File System Management

1. Creating a file system without LVM

1.1. Display information about a free disk ("vdb"):

```
# fdisk -l

Disk /dev/vda: 10.5 GB, 10522670080 bytes, 20552090 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x00002d39

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1   *        2048     1050623      524288   83  Linux
/dev/vda2         1050624    20551679     9750528   8e  Linux LVM

...

Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

or

```
# lsblk
NAME             MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0               11:0    1 1024M  0 rom
vda              252:0    0  9.8G  0 disk
├─vda1           252:1    0  512M  0 part /boot
└─vda2           252:2    0  9.3G  0 part
  ├─vg00-root_lv 253:0    0    1G  0 lvm  /
  ├─vg00-swap_lv 253:1    0    1G  0 lvm  [SWAP]
  ├─vg00-usr_lv  253:2    0  4.8G  0 lvm  /usr
  ├─vg00-var_lv  253:3    0    1G  0 lvm  /var
  ├─vg00-tmp_lv  253:4    0    1G  0 lvm  /tmp
  └─vg00-home_lv 253:5    0  512M  0 lvm  /home
vdb              252:16   0   10G  0 disk
```

1.2. Create a primary partition on the disk that is 5 GB in size:

```
# fdisk /dev/vdb
Welcome to fdisk (util-linux 2.23.2).
```

```
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x610c7c18.

Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-20971519, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-20971519, default 20971519): +5G
Partition 1 of type Linux and of size 5 GiB is set

Command (m for help): p

Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x610c7c18

   Device Boot      Start         End      Blocks   Id  System
/dev/vdb1            2048    10487807     5242880   83  Linux

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

1.3. Optionally verify that the partition has been created:

```
# fdisk -l /dev/vdb

Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0xefe0464f

   Device Boot      Start         End      Blocks   Id  System
/dev/vdb1            2048    10487807     5242880   83  Linux
```

1.4. Inform the OS of partition table changes:

```
# partprobe
```

1.5. Format the partition with the XFS file system:

```
# mkfs.xfs /dev/vdb1
meta-data=/dev/vdb1              isize=512    agcount=4, agsize=327680 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=1310720, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

1.6. Create a mount point for the file system:

```
# mkdir /projects
```

1.7. Display the UUID of the partition:

```
# blkid /dev/vdb1
/dev/vdb1: UUID="d031e078-6d95-47ae-8d21-c955d841cf0e" TYPE="xfs"
```

1.8. Edit /etc/fstab to make the file system available permanently:

```
# echo 'UUID=d031e078-6d95-47ae-8d21-c955d841cf0e /projects xfs defaults 0
0' >> /etc/fstab
```

1.9. Update systemd with the new /etc/fstab configuration (RHEL/CentOS 7/8):

```
# systemctl daemon-reload
```

1.10. Mount the file system:

```
# mount /projects
```

1.11. Verify that the new file system is mounted:

```
# df -h /projects
Filesystem      Size  Used Avail Use% Mounted on
/dev/vdb1       5.0G   33M  5.0G   1% /projects
```

2. Removing a file system without LVM

2.1. Umount the "/projects" file system:

```
# umount /projects
```

2.2. Remove the mount point:

```
# rmdir /projects
```

2.3. Remove the entry from /etc/fstab:

```
# sed -i '/projects/d' /etc/fstab
```

2.4. Update systemd with the new /etc/fstab configuration (RHEL/CentOS 7/8):

```
# systemctl daemon-reload
```

2.5. Remove the partition:

```
# fdisk /dev/vdb
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): d
Selected partition 1
Partition 1 is deleted

Command (m for help): p

Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x610c7c18

   Device Boot      Start         End      Blocks   Id  System

Command (m for help): w
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
Syncing disks.
```

2.6. Optionally verify that the partition has been removed:

```
# fdisk -l /dev/vdb

Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0xefe0464f

   Device Boot      Start         End      Blocks   Id  System
```

2.7. Inform the OS of partition table changes:

```
# partprobe
```

2.8. Verify that the file system does not exist:

```
# df -h /projects
df: ‘/projects’: No such file or directory
```

---

3. Creating a file system within LVM

3.1. Creating a single file system

3.1.1. Display information about a free disk "vdb":

```
# fdisk -l /dev/vdb

Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

or

```
# lsblk /dev/vdb
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vdb  252:16   0  10G  0 disk
```

3.1.2. Create a primary partition of type "Linux LVM" across the entire disk (or continue directly with the "pvcreate" command below and use the entire disk):

```
# fdisk /dev/vdb
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-20971519, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-20971519, default 20971519):
Using default value 20971519
Partition 1 of type Linux and of size 10 GiB is set

Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): L

 0  Empty           24  NEC DOS         81  Minix / old Lin bf  Solaris
 1  FAT12           27  Hidden NTFS Win 82  Linux swap / So c1  DRDOS/sec
(FAT-
 2  XENIX root      39  Plan 9          83  Linux           c4  DRDOS/sec
(FAT-
 3  XENIX usr       3c  PartitionMagic  84  OS/2 hidden C:  c6  DRDOS/sec
(FAT-
 4  FAT16 <32M      40  Venix 80286     85  Linux extended  c7  Syrinx
 5  Extended        41  PPC PReP Boot   86  NTFS volume set da  Non-FS data
 6  FAT16           42  SFS             87  NTFS volume set db  CP/M / CTOS
/ .
 7  HPFS/NTFS/exFAT 4d  QNX4.x          88  Linux plaintext de  Dell Utility
 8  AIX             4e  QNX4.x 2nd part 8e  Linux LVM       df  BootIt
 9  AIX bootable    4f  QNX4.x 3rd part 93  Amoeba          e1  DOS access
 a  OS/2 Boot Manag 50  OnTrack DM      94  Amoeba BBT      e3  DOS R/O
 b  W95 FAT32       51  OnTrack DM6 Aux 9f  BSD/OS          e4  SpeedStor
 c  W95 FAT32 (LBA) 52  CP/M            a0  IBM Thinkpad hi eb  BeOS fs
 e  W95 FAT16 (LBA) 53  OnTrack DM6 Aux a5  FreeBSD         ee  GPT
 f  W95 Ext'd (LBA) 54  OnTrackDM6      a6  OpenBSD         ef  EFI
(FAT-12/16/
10  OPUS            55  EZ-Drive        a7  NeXTSTEP        f0  Linux/PA-
RISC b
11  Hidden FAT12    56  Golden Bow      a8  Darwin UFS      f1  SpeedStor
12  Compaq diagnost 5c  Priam Edisk     a9  NetBSD          f4  SpeedStor
14  Hidden FAT16 <3 61  SpeedStor       ab  Darwin boot     f2  DOS
```

```
secondary
16  Hidden FAT16    63  GNU HURD or Sys af  HFS / HFS+      fb  VMware VMFS
17  Hidden HPFS/NTF 64  Novell Netware  b7  BSDI fs         fc  VMware
VMKCORE
18  AST SmartSleep  65  Novell Netware  b8  BSDI swap       fd  Linux raid
auto
1b  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid fe  LANstep
1c  Hidden W95 FAT3 75  PC/IX           be  Solaris boot    ff  BBT
1e  Hidden W95 FAT1 80  Old Minix
Hex code (type L to list all codes): 8e
Changed type of partition 'Linux' to 'Linux LVM'

Command (m for help): p

Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x610c7c18

   Device Boot      Start         End      Blocks   Id  System
/dev/vdb1            2048    20971519    10484736   8e  Linux LVM

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

3.1.3. Optionally verify that the partition has been created:

