**Create, mount, unmount, and use vfat, ext4 and xfs file systems.**

**Prerequisite**

To create a new partition called **lv\_vol** with a size of **100MB**, type:

# **lvcreate --size 100M --name lv\_vol /dev/vg**

**Ext4 File System**

To create an **ext4** file system (here called **/dev/vg/lv\_vol**), type:

# **mkfs.ext4 /dev/vg/lv\_vol**

mke2fs 1.42.9 (28-Dec-2013)

Filesystem label=

OS type: Linux

Block size=1024 (log=0)

Fragment size=1024 (log=0)

Stride=0 blocks, Stripe width=0 blocks

25688 inodes, 102400 blocks

5120 blocks (5.00%) reserved for the super user

First data block=1

Maximum filesystem blocks=33685504

13 block groups

8192 blocks per group, 8192 fragments per group

1976 inodes per group

Superblock backups stored on blocks:

8193, 24577, 40961, 57345, 73729

Allocating group tables: done

Writing inode tables: done

Creating journal (4096 blocks): done

Writing superblocks and filesystem accounting information: done

To mount this file system, type:

# **mount /dev/vg/lv\_vol /mnt**

To mount it permanently, edit the **/etc/fstab** file and add the following line:

**/dev/mapper/vg-lv\_vol /mnt ext4 defaults 1 2**

Note1: The last number (here **2**) is related to the **fsck** command: ‘**0**‘ means no **fsck** run at boot (very dangerous), ‘**1**‘ **fsck** is run first (root filesystem), ‘**2**‘ **fsck** is run just after the root filesystem. The second to last argument is in relation with the **dump** command (normally set at ‘**1**‘ for real filesystems, ‘**0**‘ for swap and NFS mounted filesystems).  
Note2: A best practice is to execute the **mount -a** command, each time you change something in the **/etc/fstab** file to detect any boot problem before it occurs.

To check an unmounted file system consistency, type:

# **fsck /dev/vg/lv\_vol**

fsck from util-linux 2.23.2

e2fsck 1.42.9 (28-Dec-2013)

/dev/mapper/vg-lv\_vol: clean, 11/25688 files, 8896/102400 block

To get details about a file system, type:

# **dumpe2fs /dev/vg/lv\_vol**

dumpe2fs 1.42.9 (28-Dec-2013)

Filesystem volume name:

Last mounted on:

Filesystem UUID: de13176f-6816-47e8-8742-2d543003d382

Filesystem magic number: 0xEF53

Filesystem revision #: 1 (dynamic)

Filesystem features: has\_journal ext\_attr resize\_inode dir\_index filetype needs\_recovery extent 64bit flex\_bg sparse\_super huge\_file uninit\_bg dir\_nlink extra\_isize

Filesystem flags: signed\_directory\_hash

Default mount options: user\_xattr acl

Filesystem state: clean

Errors behavior: Continue

Filesystem OS type: Linux

Inode count: 25688

Block count: 102400

Reserved block count: 5120

Free blocks: 93504

Free inodes: 25677

First block: 1

Block size: 1024

Fragment size: 1024

Group descriptor size: 64

Reserved GDT blocks: 256

Blocks per group: 8192

Fragments per group: 8192

Inodes per group: 1976

Inode blocks per group: 247

Flex block group size: 16

Filesystem created: Tue Jul 29 17:47:36 2014

Last mount time: Tue Jul 29 17:48:56 2014

Last write time: Tue Jul 29 17:48:56 2014

Mount count: 1

Maximum mount count: -1

Last checked: Tue Jul 29 17:47:36 2014

Check interval: 0 ()

Lifetime writes: 4447 kB

Reserved blocks uid: 0 (user root)

Reserved blocks gid: 0 (group root)

First inode: 11

Inode size: 128

Journal inode: 8

Default directory hash: half\_md4

Directory Hash Seed: 1780d5a3-72df-4cc4-ba19-4a6915c064e3

Journal backup: inode blocks

Journal features: journal\_64bit

Journal size: 4096k

Journal length: 4096

Journal sequence: 0x00000002

Journal start: 1

...

