RHCEv8 Online Class 22052021 10:00pm
RHCSA-sysadmin
- Managing Logical Volumes-LVM
- Installing and Updating Software Packages
----

recap normal partition on Linux
order:
create **1.2gb** std partition with **xfs** filesystem and mount it persistently on **/mnt/disk1**

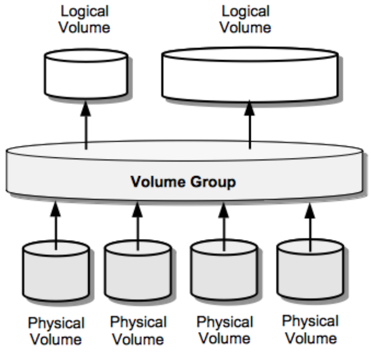
```
# cat /proc/partitions
# fdisk /dev/sdb
p
n
e
Partition number: Enter
First sector: Enter
Last sector: Enter
p
/dev/sdb1       2048 10485759 10483712   5G  5 Extended
n
Adding logical partition 5
First sector: Enter
Last sector: +1.2G
w
# udevadm settle
# fdisk -l /dev/sdb

# mkfs.xfs /dev/sdb5
# blkid
/dev/sdb5: UUID="3e096655-f2d3-4615-b957-d23916918c77" TYPE="xfs" PARTUUID="ea64c04a-05"
# mkdir /mnt/disk1
# echo "/dev/sdb5 /mnt/disk1 xfs defaults 0 0" >>/etc/fstab
or
# echo "UUID=3e096655-f2d3-4615-b957-d23916918c77 /mnt/disk1 xfs defaults 0 0" >>/etc/fstab
# mount -a
# df -hT
/dev/sdb5          xfs       1.2G   41M  1.2G   4% /mnt/disk1
```

Managing Logical Volumes
Logical Volume Management-LVM

-Logical volumes and logical volume management make it easier to manage disk space.
-If a file system that hosts a logical volume needs more space, it can be allocated to its logical volume from the free space in its volume group and the file system can be resized.

LVM Definitions

step1- prepare Physical devices
  Physical devices are the storage devices used to save data stored in a logical volume
step2- create Physical volumes (PVs)
  must initialize a device as a physical volume before using it in an LVM system
step3- create Volume groups (VGs)
  Volume groups are storage pools made up of one or more physical volumes
step4- create Logical volumes (LVs) partition
  Logical volumes are created from free physical extents in a volume group and provide the "storage" device used by applications, users, and the operating system.
step5- format, mount and use lvm's



Logical Volume
Management Components

Implement LVM on Linux

step1- prepare Physical devices

```
# fdisk /dev/sdc
p
n
e
Enter
Enter
Enter
p
/dev/sdc1       2048 10485759 10483712   5G  5 Extended
n
Adding logical partition 5
Enter
+2G
p
/dev/sdc1       2048 10485759 10483712   5G  5   Extended
/dev/sdc5       4096  4198399  4194304      2G 83 Linux
t
Partition number (1,5, default 5): Enter
8e
p
/dev/sdc1       2048 10485759 10483712   5G  5  Extended
/dev/sdc5       4096  4198399  4194304     2G 8e Linux LVM
w
# udevadm settle
```

step2- create Physical volumes (PVs)

```
# fdisk -l /dev/sdc
# pvcreate /dev/sdc5
# pvs
/dev/sdc5     lvm2 ---    2.00g 2.00g
# pvdisplay /dev/sdc5
```

step3- create Volume groups (VGs)

```
# vgcreate <vgname> <existing pvname>
# vgcreate vg1 /dev/sdc5
# vgs
vg1   1   0   0 wz--n-  <2.00g <2.00g
# pvs
/dev/sdc5  vg1 lvm2 a--   <2.00g <2.00g
# vgdisplay vg1
```

