devices
  LV   VG   Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert Devices
  root rhel -wi-ao----  13.56g                                                     /dev/sda2(239)
  swap rhel -wi-ao---- 956.00m                                                     /dev/sda2(0)  <--Our Logical volume no more visible
```

bash

```
[root@centos-8 ~]# vgs
  VG   #PV #LV #SN Attr   VSize   VFree
  rhel   1   2   0 wz--n- <14.50g    0 <-- test_vg no more visible
```

bash

```
[root@centos-8 ~]# pvs
  PV         VG   Fmt  Attr PSize   PFree
  /dev/sda2  rhel lvm2 a--  <14.50g    0 <-- /dev/sdb no more visible
```

Similarly with `lsblk` also we can verify that there are no LVM2 partitions under `/dev/sdb`

bash

```
[root@centos-8 ~]# lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda             8:0    0   15G  0 disk
├─sda1          8:1    0  512M  0 part /boot
└─sda2          8:2    0 14.5G  0 part
  ├─rhel-root 253:0    0 13.6G  0 lvm  /
  └─rhel-swap 253:1    0  956M  0 lvm  [SWAP]
sdb             8:16   0    8G  0 disk
sr0            11:0    1 1024M  0 rom
sr1            11:1    1 1024M  0 rom
```

ALSO READ:

5 easy steps to resize root LVM partition in RHEL/CentOS 7/8 Linux

Step 1: List backup file to restore LVM metadata in Linux

- **LVM metadata backups and archives are automatically created** whenever there is a configuration change for a volume group or logical volume, unless this feature is disabled in the `lvm.conf` file.
- 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.
- How long the metadata archives stored in the `/etc/lvm/archive` file are kept and how many archive files are kept is determined by parameters you can set in the `lvm.conf` file.
- A daily system backup should include the contents of the `/etc/lvm` directory in the backup.
- You can manually back up the LVM metadata to the `/etc/lvm/backup` file with the `vgcfgbackup` command.
- You can restore LVM metadata with the `vgcfgrestore` command.

To list the available backups of LVM metadata use `vgcfgrestore --list`. Currently we have three backup stages where the last backup was taken after we created `test_lv1` logical volume.

bash

```
[root@centos-8 ~]# vgcfgrestore --list test_vg

  File:         /etc/lvm/archive/test_vg_00000-1327770182.vg
  VG name:      test_vg
  Description:  Created *before* executing 'vgcreate test_vg /dev/sdb'
  Backup Time:  Sun Apr  5 13:43:26 2020


  File:         /etc/lvm/archive/test_vg_00001-1359568949.vg
  VG name:      test_vg
  Description:  Created *before* executing 'lvcreate -L 1G -n test_lv1 test_vg'
  Backup Time:  Sun Apr  5 13:44:02 2020


  File:         /etc/lvm/backup/test_vg
  VG name:      test_vg
  Description:  Created *after* executing 'lvcreate -L 1G -n test_lv1 test_vg'
  Backup Time:  Sun Apr  5 13:44:02 2020
```

So we will use the last backup i.e. `/etc/lvm/backup/test_vg` to restore LVM metadata till the stage where `test_lv1` was created.

Step 2: Restore PV (Physical Volume) in Linux

IMPORTANT NOTE:

In my case the physical volume was also missing hence I am creating a new Physical Volume, but if in your case your Physical Volume is present and only Volume Groups and Logical Volumes are missing then you can ignore this step.
You must perform proper pre-checks and take backup of your file system before executing these steps in production environment to prevent any data loss.

- It is **very important that to restore PV, you create the new PV using the same UUID** as it was earlier or else restore VG and recover LVM2 partition will fail in the next steps.
- You can get the UUID of your Physical Volume from backup file " `/etc/lvm/backup/test_vg` "
- Below is a sample content of physical_volumes from the backup file. If you have more than one physical volumes then you need to search for the missing PV's UUID
- In my case `SBJi2o-jG2O-TfWb-3pyQ-Fh6k-fK6A-AslOg1` is the UUID of the missing PV so I will use this to restore PV in Linux

bash

```
        physical_volumes {

                pv0 {
                        id = "SBJi2o-jG2O-TfWb-3pyQ-Fh6k-fK6A-AslOg1"
                        device = "/dev/sdb"     # Hint only
```

```
                        status = ["ALLOCATABLE"]
                        flags = []
                        dev_size = 16777216     # 8 Gigabytes
                        pe_start = 2048
                        pe_count = 2047 # 7.99609 Gigabytes
                }
        }
```

Next again it is **important** that you test the physical volume restore. **We use `--test` mode to verify the operation**. With `--test` commands will not update LVM metadata. This is implemented by disabling all metadata writing but nevertheless returning success to the calling function.

ALSO READ:

Create Mirrored Logical Volume in Linux [Step-by-Step]

So here I have provided the same UUID of `/dev/sdb` as we collected earlier, followed by the backup file we want to use to restore PV and then the device name using which we will perform pvcreate. The `pvcreate` command overwrites only the LVM metadata areas and does not affect the existing data areas.

bash