```
# fdisk -l /dev/vdb

Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x610c7c18

   Device Boot      Start         End      Blocks   Id  System
/dev/vdb1            2048    20971519    10484736   8e  Linux LVM
```

3.1.4. Inform the OS of partition table changes:

```
# partprobe
```

3.1.5. Create a physical volume from the partition that will be used within LVM:

```
# pvcreate /dev/vdb1
  Physical volume "/dev/vdb1" successfully created.
```

3.1.6. Create a volume group "data_vg" from the physical volume:

```
# vgcreate data_vg /dev/vdb1
  Volume group "data_vg" successfully created
```

3.1.7. Create a 2 GB logical volume "data1_lv" in the volume group:

```
# lvcreate -n data1_lv -L 2G data_vg
  Logical volume "data1_lv" created.
```

3.1.8. Format the logical volume with the XFS file system:

```
# mkfs.xfs /dev/data_vg/data1_lv
meta-data=/dev/data_vg/data1_lv  isize=512    agcount=4, agsize=131072 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=524288, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

3.1.9. Create a mount point for the file system:

```
# mkdir /data1
```

3.1.10. Edit /etc/fstab to make the file system available permanently:

```
# echo '/dev/mapper/data_vg-data1_lv /data1 xfs defaults 0 0' >> /etc/fstab
```

3.1.11. Update systemd with the new /etc/fstab configuration (RHEL/CentOS 7/8):

```
# systemctl daemon-reload
```

3.1.12. Mount the file system:

```
# mount /data1
```

3.1.13. Verify that the new file system is mounted:

```
# df -h /data1
Filesystem                     Size  Used Avail Use% Mounted on
/dev/mapper/data_vg-data1_lv  2.0G   33M  2.0G   2% /data1
```

3.2. Creating multiple file systems

3.2.1. Create logical volumes, file systems and mount points of the specified parameters:

```
# echo "data2_lv /data2 xfs 2G data_vg
data3_lv /data3 xfs 2G data_vg
data4_lv /data4 xfs 2G data_vg
data5_lv /data5 xfs 2G data_vg" | while read a b c d e; do lvcreate -n $a -L
$d $e; mkfs.xfs /dev/mapper/${e}-$a; mkdir -p $b; echo "/dev/mapper/${e}-$a
$b $c defaults 0 0" >> /etc/fstab; done
```

3.2.2. Update systemd with the new /etc/fstab configuration (RHEL/CentOS 7/8):

```
# systemctl daemon-reload
```

3.2.3. Mount the file systems:

```
# mount -a
```

3.2.4. Verify that the new file systems are mounted:

```
# df -h /data*
Filesystem                     Size  Used Avail Use% Mounted on
/dev/mapper/data_vg-data1_lv  2.0G   33M  2.0G   2% /data1
/dev/mapper/data_vg-data2_lv  2.0G   33M  2.0G   2% /data2
/dev/mapper/data_vg-data3_lv  2.0G   33M  2.0G   2% /data3
/dev/mapper/data_vg-data4_lv  2.0G   33M  2.0G   2% /data4
/dev/mapper/data_vg-data5_lv  2.0G   33M  2.0G   2% /data5
```

---

4. Extending a file system within LVM

4.1. After adding a disk

4.1.1. Verify if there is enough free space to extend the logical volume in the volume group ("data_vg"):

```
# vgs
  VG      #PV #LV #SN Attr   VSize   VFree
  data_vg   1   5   0 wz--n- <10.00g    0
  vg00      1   6   0 wz--n-  <9.30g    0
```

4.1.2. If there is not enough space in the volume group, add a new disk.

4.1.2.1. Find out the type of the logical volume (pay special attention to the "striped" or "mirrored" type):

```
# lvs -a -o segtype,devices,lv_name,vg_name | grep data5_lv
  linear /dev/vdb1(2048) data5_lv data_vg
```

4.1.2.2. If a SCSI disk has been added and has not been automatically detected by the system, force it to load:

```
# rescan-scsi-bus.sh -a
```

or

```
# for host in $(ls -1d /sys/class/scsi_host/*); do echo "- - -" >
${host}/scan; done
```

4.1.2.3. Display information about the added disk ("vdc"):

```
# lsblk
NAME                 MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0                   11:0    1 1024M  0 rom
vda                  252:0    0  9.8G  0 disk
├─vda1               252:1    0  512M  0 part /boot
└─vda2               252:2    0  9.3G  0 part
  ├─vg00-root_lv     253:0    0    1G  0 lvm  /
  ├─vg00-swap_lv     253:1    0    1G  0 lvm  [SWAP]
  ├─vg00-usr_lv      253:2    0  4.8G  0 lvm  /usr
  ├─vg00-var_lv      253:3    0    1G  0 lvm  /var
  ├─vg00-tmp_lv      253:4    0    1G  0 lvm  /tmp
  └─vg00-home_lv     253:5    0  512M  0 lvm  /home
vdb                  252:16   0   10G  0 disk
└─vdb1               252:17   0   10G  0 part
  ├─data_vg-data1_lv 253:6    0    2G  0 lvm  /data1
  ├─data_vg-data2_lv 253:7    0    2G  0 lvm  /data2
  ├─data_vg-data3_lv 253:8    0    2G  0 lvm  /data3
  ├─data_vg-data4_lv 253:9    0    2G  0 lvm  /data4
  └─data_vg-data5_lv 253:10   0    2G  0 lvm  /data5
vdc                  252:32   0   10G  0 disk
```

