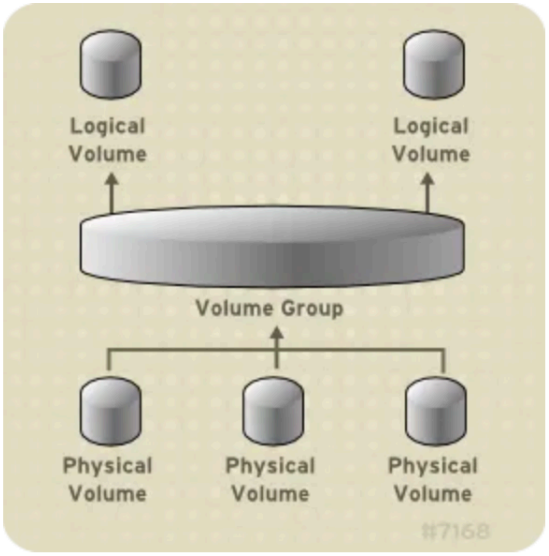



```
OBJECTIVE: SCORE
Manage basic networking: 100%
Understand and use essential tools: 10%
Operate running systems: 0%
Configure local storage: 25%
Create and configure file systems: 25%
Deploy, configure and maintain systems: 29%
Manage users and groups: 0%
Manage security: 0%
Manage containers: 0%
Create simple shell scripts: 0%
```

RHCSA

Kernel Package	Description
kernel	Contains no files, but ensures other kernel packages are accurately installed
kernel-core	Includes a minimal number of modules to provide core functionality
kernel-devel	Includes support for building kernel modules
kernel-modules	Contains modules for common hardware devices
kernel-modules-extra	Contains modules for not-so-common hardware devices
kernel-headers	Includes files to support the interface between the kernel and userspace libraries and programs
kernel-tools	Includes tools to manipulate the kernel
kernel-tools-libs	Includes the libraries to support the kernel tools

cat
cat
cat

groupadd -o -g 5000 dba
groupadd -o -g linuxadm
useradd user1000
usermod -aG dba user1000
/etc/group
/etc/passwd
groupmod -n sysadm linuxadm
groupmod -g 6000 sysadm
groupdel sysadm
/etc/group
visudo

groupadd -g 6000 lnxgroup
useradd -u 5000 -g 6000 user5000
passwd user5000
chage -m 4 -M 30 user5000
groupmod -g 4000 lnxgroup
`sudo groupmod -g 7000 lnxgrp`

When two groups have an identical GID, members of both groups get identical rights on each other's files.

Job scheduling and execution is taken care of by two service daemons: atd and crond.

While atd manages the jobs scheduled to run one time in the future, crond is responsible for running jobs repetitively at pre-specified times.

/var/spool/cron and /etc/cron.d directories

/etc directory for either service. These files are named at.allow and at.deny for the at service, and cron.allow and

cron.deny for the cron service.

Variable	Description
DISPLAY	Stores the hostname or IP address for graphical terminal sessions
HISTFILE	Defines the file for storing the history of executed commands
HISTSIZE	Defines the maximum size for the HISTFILE
HOME	Sets the home directory path
LOGNAME	Retains the login name
MAIL	Contains the path to the user mail directory
PATH	Defines a colon-separated list of directories to be searched when executing a command. A correct setting of this variable eliminates the need to specify the absolute path of a command to run it.
PPID	Holds the identifier number for the parent program
PS1	Defines the primary command prompt
PS2	Defines the secondary command prompt
PWD	Stores the current directory location
SHELL	Holds the absolute path to the primary shell file
TERM	Holds the terminal type value
UID	Holds the logged-in user's UID
USER	Retains the name of the logged-in user

dnf list installed kernel*

export and unset command. env/printenv will print env variables. command subs: \u@\h