```
[root@centos-8 ~]# pvcreate --test --uuid "SBJi2o-jG2O-TfWb-3pyQ-Fh6k-fK6A-AslOg1" --restorefile /etc/lvm/backup/test_vg /dev/sdb
  TEST MODE: Metadata will NOT be updated and volumes will not be (de)activated.
  WARNING: Couldn't find device with uuid SBJi2o-jG2O-TfWb-3pyQ-Fh6k-fK6A-AslOg1.
  Physical volume "/dev/sdb" successfully created.
```

With `--test` mode we know that the command execution is successful. So we will run the same command without `--test` to restore PV in real.

bash

```
[root@centos-8 ~]# pvcreate  --uuid "SBJi2o-jG2O-TfWb-3pyQ-Fh6k-fK6A-AslOg1" --restorefile /etc/lvm/backup/test_vg /dev/sdb
  WARNING: Couldn't find device with uuid SBJi2o-jG2O-TfWb-3pyQ-Fh6k-fK6A-AslOg1.
  Physical volume "/dev/sdb" successfully created.
```

Next verify the list of available Physical Volumes

bash

```
[root@centos-8 ~]# pvs
  PV         VG   Fmt  Attr PSize   PFree
  /dev/sda2  rhel lvm2 a--  <14.50g    0
  /dev/sdb        lvm2 ---    8.00g 8.00g  <-- Now /dev/sdb is visible
```

Step 3: Restore VG to recover LVM2 partition

- After we restore PV, next step is to restore VG which will further recover LVM2 partitions and also will recover LVM metadata.
- Similar to `pvcreate`, we will **execute vgcfgrestore with `--test` mode** to check the if restore VC would be success or fail.
- This command will not update any LVM metadate

bash

```
[root@centos-8 ~]# vgcfgrestore --test -f /etc/lvm/backup/test_vg test_vg
  TEST MODE: Metadata will NOT be updated and volumes will not be (de)activated.
  Restored volume group test_vg.
```

As we see that the command execution in `--test` mode was successful so now we can safely execute our command to restore VG and recover LVM2 partition in Linux using `vgcfgrestore`.

bash

```
[root@centos-8 ~]# vgcfgrestore  -f /etc/lvm/backup/test_vg test_vg
  Restored volume group test_vg.
```

Using `vgs` your can check if restore VG was successful.

bash

```
[root@centos-8 ~]# vgs
  VG      #PV #LV #SN Attr   VSize   VFree
  rhel      1   2   0 wz--n- <14.50g     0
  test_vg   1   1   0 wz--n-  <8.00g <7.00g  <-- test_vg is not visible
```

Next verify the if you were able to restore deleted lvm and recover LVM2 partition using `lvs`.

bash

```
[root@centos-8 ~]# lvs
  LV       VG      Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  root     rhel    -wi-ao----  13.56g
  swap     rhel    -wi-ao---- 956.00m
  test_lv1 test_vg -wi-------   1.00g  <-- our logical volume is also visible
```

ALSO READ:

How to mount filesystem in certain order one after the other in CentOS/RHEL 7 & 8

Step 4: Activate the Volume Group

Next activate the volume group `test_vg`

bash

```
[root@centos-8 ~]# vgchange -ay test_vg
  1 logical volume(s) in volume group "test_vg" now active
```

Step 5: Verify the data loss after LVM2 partition recovery

The most crucial part, make sure there was no data loss in the entire process to restore PV, restore VG, restore LVM metadata and recover LVM2 partition.

bash

```
[root@centos-8 ~]# mount /dev/mapper/test_vg-test_lv1 /test/
```

If we are able to mount the logical volume so **it means our ext4 file system signature is intact** and not lost or else the mount would fail.

```bash
[root@centos-8 ~]# ls -l /test/
total 16
-rw-r--r-- 1 root root     0 Apr  5 13:45 file
drwx------ 2 root root 16384 Apr  5 13:44 lost+found
```

Our test file exists and the `md5sum` matches the value of what we had taken before deleting the LVM metadata

```bash
[root@centos-8 ~]# md5sum /test/file
d41d8cd98f00b204e9800998ecf8427e  /test/file  <-- same as earlier
```

So overall restore PV, restore VG, restore LVM metadata and recover LVM2 partition was successful.

Lastly I hope the steps from the article to recover LVM2 partition using vgcfgrestore on Linux was helpful. So, let me know your suggestions and feedback using the comment section.

References:

Restore deleted LVM metadata

Performing Logical Volume Backup and restore using vgcfgbackup and vgcfgrestore

Related Searches: lvm backup and restore in linux, lvm disaster recovery, how to restore vg in linux, how to restore pv in linux, how to restore lvm metadata in linux

Recover LVM2 Partition, Storage