4.1.2.4. Create a physical volume from the disk that will be used within LVM:

```
# pvcreate /dev/vdc
  Physical volume "/dev/vdc" successfully created.
```

4.1.2.5. Add the physical volume to the specified volume group ("data_vg"):

```
# vgextend data_vg /dev/vdc
  Volume group "data_vg" successfully extended

# vgs
  VG      #PV #LV #SN Attr   VSize  VFree
  data_vg   2   5   0 wz--n- 20.00g <10.00g
  vg00      1   6   0 wz--n- <9.30g      0
```

4.1.3. Extend the logical volume with the "/data5" file system by 5 GB:

```
# lvresize -L +5G data_vg/data5_lv
  Size of logical volume data_vg/data5_lv changed from <2.00 GiB (511
extents) to <7.00 GiB (1791 extents).
  Logical volume data_vg/data5_lv successfully resized.

# xfs_growfs /dev/data_vg/data5_lv
meta-data=/dev/mapper/data_vg-data5_lv isize=512    agcount=4, agsize=130816
blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0 spinodes=0
data     =                       bsize=4096   blocks=523264, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal               bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
data blocks changed from 523264 to 1833984
```

or

```
# lvresize -rL +5G data_vg/data5_lv
```

4.1.4. Verify that the file system size has changed:

```
# df -h /data5
Filesystem                     Size  Used Avail Use% Mounted on
/dev/mapper/data_vg-data5_lv  7.0G   33M  7.0G   1% /data5
```

4.2. After increasing a disk capacity

4.2.1. Verify if there is enough free space to extend the logical volume in the volume group ("data_vg"):

```
# vgs
  VG      #PV #LV #SN Attr   VSize   VFree
  data_vg   1   5   0 wz--n- <10.00g    0
  vg00      1   6   0 wz--n-  <9.30g    0
```

4.2.2. If there is not enough space in the volume group, increase the disk capacity.

4.2.2.1. If a SCSI disk capacity has been increased and has not been automatically detected by the system, force it to load:

```
# rescan-scsi-bus.sh
```

or

```
# for disk in $(ls -1d /sys/class/scsi_disk/*); do echo "1" >
${disk}/device/rescan; done
```

4.2.2.2. Display information about the increased disk ("vdb"):

```
# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0                  11:0    1 1024M  0 rom
vda                 252:0    0 10.8G  0 disk
├─vda1              252:1    0  512M  0 part /boot
└─vda2              252:2    0  9.3G  0 part
  ├─vg00-root_lv    253:0    0    1G  0 lvm  /
  ├─vg00-swap_lv    253:1    0    1G  0 lvm  [SWAP]
  ├─vg00-usr_lv     253:2    0  4.8G  0 lvm  /usr
  ├─vg00-var_lv     253:5    0    1G  0 lvm  /var
  ├─vg00-tmp_lv     253:9    0    1G  0 lvm  /tmp
  └─vg00-home_lv    253:10   0  512M  0 lvm  /home
vdb                 252:16   0   15G  0 disk
└─vdb1              252:17   0   10G  0 part
  ├─data_vg-data1_lv 253:3    0    2G  0 lvm  /data1
  ├─data_vg-data2_lv 253:4    0    2G  0 lvm  /data2
  ├─data_vg-data3_lv 253:6    0    2G  0 lvm  /data3
  ├─data_vg-data4_lv 253:7    0    2G  0 lvm  /data4
  └─data_vg-data5_lv 253:8    0    2G  0 lvm  /data5
```

4.2.2.3. Increase the capacity of the partition ("vdb1") by all the free space on the disk:

```
# parted /dev/vdb resizepart 1 100%
```

```
Information: You may need to update /etc/fstab.
```

4.2.2.4. Verify that the partition ("vdb1") has been increased:

```
# lsblk
NAME                  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0                    11:0    1 1024M  0 rom
vda                   252:0    0 10.8G  0 disk
├─vda1                252:1    0  512M  0 part /boot
└─vda2                252:2    0  9.3G  0 part
  ├─vg00-root_lv      253:0    0    1G  0 lvm  /
  ├─vg00-swap_lv      253:1    0    1G  0 lvm  [SWAP]
  ├─vg00-usr_lv       253:2    0  4.8G  0 lvm  /usr
  ├─vg00-var_lv       253:5    0    1G  0 lvm  /var
  ├─vg00-tmp_lv       253:9    0    1G  0 lvm  /tmp
  └─vg00-home_lv      253:10   0  512M  0 lvm  /home
vdb                   252:16   0   15G  0 disk
└─vdb1                252:17   0   15G  0 part
  ├─data_vg-data1_lv 253:3    0    2G  0 lvm  /data1
  ├─data_vg-data2_lv 253:4    0    2G  0 lvm  /data2
  ├─data_vg-data3_lv 253:6    0    2G  0 lvm  /data3
  ├─data_vg-data4_lv 253:7    0    2G  0 lvm  /data4
  └─data_vg-data5_lv 253:8    0    2G  0 lvm  /data5
```

4.2.2.5. Inform the OS of partition table changes:

```
# partprobe
```

4.2.2.6. Display the capacity of physical volumes:

```
# pvs
  PV         VG      Fmt  Attr PSize   PFree
  /dev/vda2  vg00    lvm2 a--   <9.30g    0
  /dev/vdb1  data_vg lvm2 a--  <10.00g    0
```

4.2.2.7. Increase the capacity of the physical volume ("vdb1") by all the free space on the partition:

```
# pvresize /dev/vdb1
  Physical volume "/dev/vdb1" changed
  1 physical volume(s) resized or updated / 0 physical volume(s) not resized
```