step4- create Logical volumes (LVs) partition

```
# lvcreate -n <lvname> -L +size{K,M,G,T,P} <vgname>
# lvcreate -n lv1 -L +600M vg1
-n name
-L size
-l pe number of block
# lvs
lv1  vg1 -wi-a----- 600.00m
# lvdisplay /dev/vg1/lv1
```

step5- format, mount and use lvm’s

```
# mkfs.xfs /dev/vg1/lv1
or
# mkfs.xfs /dev/mapper/vg1-lv1
# blkid
/dev/mapper/vg1-lv1: UUID="9a8d09b8-6dad-4f66-944a-b7a3f7da5625" TYPE="xfs"
# mkdir /mnt/lv1
# echo "/dev/mapper/vg1-lv1 /mnt/lv1 xfs defaults 0 0" >>/etc/fstab
# echo "/dev/vg1/lv1 /mnt/lv1  xfs  defaults 0 0" >>/etc/fstab
# echo "UUID=9a8d09b8-6dad-4f66-944a-b7a3f7da5625 /mnt/lv1 xfs defaults 0 0" >>/etc/fstab
# mount -a
# df -hT
/dev/mapper/vg1-lv1 xfs       595M   35M  561M   6% /mnt/lv1
```

Physical Extents-PE to create LV

what is Physical Extents-PE

Each physical volume is divided in to small chunks of data, known as **physical extents-pe**

ex:

create **740mb** lv partition based-on pe

```
# vgs
vg1   1   1   0 wz--n-  <2.00g 1.41g
# vgdisplay vg1
PE Size              4.00 MiB                   pe default block size
# bc
740/4
185                                             number of pe
# lvcreate -n lv2 -l 185 vg1
# lvs
lv1  vg1 -wi-ao---- 600.00m
lv2  vg1 -wi-a----- 740.00m
# mkfs.ext4 /dev/mapper/vg1-lv2
# mkdir /mnt/lv2
# blkid
/dev/mapper/vg1-lv2: UUID="d6490a55-4c1d-4c09-9247-a5d0065f1328" TYPE="ext4"
# echo "/dev/mapper/vg1-lv2 /mnt/lv2 ext4 defaults 0 0" >>/etc/fstab
# mount -a
# df -hT
/dev/mapper/vg1-lv2 ext4      713M  1.5M  660M   1% /mnt/lv2
```

create lv partition with vg left over size

```
# lvcreate -l 100%FREE vg1                        if don't put lv name machine will select random name
or
# lvcreate -n lv3  -l 100%FREE vg1
# pvs
/dev/sdc5  vg1 lvm2 a--   <2.00g    0
# vgs
vg1   1   3   0 wz--n-  <2.00g    0
# lvs
lv1   vg1 -wi-ao---- 600.00m
lv2   vg1 -wi-ao---- 740.00m
lvol0 vg1 -wi-a----- 704.00m
```

LVExtend

extend lv1 to 1.3gb   1.3-600=731.2Mb

extend lv2 to 2.1gb   2.1-740=1410.4Mb

step1- check vg for free space

```
# vgs
vg1   1   3   0 wz--n-  <2.00g    0
# fdisk /dev/sdc
p
/dev/sdc1       2048 10485759 10483712   5G  5  Extended
/dev/sdc5       4096  4198399  4194304     2G 8e Linux LVM
n
Adding logical partition 6
Enter
Enter
p
/dev/sdc1         2048 10485759 10483712      5G  5 Extended
/dev/sdc5         4096  4198399  4194304       2G 8e Linux LVM
/dev/sdc6      4200448 10485759  6285312   3G 83 Linux
t
Partition number (1,5,6, default 6): 6
Hex code (type L to list all codes): 8e
p
/dev/sdc1         2048 10485759 10483712      5G  5 Extended
/dev/sdc5         4096  4198399  4194304       2G 8e Linux LVM
/dev/sdc6      4200448 10485759  6285312   3G 8e Linux LVM
w
```

```
# udevadm settle
# pvcreate /dev/sdc6
# pvs
/dev/sdc5  vg1 lvm2 a--   <2.00g     0
/dev/sdc6        lvm2 ---   <3.00g <3.00g
```