1. Where does GRUB2 read its configuration from on a BIOS system?/boot/grub2/grub.cfg
2. How can the redhat-support-tool be used to search and display the same Knowledgebase content as on the Red Hat Customer Portal? Using the search command followed by keywords or error codes
3.How does file system metadata alignment impact the performance of striped arrays (RAID 0, RAID 4, RAID 5, RAID 6)? If the write request is wider than the strip size, I/O requests could require two writes per disk instead of one or having all metadata ends up on one disk, causing that disk to become a hot spot.
4. How does the spare area of SSDs impact their performance on random writes?
improvement
5. How does data striping in RAID increase throughput?
By dividing data into stripes and distributing them among several disks in the RAID array
6. What are the steps in a typical change management procedure for performance tuning changes? Set a baseline by running the test workload and gathering metrics. 2. Perform changes one at a time, measuring the effect after each change. 3. Verify the effectiveness of the change by rerunning the test workload. 4. Reverse the change and compare with the baseline. 5. Apply and document the definitive change.
7. What are the main ideas behind the USE Method in performance tuning?
Checking Utilization, Saturation, and Errors for user interactions
What is a drop-in file in systemd and how is it used to configure unit settings?

file in systemd that overrides or adds specific options for a unit, created by making a directory under /etc/systemd/system/ named after the unit with .d appended, and then creating .conf files in this directory. For example, enabling memory accounting for sshd.service can be done by creating a 20-accounting.conf file in the directory /etc/systemd/system/sshd.service.d/.

While the sticky bit is most commonly used with directories, it can also be set on files. On files, the sticky bit has an outdated and limited use. It was historically used to keep a file in memory after execution.

How can you enable CPU, memory, and block I/O accounting for a service or a slice in systemd?

Create a drop-in file under /etc/systemd/system/ with the desired unit or slice name and .d appended and include the CPUAccounting, MemoryAccounting, and BlockIOAccounting settings.

What are the advantages of using custom slices in systemd?

System resource granularity and distribution equality

chmod +t (add sticky bit) drwxrwxrwxt (t) at the end refers to sticky
chmod g-s /usr/bin/write -v (gid)

ncdu
sudo find / -type f -exec du -h {} + 2>/dev/null | sort -rh | head -20
sudo du -ah / 2>/dev/null | sort -rh | head -20

find /var/log -min -100 -exec file {} \;
find /usr -maxdepth -type d -name src
find /tmp -perm -u=r
find /tmp -type f -exec ls -ld {} \;
find /tmp -name *.txt -ok cp {} \;
locate .sh -n2
locate -S
setfacl -m u:user1:r a.txt
setfacl -dm u:user100:7,user200:rwx /tmp/prj

find /dev -type c -perm 660
useradd user1000
usermod -aG sgroup user1000

ps -efl
pidof rsyslogd
pgrep rsyslogd
ps -U user1
ps -G root
nice -n -10
renice 5 5572
can also renice from top command(type r and give pID)
cat /var/log/cron

*/1 * 1-10 3 * : Any day, in March, from 1 to 10th, every hour, every one minute.

Column	Description
UID	User ID or name of the process owner
PID	Process ID of the process
PPID	Process ID of the parent process
C	CPU utilization for the process
STIME	Process start date or time
TTY	The controlling terminal the process was started on. “Console” represents the system console and “?” represents a daemon process.
TIME	Aggregated execution time for a process
CMD	The command or program name

useradd user100
useradd user200
usermod -aG sgroup user100
usermod -aG sgroup user200
mkdir /sdir
chmod g+s /sdir
chmod o-t /tmp

tree -hapf
uname -snovpr
wc -l , -w, -c
rpm -qa (list all packages)
rpm -q perl (list perl packages)
rpm -qf /etc/passwd (see which package owns the file)
rpm -qf /etc/group
dnf install policycoreutils
dnf info policycoreutils
dnf deflist policycoreutils
rpm -qi setup
rpm sushi -ve (remove package)
rpm -qf /etc/chrony.conf (see where the package is coming from)
cd /tmp
rmp2cpio /mnt/baseos/package/chrony-3.3.e18.x86_64.rpm | cpio -imd
rpm2cpio
find . -name chrony.conf 9
rpm -K /tmp/chrony.. –no-signature (Use the MD5 checksum for verifying its integrity and the GNU Privacy Guard (GnuPG or GPG) public key signature for ensuring the credibility of its developer or publisher.
cp -r /tmp/chrony.. /etc/chrony
rpmkeys –import /etc/pki/rpm-gpg/rpm/gpg-key-redhat-release
rpmkeys -K /mnt/BaseOS/packages/zsh-3.3 (answer should be digests signatures ok)
rpm -q gpg-pubkey
rpm -qi <key> (view specific details)
rpm -Vf /etc/sysconfig
rpm -qi zlib
rpm -qa | sort

environment groups and package groups:

The environment groups available in RHEL 8 are server, server with GUI, minimal install, workstation, virtualization host, and custom operating system. These are listed on the software selection window during RHEL 8 installation. The package groups include container management, smart card support, security tools, system tools, network servers, etc. LOGS: **/var/log/dnf.log**

main configuration file for dnf is /etc/dnf/dnf.conf preferred configuration location /etc/yum.repos.d . **dnf runs rpm in the background.**

RHEL 8 is shipped with two core repositories called BaseOS and Application Stream (AppStream).