4.2.2.8. Verify that the physical volume ("vdb1") has been increased:

```
# pvs
  PV         VG      Fmt  Attr PSize   PFree
  /dev/vda2  vg00    lvm2 a--   <9.30g    0
```

```
  /dev/vdb1  data_vg lvm2 a--  <15.00g 5.00g
# vgs
  VG      #PV #LV #SN Attr   VSize   VFree
  data_vg   1   5   0 wz--n- <15.00g 5.00g
  vg00      1   6   0 wz--n-  <9.30g    0
```

4.2.3. Extend the logical volume with the "/data5" file system by all the free space in the volume group:

```
# lvresize -l +100%FREE data_vg/data5_lv
  Size of logical volume data_vg/data5_lv changed from <2.00 GiB (511
extents) to <7.00 GiB (1791 extents).
  Logical volume data_vg/data5_lv successfully resized.

# xfs_growfs /dev/data_vg/data5_lv
meta-data=/dev/mapper/data_vg-data5_lv isize=512    agcount=4, agsize=130816
blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0 spinodes=0
data     =                       bsize=4096   blocks=523264, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal               bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
data blocks changed from 523264 to 1833984
```

or

```
# lvresize -rl +100%FREE data_vg/data5_lv
```

4.2.4. Verify that the file system size has changed:

```
# df -h /data5
Filesystem                     Size  Used Avail Use% Mounted on
/dev/mapper/data_vg-data5_lv  7.0G   33M  7.0G   1% /data5
```

---

5. Removing a file system within LVM

5.1. Umount the "/data5" file system:

```
# umount /data5
```

5.2. Remove the mount point:

```
# rmdir /data5
```

5.3. Deactivate the logical volume:

```
# lvchange -a n data_vg/data5_lv
```

5.4. Remove the logical volume including the file system:

```
# lvremove data_vg/data5_lv
  Logical volume "data5_lv" successfully removed
```

5.5. Remove the entry from /etc/fstab:

```
# sed -i '/data5_lv/d' /etc/fstab
```

5.6. Update systemd with the new /etc/fstab configuration (RHEL/CentOS 7/8):

```
# systemctl daemon-reload
```

5.7. Verify that the file system does not exist:

```
# df -h /data5
df: ‘/data5’: No such file or directory
```

5.8. Optionally remove the unused disk from the system:

5.8.1. Remove the physical volume from the volume group (the physical volume must be empty):

```
# pvs
  PV         VG      Fmt  Attr PSize   PFree
  /dev/vda2  vg00    lvm2 a--   <9.30g      0
  /dev/vdb1  data_vg lvm2 a--  <10.00g  <2.00g
  /dev/vdc   data_vg lvm2 a--  <10.00g <10.00g
# vgreduce data_vg /dev/vdc
  Removed "/dev/vdc" from volume group "data_vg"
```

5.8.2. Remove the physical volume from LVM:

```
# pvremove /dev/vdc
  Labels on physical volume "/dev/vdc" successfully wiped.
```

5.8.3. Optionally remove the SCSI disk from the system:

```
# echo 1 > /sys/block/vdc/device/delete
# lsblk /dev/vdc
lsblk: /dev/vdc: not a block device
```

---

6. Restoring a file system within LVM

6.1. Create some data in the file system (for the example purpose):

```
# ls -l /data4
total 0
# echo "This is just the beginning." > /data4/test
# ls -l /data4
total 4
-rw-r--r--. 1 root root 28 Feb 15 23:06 test
```

6.2. Accidentally remove the file system including the logical volume:

```
# umount /data4
# rmdir /data4
# lvremove data_vg/data4_lv
Do you really want to remove active logical volume data_vg/data4_lv? [y/n]:
y
  Logical volume "data4_lv" successfully removed
```

6.3. Examine the archive files for the "data_vg" volume group and locate the one that has a description of "Created *before* executing 'lvremove data_vg/data4_lv'":

```
# vgcfgrestore -l data_vg
...

  File:     /etc/lvm/archive/data_vg_00030-1155990744.vg
  VG name:      data_vg
  Description:  Created *before* executing 'lvremove data_vg/data4_lv'
  Backup Time:  Mon Feb 15 23:09:18 2021

  File:     /etc/lvm/backup/data_vg
  VG name:      data_vg
  Description:  Created *after* executing 'lvremove data_vg/data4_lv'
  Backup Time:  Mon Feb 15 23:09:18 2021
```

6.4. Using the archive file, revert the operation "lvremove data_vg/data4_lv":

```
# vgcfgrestore -f /etc/lvm/archive/data_vg_00030-1155990744.vg data_vg
  Volume group data_vg has active volume: data1_lv.
  Volume group data_vg has active volume: data3_lv.
```

```
  Volume group data_vg has active volume: data2_lv.
  WARNING: Found 3 active volume(s) in volume group "data_vg".
  Restoring VG with active LVs, may cause mismatch with its metadata.
Do you really want to proceed with restore of volume group "data_vg", while
3 volume(s) are active? [y/n]: n
  Restore aborted.

# umount /data*
# lvchange -a n data_vg/data{1..3}_lv
# vgcfgrestore -f /etc/lvm/archive/data_vg_00030-1155990744.vg data_vg
  Restored volume group data_vg
```

6.5. Recreate the mount point and reactivate the logical volumes:

```
# mkdir /data4
# lvchange -a y data_vg/data{1..4}_lv
# mount -a
# lvs
  LV       VG      Attr       LSize   Pool Origin Data%  Meta%  Move Log
Cpy%Sync Convert
  data1_lv data_vg -wi-ao----   2.00g
  data2_lv data_vg -wi-ao----   2.00g
  data3_lv data_vg -wi-ao----   2.00g
  data4_lv data_vg -wi-ao----   2.00g
...
```

6.6. Verify the contents of the previously removed file system:

```
# ls -l /data4
total 4
-rw-r--r--. 1 root root 28 Feb 15 23:06 test
# cat /data4/test
This is just the beginning.
```