VGExtend

```
# vgextend <existing vg>  <new pv>
# vgextend vg1 /dev/sdc6
# vgs
vg1   2   3   0 wz--n-   4.99g <3.00g
# pvs
/dev/sdc5  vg1 lvm2 a--   <2.00g     0
/dev/sdc6  vg1 lvm2 a--   <3.00g <3.00g
```

LVExtend

-size

```
# bc
1.3*1024
1331.2                ->total extend size
1331.2-600
731.2                 ->append size
2.1*1024
2150.4
2150.4-740
1410.4
```

-pe block

```
1.3*1024
1331.2                ->total extend size
1331.2/4
332                   ->total extend pe blocks
600/4
150                   ->existing pe block
332-150
182                   ->append pe block
2.1*1024
2150.4                ->total extend size
2150.4/4
537                   ->total extend pe blocks
740/4
185                   ->existing pe block
537-185
352                   ->append pe block
```

-size

```
# lvextend -L 1331.2M /dev/vg1/lv1                    ->extend with total size
or
# lvextend -L +731.2M  /dev/vg1/lv1                   ->extend with append size
```

-pe block

```
# lvextend -l 537 /dev/vg1/lv2                        ->extend with total pe block
or
# lvextend -l +352 /dev/vg1/lv2                       -> extend with append pe block
# lvs
lv1   vg1 -wi-ao----   1.30g
lv2   vg1 -wi-ao----  <2.10g
# df -hT
/dev/mapper/vg1-lv1 xfs        595M   35M  561M   6% /mnt/lv1
/dev/mapper/vg1-lv2 ext4      713M  1.5M  660M   1% /mnt/lv2
```

update kernel

-xfs

```
# xfs_growfs /dev/mapper/vg1-lv1
```

-ext2,3,4

```
# resize2fs /dev/mapper/vg1-lv2
# df -hT
/dev/mapper/vg1-lv1 xfs         1.3G   40M  1.3G   3% /mnt/lv1
/dev/mapper/vg1-lv2 ext4      2.1G  2.2M  2.0G   1% /mnt/lv2
```

DeleteLV

step1-take backup
step2-detach it from Linux
step3-delete lv

```
# umount /mnt/lv1
# umount /mnt/lv2
# vim /etc/fstab
remove lvm records
:wq!
# lvremove /dev/vg1/lv1
Do you really want to remove active logical volume vg1/lv1? [y/n]: yes
# lvremove /dev/vg1/lv2
Do you really want to remove active logical volume vg1/lv2? [y/n]: yes
# lvremove /dev/vg1/lvol0
Do you really want to remove active logical volume vg1/lvol0? [y/n]: yes
```

VGReduce

```
# vgs
vg1   2   1   0 wz--n-   4.99g 4.30g
# pvs
/dev/sdc5  vg1 lvm2 a--   <2.00g <1.31g
/dev/sdc6  vg1 lvm2 a--   <3.00g <3.00g
# vgreduce vg1 /dev/sdc5
# vgs
vg1   1   1   0 wz--n-   4.99g 4.30g
# pvs
/dev/sdc5          lvm2 a--   <2.00g <1.31g
/dev/sdc6  vg1 lvm2 a--   <3.00g <3.00g
```

DeleteVG

```
# vgremove vg1
# vgs
# pvs
/dev/sdc5          lvm2 a--   <2.00g <1.31g
/dev/sdc6          lvm2 a--   <3.00g <3.00g
```

DeletePV

```
# pvremove /dev/sdc5 /dev/sdc6
# pvs
```

Create VG with custom pe-size

pe default size is 4mb but it can be 8, 16 and 32mb too

```
# pvcreate /dev/sdc5
# pvs
/dev/sdc5      lvm2 a--    1.98g 1.98g
# vgcreate -s 16M vg1 /dev/sdc5
# pvs
/dev/sdc5  vg1 lvm2 a--    1.98g 1.98g
# vgdisplay vg1
PE Size              16.00 MiB
```

Installing and Updating Software Packages

what is package?