**Xfs File System**

To create an **xfs** file system (here called **/dev/vg/lv\_vol**), type:

# **mkfs.xfs /dev/vg/lv\_vol**

meta-data=/dev/vg/lv\_vol        isize=256    agcount=4, agsize=6400 blks

         =                       sectsz=512   attr=2, projid32bit=1

         =                       crc=0

data     =                       bsize=4096   blocks=25600, imaxpct=25

         =                       sunit=0      swidth=0 blks

naming   =version 2              bsize=4096   ascii-ci=0 ftype=0

log      =internal log           bsize=4096   blocks=853, version=2

         =                       sectsz=512   sunit=0 blks, lazy-count=1

realtime =none                   extsz=4096   blocks=0, rtextents=0

To mount this file system, type:

# **mount /dev/vg/lv\_vol /mnt**

To mount it permanently, edit the **/etc/fstab** file and add the following line:

**/dev/mapper/vg-lv\_vol /mnt xfs defaults 1 2**

Note: The last number (here **2**) is related to the **fsck** command: ‘**0**‘ means no **fsck** run at boot (very dangerous), ‘**1**‘ **fsck** is run first (root filesystem), ‘**2**‘ **fsck** is run just after the root filesystem. The second to last argument is in relation with the **dump** command (normally set at ‘**1**‘ for real filesystems, ‘**0**‘ for swap and NFS mounted filesystems).  
Note2: A best practice is to execute the **mount -a** command, each time you change something in the **/etc/fstab** file to detect any boot problem before it occurs.

To repair an unmounted file system consistency, type:

# **xfs\_repair /dev/vg/lv\_vol**

Phase 1 - find and verify superblock...

Phase 2 - using internal log

        - zero log...

        - scan filesystem freespace and inode maps...

        - found root inode chunk

Phase 3 - for each AG...

        - scan and clear agi unlinked lists...

        - process known inodes and perform inode discovery...

        - agno = 0

        - agno = 1

        - agno = 2

        - agno = 3

        - process newly discovered inodes...

Phase 4 - check for duplicate blocks...

        - setting up duplicate extent list...

        - check for inodes claiming duplicate blocks...

        - agno = 0

        - agno = 1

        - agno = 2

        - agno = 3

Phase 5 - rebuild AG headers and trees...

        - reset superblock...

Phase 6 - check inode connectivity...

        - resetting contents of realtime bitmap and summary inodes

        - traversing filesystem ...

        - traversal finished ...

        - moving disconnected inodes to lost+found ...

Phase 7 - verify and correct link counts...

done

To get details about a mounted file system, type:

# **xfs\_info /dev/vg/lv\_vol**

meta-data=/dev/mapper/vg-lv\_vol isize=256    agcount=4, agsize=6400 blks

         =                       sectsz=512   attr=2, projid32bit=1

         =                       crc=0

data     =                       bsize=4096   blocks=25600, imaxpct=25

         =                       sunit=0      swidth=0 blks

naming   =version 2              bsize=4096   ascii-ci=0 ftype=0

log      =internal               bsize=4096   blocks=853, version=2

         =                       sectsz=512   sunit=0 blks, lazy-count=1

realtime =none                   extsz=4096   blocks=0, rtextents=0

**Vfat File System**

To create an **vfat** file system (here called **/dev/vg/lv\_vol**), type:

# **mkfs.vfat /dev/vg/lv\_vol**

mkfs.fat 3.0.20 (12 Jun 2013)

unable to get drive geometry, using default 255/63

To mount this file system, type:

# **mount /dev/vg/lv\_vol /mnt**

To mount it permanently, edit the **/etc/fstab** file and add the following line:

**/dev/mapper/vg-lv\_vol /mnt vfat defaults 1 2**

To repair an unmounted file system consistency, type:

# **fsck.vfat /dev/vg/lv\_vol**

fsck.fat 3.0.20 (12 Jun 2013)

/dev/vg/lv\_vol: 0 files, 0/51091 clusters

**Useful Tip**

The time to create a filesystem has been dramatically reduced between **Ext3** and **Ext4** (by a factor of 100). However, although the creation time is now very quick, a process in the background continues to work for several minutes (according to the size of the filesystem)!

This process called **ext4lazyinit** creates the remaining index nodes which are used to reference leaf nodes, kind of pointers to data on the filesystem.

Although definitively not required for the **RHCSA** exam, you can create an **ext4** filesystem without this behavior by typing:

# **mkfs.ext4 -E lazy\_itable\_init=0,lazy\_journal\_init=0 /dev/vg/lv\_vol**