---

7. Managing file systems on a multipath device

7.1. Configuring device mapper multipath (FC SAN)

7.1.1. List the fibre channel host bus adapters (HBA):

```
# lspci | grep -i fibre
4b:00.0 Fibre Channel: Emulex Corporation LPe35000/LPe36000 Series 32Gb/64Gb
Fibre Channel Adapter (rev 10)
4b:00.1 Fibre Channel: Emulex Corporation LPe35000/LPe36000 Series 32Gb/64Gb
Fibre Channel Adapter (rev 10)
98:00.0 Fibre Channel: Emulex Corporation LPe35000/LPe36000 Series 32Gb/64Gb
```

```
Fibre Channel Adapter (rev 10)
98:00.1 Fibre Channel: Emulex Corporation LPe35000/LPe36000 Series 32Gb/64Gb
Fibre Channel Adapter (rev 10)
```

7.1.2. Check the port status of the fibre channel host bus adapters:

```
# grep -v "xyz" /sys/class/fc_host/host*/port_state
/sys/class/fc_host/host15/port_state:Online
/sys/class/fc_host/host16/port_state:Linkdown
/sys/class/fc_host/host17/port_state:Online
/sys/class/fc_host/host18/port_state:Linkdown
```

7.1.3. Print the world wide port name (WWPN) of the fibre channel host bus adapters needed for the storage request:

```
# grep -v "xyz" /sys/class/fc_host/host*/port_name
/sys/class/fc_host/host15/port_name:0x100000109bef6c75
/sys/class/fc_host/host16/port_name:0x100000109bef6c76
/sys/class/fc_host/host17/port_name:0x100000109bef6cb1
/sys/class/fc_host/host18/port_name:0x100000109bef6cb2
```

7.1.4. Verify that a new disk (LUN) has been added from the storage (SAN):

```
# lsblk
NAME                        MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                           8:0    0  1.8T  0 disk
├─sda1                        8:1    0  256M  0 part  /boot/efi
├─sda2                        8:2    0    1G  0 part  /boot
└─sda3                        8:3    0 68.8G  0 part
  ├─vg00-root_lv            253:0    0    3G  0 lvm   /
  ├─vg00-swap_lv            253:1    0    4G  0 lvm   [SWAP]
  ├─vg00-usr_lv             253:2    0   10G  0 lvm   /usr
  ├─vg00-home_lv            253:3    0    3G  0 lvm   /home
  ├─vg00-var_lv             253:4    0    5G  0 lvm   /var
  ├─vg00-tmp_lv             253:5    0    5G  0 lvm   /tmp
  └─vg00-opt_lv             253:6    0    2G  0 lvm   /opt
sdb                           8:16   0    1T  0 disk
sdc                           8:32   0    1T  0 disk
sdd                           8:48   0    1T  0 disk
sde                           8:64   0    1T  0 disk

# lsscsi -suvw
[0:0:68:0]   enclosu 300705b011bc7cc0
-               -
  dir: /sys/bus/scsi/devices/0:0:68:0
[/sys/devices/pci0000:30/0000:30:02.0/0000:31:00.0/host0/target0:0:68/0:0:68
:0]
[0:2:0:0]    disk    600605b011bc7cc029fa6b79195b44d5
```

```
0x600605b011bc7cc029fa6b79195b44d5  /dev/sda   1.91TB
  dir: /sys/bus/scsi/devices/0:2:0:0
[/sys/devices/pci0000:30/0000:30:02.0/0000:31:00.0/host0/target0:2:0/0:2:0:0
]
[15:0:12:0]  disk    600507640081019560000000000000239
/dev/sdb   1.09TB
  dir: /sys/bus/scsi/devices/15:0:12:0
[/sys/devices/pci0000:4a/0000:4a:02.0/0000:4b:00.0/host15/rport-15:0-21/targ
et15:0:12/15:0:12:0]
[15:0:13:0]  disk    600507640081019560000000000000239
/dev/sdc   1.09TB
  dir: /sys/bus/scsi/devices/15:0:13:0
[/sys/devices/pci0000:4a/0000:4a:02.0/0000:4b:00.0/host15/rport-15:0-22/targ
et15:0:13/15:0:13:0]
[17:0:12:0]  disk    600507640081019560000000000000239
/dev/sdd   1.09TB
  dir: /sys/bus/scsi/devices/17:0:12:0
[/sys/devices/pci0000:97/0000:97:02.0/0000:98:00.0/host17/rport-17:0-21/targ
et17:0:12/17:0:12:0]
[17:0:13:0]  disk    600507640081019560000000000000239
/dev/sde   1.09TB
  dir: /sys/bus/scsi/devices/17:0:13:0
[/sys/devices/pci0000:97/0000:97:02.0/0000:98:00.0/host17/rport-17:0-22/targ
et17:0:13/17:0:13:0]
```

7.1.5. If the disk has been added but has not been automatically detected by the system, force it to load:

```
# rescan-scsi-bus.sh -a
```

or

```
# for host in $(ls -1d /sys/class/scsi_host/*); do echo "- - -" >
${host}/scan; done
```

7.1.6. Install "device-mapper-multipath":

```
# yum install -y device-mapper-multipath
```

7.1.7. Enable the multipath configuration:

```
# mpathconf --enable
```

7.1.8. Display the multipath configuration:

```
# mpathconf
multipath is enabled
```

```
find_multipaths is yes
user_friendly_names is enabled
default property blacklist is enabled
enable_foreign is set (no foreign multipath devices will be shown)
dm_multipath module is loaded
multipathd is not running
```