MS-Win application/software == Linux Packages

package architect

https://pkgs.org/

firefox-78.10.0-1.el8_3.x86_64.rpm

firefox	package name	
78.10.0	version	
1.el8_3	release	
x86_64	Arch	
	64bit(x86_64)	64bit Linux
	32bit(i686)	32bit Linux
	noarch	32/64 bit

```
# arch
# hostnamectl
# uname -m
# cat /etc/redhat-release
```

rpm	Extension
*.rmp	rhel based
*.deb	Debian based
*.sh, *.bin	scripts
*.tar	source code

how to install package on Linux?

1-rpm
2-yum

1-Red Hat Package Management-rpm

1-1-chceck
1-2-install

1-1-chceck

# rpm	-q	query
	-a	all
	-p	package
	-c	config file
	-d	doc files
	-l	list
	-i	info
	-f	from file
	-s	status
	-R	dependencies

NOTE: don't use options without 'q'

ex:
find out ssh package

```
# rpm -qa
# rpm -qa --last
# rpm -qa | wc -l
615
# rpm -qa ssh
# rpm -qa *ssh
# rpm -qa ssh*
# rpm -qa *ssh*
# rpm -qa | grep "ssh"
openssh-server-8.0p1-4.el8_1.x86_64
# rpm -qi openssh-server-8.0p1-4.el8_1.x86_64
# rpm -qc openssh-server-8.0p1-4.el8_1.x86_64
# rpm -qd openssh-server-8.0p1-4.el8_1.x86_64
# rpm -ql openssh-server-8.0p1-4.el8_1.x86_64
# rpm -qs openssh-server-8.0p1-4.el8_1.x86_64
# rpm -qR openssh-server-8.0p1-4.el8_1.x86_64
```

check package's script

-after Install

```
# rpm -q --scripts openssh-server-8.0p1-4.el8_1.x86_64
```

-before Install

```
# rpm -qp --scripts  zsh-5.5.1-6.el8_1.2.x86_64.rpm
```

package from file

find out **useradd** package

```
# which useradd
/usr/sbin/useradd rpm -qf /usr/sbin/useradd
shadow-utils-4.6-8.el8.x86_64
```

1-2-install

```
# rpm   -i       install
        -e       erase/remove
        -v       verbosity
        -U       update
        --force  forceful install
```

ex:

install **zsh** package on Linux

-online

find it out

https://centos.pkgs.org/

```
# wget http://mirror.centos.org/centos/8/BaseOS/x86_64/os/Packages/zsh-5.5.1-6.el8_1.2.x86_64.rpm
# ls
 zsh-5.5.1-6.el8_1.2.x86_64.rpm
# rpm -qip zsh-5.5.1-6.el8_1.2.x86_64.rpm
# rpm -qcp zsh-5.5.1-6.el8_1.2.x86_64.rpm
# rpm -qdp zsh-5.5.1-6.el8_1.2.x86_64.rpm
# rpm -qlp zsh-5.5.1-6.el8_1.2.x86_64.rpm
```

install

```
# rpm -iv zsh-5.5.1-6.el8_1.2.x86_64.rpm
```

remove

```
# rpm -qa | grep "zsh"
zsh-5.5.1-6.el8_1.2.x86_64
# rpm -ev zsh-5.5.1-6.el8_1.2.x86_64
```

offline

-attach iso image to virtual/physical host

```
# cat /proc/partitions
# mkdir /media/cdrom
```

temporary

```
# mount /dev/sr0 /media/cdrom/
```

persistent

```
# echo "/dev/sr0 /media/cdrom iso9660 defaults 0 0" >>/etc/fstab
# mount -a
# ls /media/cdrom
AppStream  BaseOS  EFI  images  isolinux  LICENSE  media.repo  TRANS.TBL
# ls /media/cdrom/AppStream/Packages/ | wc -l
5685
# ls /media/cdrom/BaseOS/Packages/ | wc -l
1695
```