7.1.9. Optionally edit the multipath configuration:

```
# vi /etc/multipath.conf
```

7.1.10. Enable and start the "multipathd" service:

```
# systemctl enable multipathd; systemctl start multipathd
```

7.1.11. Verify that the multipath device "mpatha" is available:

```
# lsblk
NAME                      MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                         8:0    0  1.8T  0 disk
├─sda1                      8:1    0  256M  0 part  /boot/efi
├─sda2                      8:2    0    1G  0 part  /boot
└─sda3                      8:3    0 68.8G  0 part
  ├─vg00-root_lv          253:0    0    3G  0 lvm   /
  ├─vg00-swap_lv          253:1    0    4G  0 lvm   [SWAP]
  ├─vg00-usr_lv           253:2    0   10G  0 lvm   /usr
  ├─vg00-home_lv          253:3    0    3G  0 lvm   /home
  ├─vg00-var_lv           253:4    0    5G  0 lvm   /var
  ├─vg00-tmp_lv           253:5    0    5G  0 lvm   /tmp
  └─vg00-opt_lv           253:6    0    2G  0 lvm   /opt
sdb                         8:16   0    1T  0 disk
└─mpatha                  253:13   0    1T  0 mpath
sdc                         8:32   0    1T  0 disk
└─mpatha                  253:13   0    1T  0 mpath
sdd                         8:48   0    1T  0 disk
└─mpatha                  253:13   0    1T  0 mpath
sde                         8:64   0    1T  0 disk
└─mpatha                  253:13   0    1T  0 mpath

# multipath -ll
mpatha (360050764008101956000000000000239) dm-13 IBM,2145
size=1.0T features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 15:0:13:0 sdc 8:32 active ready running
| `- 17:0:12:0 sdd 8:48 active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 15:0:12:0 sdb 8:16 active ready running
  `- 17:0:13:0 sde 8:64 active ready running
```

7.2. Creating a file system on the multipath device

7.2.1. Create a primary partition of type "Linux LVM" on the multipath device (or continue directly with the "pvcreate" command below and use the entire disk):

```
# fdisk /dev/mapper/mpatha

Welcome to fdisk (util-linux 2.32.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xaa4e142f.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-2147483647, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-2147483647, default
2147483647): +600G

Created a new partition 1 of type 'Linux' and of size 600 GiB.

Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): L

 0  Empty           24  NEC DOS         81  Minix / old Lin bf  Solaris
 1  FAT12           27  Hidden NTFS Win 82  Linux swap / So c1  DRDOS/sec
(FAT-
 2  XENIX root      39  Plan 9          83  Linux           c4  DRDOS/sec
(FAT-
 3  XENIX usr       3c  PartitionMagic  84  OS/2 hidden or  c6  DRDOS/sec
(FAT-
 4  FAT16 <32M      40  Venix 80286     85  Linux extended  c7  Syrinx
 5  Extended        41  PPC PReP Boot   86  NTFS volume set da  Non-FS data
 6  FAT16           42  SFS             87  NTFS volume set db  CP/M / CTOS
/ .
 7  HPFS/NTFS/exFAT 4d  QNX4.x          88  Linux plaintext de  Dell Utility
 8  AIX             4e  QNX4.x 2nd part 8e  Linux LVM       df  BootIt
 9  AIX bootable    4f  QNX4.x 3rd part 93  Amoeba          e1  DOS access
 a  OS/2 Boot Manag 50  OnTrack DM      94  Amoeba BBT      e3  DOS R/O
 b  W95 FAT32       51  OnTrack DM6 Aux 9f  BSD/OS          e4  SpeedStor
 c  W95 FAT32 (LBA) 52  CP/M            a0  IBM Thinkpad hi ea  Rufus
alignment
 e  W95 FAT16 (LBA) 53  OnTrack DM6 Aux a5  FreeBSD         eb  BeOS fs
 f  W95 Ext'd (LBA) 54  OnTrackDM6      a6  OpenBSD         ee  GPT
```

```
10  OPUS            55  EZ-Drive        a7  NeXTSTEP        ef  EFI
(FAT-12/16/
11  Hidden FAT12    56  Golden Bow      a8  Darwin UFS      f0  Linux/PA-
RISC b
12  Compaq diagnost 5c  Priam Edisk     a9  NetBSD          f1  SpeedStor
14  Hidden FAT16 <3 61  SpeedStor       ab  Darwin boot     f4  SpeedStor
16  Hidden FAT16    63  GNU HURD or Sys af  HFS / HFS+      f2  DOS
secondary
17  Hidden HPFS/NTF 64  Novell Netware  b7  BSDI fs         fb  VMware VMFS
18  AST SmartSleep  65  Novell Netware  b8  BSDI swap       fc  VMware
VMKCORE
1b  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid fd  Linux raid
auto
1c  Hidden W95 FAT3 75  PC/IX           bc  Acronis FAT32 L fe  LANstep
1e  Hidden W95 FAT1 80  Old Minix       be  Solaris boot    ff  BBT
Hex code (type L to list all codes): 8e
Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): p
Disk /dev/mapper/mpatha: 1 TiB, 1099511627776 bytes, 2147483648 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 32768 bytes / 32768 bytes
Disklabel type: dos
Disk identifier: 0xaa4e142f

Device                    Boot Start        End    Sectors  Size Id Type
/dev/mapper/mpatha-part1       2048 1258293247 1258291200  600G 8e Linux LVM

Command (m for help): w
The partition table has been altered.
Failed to add partition 1 to system: Invalid argument

The kernel still uses the old partitions. The new table will be used at the
next reboot.
Syncing disks.
```

(The error can be ignored. As an alternative use the "parted" command instead.)

7.2.2. Inform the OS of partition table changes:

```
# partprobe
```

7.2.3. Verify that a device mapper device for the new partition has been created:

```
# ls -l /dev/mapper/mpatha*
lrwxrwxrwx 1 root root 8 Aug  1 16:21 /dev/mapper/mpatha -> ../dm-13
lrwxrwxrwx 1 root root 8 Aug  1 16:12 /dev/mapper/mpatha1 -> ../dm-14

# fdisk -l /dev/mapper/mpatha
```

```
Disk /dev/mapper/mpatha: 1 TiB, 1099511627776 bytes, 2147483648 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 32768 bytes / 32768 bytes
Disklabel type: dos
Disk identifier: 0xaa4e142f

Device               Boot Start        End    Sectors  Size Id Type
/dev/mapper/mpatha1       2048 1258293247 1258291200  600G 8e Linux LVM

# lsblk
NAME                      MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                         8:0    0  1.8T  0 disk
├─sda1                      8:1    0  256M  0 part  /boot/efi
├─sda2                      8:2    0    1G  0 part  /boot
└─sda3                      8:3    0 68.8G  0 part
  ├─vg00-root_lv          253:0    0    3G  0 lvm   /
  ├─vg00-swap_lv          253:1    0    4G  0 lvm   [SWAP]
  ├─vg00-usr_lv           253:2    0   10G  0 lvm   /usr
  ├─vg00-home_lv          253:3    0    3G  0 lvm   /home
  ├─vg00-var_lv           253:4    0    5G  0 lvm   /var
  ├─vg00-tmp_lv           253:5    0    5G  0 lvm   /tmp
  └─vg00-opt_lv           253:6    0    2G  0 lvm   /opt
sdb                         8:16   0    1T  0 disk
├─sdb1                      8:17   0  600G  0 part
└─mpatha                  253:13   0    1T  0 mpath
  └─mpatha1               253:14   0  600G  0 part
sdc                         8:32   0    1T  0 disk
├─sdc1                      8:33   0  600G  0 part
└─mpatha                  253:13   0    1T  0 mpath
  └─mpatha1               253:14   0  600G  0 part
sdd                         8:48   0    1T  0 disk
├─sdd1                      8:49   0  600G  0 part
└─mpatha                  253:13   0    1T  0 mpath
  └─mpatha1               253:14   0  600G  0 part
sde                         8:64   0    1T  0 disk
├─sde1                      8:65   0  600G  0 part
└─mpatha                  253:13   0    1T  0 mpath
  └─mpatha1               253:14   0  600G  0 part
```

7.2.3.1. If not, create a device mapper device for the new partition:

```
# kpartx -a /dev/mapper/mpatha
```

7.2.4. Create a physical volume from the partition that will be used within LVM:

```
# pvcreate /dev/mapper/mpatha1
  Physical volume "/dev/mapper/mpatha1" successfully created.
```

7.2.5. Create a volume group "db_vg" from the physical volume:

```
# vgcreate db_vg /dev/mapper/mpatha1
  Volume group "db_vg" successfully created
```

7.2.6. Create a 125 GB logical volume "db01_lv" in the volume group:

```
# lvcreate -n db01_lv -L 125G db_vg
  Logical volume "db01_lv" created.
```

7.2.7. Format the logical volume with the XFS file system:

```
# mkfs.xfs /dev/db_vg/db01_lv
mkfs.xfs: Volume reports stripe unit of 32768 bytes and stripe width of 0,
ignoring.
meta-data=/dev/db_vg/db01_lv     isize=512    agcount=4, agsize=8192000 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=0 inobtcount=0
data     =                       bsize=4096   blocks=32768000, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=16000, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

7.2.8. Create a mount point for the file system:

```
# mkdir /db01
```

7.2.9. Edit /etc/fstab to make the file system available permanently:

```
# echo '/dev/mapper/db_vg-db01_lv /db01 xfs defaults 0 0' >> /etc/fstab
```

7.2.10. Update systemd with the new /etc/fstab configuration (RHEL/CentOS 7/8):

```
# systemctl daemon-reload
```

7.2.11. Mount the file system:

```
# mount /db01
```

7.2.12. Verify that the new file system is mounted:

```
# df -h /db01
Filesystem                                Size  Used Avail Use% Mounted on
/dev/mapper/db_vg-db01_lv                 125G  925M  125G   1% /db01
```

7.3. Extending a file system after resizing the physical storage device (LUN)

7.3.1. If a SCSI disk capacity has been increased and has not been automatically detected by the system, force it to load:

```
# rescan-scsi-bus.sh
```

or

```
# for disk in $(ls -1d /sys/class/scsi_disk/*); do echo "1" >
${disk}/device/rescan; done
```

7.3.2. Resize the multipath device:

```
# multipathd resize map /dev/mapper/mpatha
```

7.3.3. Display information about the increased multipath device:

```
# lsblk | grep -w mpatha
```

7.3.4. Extend the file system (including the corresponding partition, physical volume and logical volume) similarly as described from 4.2.2.3.

7.4. Removing a file system including the multipath device

7.4.1. Umount the "/db01" file system:

```
# umount /db01
```

7.4.2. Remove the mount point:

```
# rmdir /db01
```

7.4.3. Deactivate the logical volume:

```
# lvchange -a n db_vg/db01_lv
```

7.4.4. Remove the logical volume including the file system:

```
# lvremove db_vg/db01_lv
  Logical volume "db01_lv" successfully removed
```

7.4.5. Remove the entry from /etc/fstab:

```
# sed -i '/db01_lv/d' /etc/fstab
```

7.4.6. Update systemd with the new /etc/fstab configuration (RHEL/CentOS 7/8):

```
# systemctl daemon-reload
```

7.4.7. Remove the physical volume from the volume group (the physical volume must be empty):

```
# vgreduce db_vg /dev/mapper/mpatha1
  Removed "/dev/mapper/mpatha1" from volume group "db_vg"
```

7.4.8. Deactivate the volume group:

```
# vgchange -a n db_vg
```

7.4.9. Remove the volume group:

```
# vgremove db_vg
Volume group "db_vg" successfully removed
```

7.4.10. Remove the physical volume from LVM:

```
# pvremove /dev/mapper/mpatha1
  Labels on physical volume "/dev/mapper/mpatha1" successfully wiped.
```

7.4.11. Remove the unused multipath device map:

```
# multipath -f mpatha
```

7.4.12. Remove all paths to the multipath device:

```
# echo "sdb
sdc
sdd
sde" | while read dev; do echo 1 > /sys/block/${dev}/device/delete; done
```

8. Creating an encrypted file system

8.1. Encrypt the partition using LUKS and enter a passphrase (key):

```
# cryptsetup luksFormat /dev/vdb1

WARNING!
========
This will overwrite data on /dev/vdb1 irrevocably.