CentOS8 has 2repository:

-AppStream

-BaseOS

```
# cp /media/cdrom/AppStream/Packages/zsh-html-5.5.1-6.el8_1.2.noarch.rpm /tmp/
# ls /tmp
# rpm -qp --scripts /tmp/zsh-html-5.5.1-6.el8_1.2.noarch.rpm
# rpm -qip /tmp/zsh-html-5.5.1-6.el8_1.2.noarch.rpm
# cp /media/cdrom/AppStream/Packages/httpd-2.4.37-30.module_el8.3.0+561+97fdbbcc.x86_64.rpm /tmp/
# rpm -qp --scripts /tmp/httpd-2.4.37-30.module_el8.3.0+561+97fdbbcc.x86_64.rpm
# rpm -qi /tmp/httpd-2.4.37-30.module_el8.3.0+561+97fdbbcc.x86_64.rpm
# rpm -iv /tmp/httpd-2.4.37-30.module_el8.3.0+561+97fdbbcc.x86_64.rpm
error: Failed dependencies
```

2-YellowDog Updater Manager-yum

released on rhel5 to exactly solve dependency problem.
yum has 2types on host:
1-yum server, provide yum repository
2-yum client, who will connect to yum server through http:// ftp:// nfs:// file://

how yum client connects to yum server

step1-need link

http://content.example.com/rhel8.0/x86_64/dvd/BaseOS
http://content.example.com/rhel8.0/x86_64/dvd/AppStream
ftp://
nfs://

step2-config yum file on client

```
# vim /etc/yum.repos.d/rhel.repo
[<name>]                  ->yum id
name=<name>               ->yum name
baseurl=http://           ->link access to yum server
gpgcheck=0                ->Gnu Privacy Guard 0/1
enabled=1                 ->enable/disable repository 0/1
:wq!
# yum repolist
```

what about host, who doesn't have access to lan/network
need to config local yum repository on single host
steps
1-load iso image
2-config local yum repository

1-load iso image

```
# cat /proc/partitions
# mkdir /media/cdrom
```

temporary

```
# mount /dev/sr0 /media/cdrom/
```

persistent

```
# echo "/dev/sr0 /media/cdrom iso9660 defaults 0 0" >>/etc/fstab
# mount -a
# ls /media/cdrom
AppStream  BaseOS  EFI  images  isolinux  LICENSE  media.repo  TRANS.TBL
# vim /etc/yum.repos.d/rhel.repo
[App]
name=AppStream
baseurl=file:///media/cdrom/AppStream/
gpgcheck=0
enabled=1

[Base]
name=BaseOS
baseurl=file:///media/cdrom/BaseOS/
gpgcheck=0
enabled=1
:wq!
# yum repolist
# yum list installed
# yum list | wc -l
# yum search apache
httpd.x86_64 : Apache HTTP Server
# yum list http*
# yum install httpd.x86_64 -y
# yum remove httpd.x86_64
# yum update httpd.x86_64
# yum history
# yum list openssh
# yum update openssh -y
```

Install 3rd party online repository

1-install epel

https://dl.fedoraproject.org/pub/epel/

```
# wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
# rpm -qip epel-release-latest-8.noarch.rpm
# rpm -iv epel-release-latest-8.noarch.rpm
```

or

```
# yum install epel* -y
```

How to install package from specific repository?

```
# yum list ansible
# yum --enablerepo=epel install ansible
# yum --enablerepo=<repo id> install ansible
```