Are you sure? (Type uppercase yes): YES
Enter passphrase for /dev/vdb1:
Verify passphrase:
```

8.2. Decrypt the encrypted partition and maps it to the logical device-mapper device "secret":

```
# cryptsetup luksOpen /dev/vdb1 secret
Enter passphrase for /dev/vdb1:
```

8.3. Optionally verify that the mapped device exists:

```
# dmsetup ls --target crypt
secret  (253, 6)
```

8.4. Optionally add another passphrase (key) to the encrypted partition:

```
# cryptsetup luksAddKey /dev/vdb1
Enter any existing passphrase:
Enter new passphrase for key slot:
Verify passphrase:
```

8.5. Optionally verify that the key(s) exist(s) for the encrypted partition:

```
# cryptsetup luksDump /dev/vdb1
LUKS header information for /dev/vdb1

Version:        1
Cipher name:    aes
Cipher mode:    xts-plain64
Hash spec:      sha256
Payload offset: 4096
MK bits:        256
MK digest:      7a 89 b0 52 d3 0f 94 6c e0 e0 ea 86 ea 06 1c aa 40 66 7d e4
MK salt:        e6 82 0f fa 9c 1c 8e 76 0e a6 44 d0 76 1e 6e b6
```

```
                7e e3 33 8a 2b f4 ad 16 02 b7 e3 ed 5f 84 84 41
MK iterations:  28845
UUID:           aad9f193-a734-45e6-815a-a9029b03a020

Key Slot 0: ENABLED
    Iterations:             463150
    Salt:                   3f fc 35 b6 09 ff 2d 4f 56 df 1a 59 e3 64 f1 28
                              44 4e 1f 5d 08 24 ad a8 27 fc b0 5a a4 e4 da
24
    Key material offset:    8
    AF stripes:                4000
Key Slot 1: ENABLED
    Iterations:             482768
    Salt:                   21 4f 42 9e 87 d9 4f 03 3a 67 69 a1 e7 e7 66 82
                              01 7e 1c 8d 27 71 2c 6b be d5 f5 de 85 ed 80
63
    Key material offset:    264
    AF stripes:                4000
Key Slot 2: DISABLED
Key Slot 3: DISABLED
Key Slot 4: DISABLED
Key Slot 5: DISABLED
Key Slot 6: DISABLED
Key Slot 7: DISABLED
```

(The corresponding number of key slots is occupied.)

8.6. Format the partition with the XFS file system:

```
# mkfs.xfs /dev/mapper/secret
meta-data=/dev/mapper/secret     isize=512    agcount=4, agsize=610240 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=2440960, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

8.7. Create a mount point for the file system:

```
# mkdir /secret
```

8.8. Mount the file system:

```
# mount /dev/mapper/secret /secret
```

8.9. Optionally create a test file:

```
# touch /secret/test
# ls -l /secret/test
-rw-r--r--. 1 root root 0 Nov  3 12:34 /secret/test
```

8.10. When done, unmount the file system and unmap the encrypted partition:

```
# umount /secret
# cryptsetup luksClose secret
```

---

9. Creating a network file system (NFS)

9.1. Configuring NFS server

9.1.1. Install the NFS server/client package (if not installed already):

```
[root@nfsserver ~]# yum -y install nfs-utils
```

9.1.2. Enable and start the NFS services:

```
[root@nfsserver ~]# systemctl enable --now nfs-server rpcbind
Created symlink from /etc/systemd/system/multi-user.target.wants/nfs-
server.service to /usr/lib/systemd/system/nfs-server.service.
```

9.1.3. Create the directory "/nfsshare" to be shared by NFS clients:

```
[root@nfsserver ~]# mkdir /nfsshare
```

9.1.4. Create some data in the directory (for the example purpose):

```
[root@nfsserver ~]# echo "$(hostname)" > /nfsshare/test
```

9.1.5. Optionally associate the IP address of the NFS client with the hostname:

```
[root@nfsserver ~]# echo "192.168.124.254 nfsclient" >> /etc/hosts
```

9.1.6. Modify the /etc/exports file to export the "/nfsshare" directory to the NFS client, allowing read and write access with root privileges:

```
[root@nfsserver ~]# echo '/nfsshare nfsclient(rw,no_root_squash)'
>>/etc/exports
```

(The NFS clients can be specified as hostnames, IP addresses or networks, separated by spaces.)

9.1.7. Reload the /etc/exports configuration file:

```
[root@nfsserver ~]# exportfs -rv
exporting nfsclient:/nfsshare
```

9.1.8. Configure the firewall to allow access to the NFS service:

```
[root@nfsserver ~]# firewall-cmd --add-service={nfs,rpc-bind,mountd} --
permanent
success
[root@nfsserver ~]# firewall-cmd --reload
success
```

9.1.9. Optionally list the NFS services and their versions:

```
[root@nfsserver ~]# rpcinfo -p
   program vers proto   port  service
    100000    4   tcp    111  portmapper
    100000    3   tcp    111  portmapper
    100000    2   tcp    111  portmapper
    100000    4   udp    111  portmapper
    100000    3   udp    111  portmapper
    100000    2   udp    111  portmapper
    100005    1   udp  20048  mountd
    100005    1   tcp  20048  mountd
    100005    2   udp  20048  mountd
    100005    2   tcp  20048  mountd
    100024    1   udp  40636  status
    100005    3   udp  20048  mountd
    100024    1   tcp  55863  status
    100005    3   tcp  20048  mountd
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100227    3   tcp   2049  nfs_acl
    100003    3   udp   2049  nfs
    100003    4   udp   2049  nfs
    100227    3   udp   2049  nfs_acl
    100021    1   udp  52141  nlockmgr
    100021    3   udp  52141  nlockmgr
    100021    4   udp  52141  nlockmgr
    100021    1   tcp  38894  nlockmgr
    100021    3   tcp  38894  nlockmgr
    100021    4   tcp  38894  nlockmgr
```

9.2. Configuring NFS client

9.2.1. Install the NFS server/client package (if not installed already):

```
[root@nfsclient ~]# yum -y install nfs-utils
```

9.2.2. Create the mount point "/mnt/nfsshare":

```
[root@nfsclient ~]# mkdir /mnt/nfsshare
```

9.2.3. Optionally associate the IP address of the NFS server with the hostname:

```
[root@nfsclient ~]# echo "192.168.124.80 nfsserver" >> /etc/hosts
```

9.2.4. Edit the /etc/fstab to mount the exported "/nfsshare" on the "/mnt/nfsshare" permanently:

```
[root@nfsclient ~]# echo 'nfsserver:/nfsshare /mnt/nfsshare nfs defaults 0
0' >> /etc/fstab
```

9.2.5. Mount the exported "/nfsshare":

```
[root@nfsclient ~]# mount -a
```

9.2.6. Verify that the "/mnt/nfsshare" is writable by root on the NFS client:

```
[root@nfsclient ~]# echo "$(hostname)" >> /mnt/nfsshare/test
[root@nfsclient ~]# cat /mnt/nfsshare/test
nfsserver
nfsclient
```