BOOTLOADING:
The firmware phase, the bootloader phase, the kernel phase, the initialisation phase.

```
[BaseOS_RHEL_8.0]
name= RHEL 8.0 base operating system components
baseurl=file:///mnt/BaseOS
enabled=1
gpgcheck=0
```

EXAM TIP: Knowing how to configure a dnf/yum repository using a URL plays an important role in completing some of the RHCSA exam tasks successfully. Use two forward slash characters (//) with the baseurl directive for an FTP, HTTP, or HTTPS source.

dnf module list (node.js, mariadb etc.)
dnf group list (security, monitoring vs)
dnf group info “system tools” (show content of group)
dnf group install “system tools”
dnf list installed
dnf repoquery cifs-utils
dnf list installed | grep cifs-utils
dnf check-update
creating a directory inside `/dev` is not advisable, because `/dev` is dynamically managed by the system and device nodes are created automatically.
dnf repolist

Use the MD5 checksum for verifying its integrity and the *GNU Privacy Guard* (GnuPG or GPG) public key signature for

ensuring the credibility of its developer or publisher.

rpm -K /mnt/package/baseOS/zsh.3.3 —no-signature (checksum)

rpm -q gpg-pubkey (viewing keys)

rpm -qi <key> (view specific details)

rpm -Vf /etc/sysconfig (show package modification details)

chmod -v 644 /etc/sysconfig/atd (back to original state)

ls-lR
ls -lai | grep dir1
set -o noclobber
!! (repeat the last command)
!?grep? (repeat last command contains ls)
alias
ls -ld /etc/??? (prints all three letters folders)
ls /usr/bin/[g]* (all folders starting with g)
ls /usr/bin[a-c]* (folders between a and c]
ls /etc | less
grep operator /etc/passwd
grep -v nologin /etc/passwd (print lines that don't have nologin)
grep -n pattern /etc/passwd (print findings with number)
grep -w acce.. /etc/lvm/lvm.conf (prints lines including word acce..(i.e accept, access)

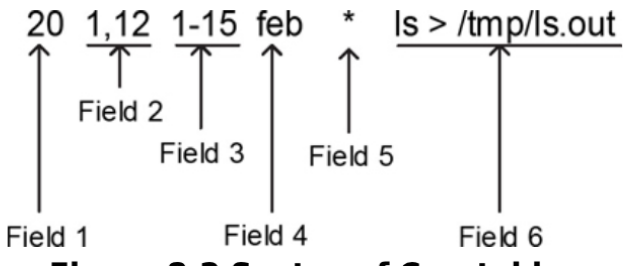
VIM <x> will delete current cursor, :<4,6d> ill delete lines
:%s/word/word1 -> replace word with word1
sed -i "s/old/new/g" replace old with new inplace
touch will change timestamp of the file
grep -i path ~/.bashrc (in-case-sensitive search)

file system check
df -h (see mounted disks)
(disk should be not mounted)
umount /dev/sdb
vi test & (run in the background)
jobs (check jobs)
fg %1 (bring job to fg)
kill <pid> (kill job)
ls /cdr /usr > output 2>&1
top then r (renice)

at 12:13pm 3/7/25
> date &> date.out
>ctrl+d

at -l (list jobs)

crontables:



The system runs the /etc/profile file first, followed by .bash_profile, .bashrc, and finally the /etc/bashrc file.

mandb
man -k xfs
In addition to the manual pages, apropos, whatis, info, and pinfo commands as well as documentation located in the /usr/share/doc directory are also available on the system.
COMMANDS:

mkdir -p A/B/C
touch (will change mod time)
stat (will print brith, mod, access time)
find / -name passwd
find / -maxdepth 3
find . -user root
ls -l | head -20 >> files
chmod ugo+rwx file / cmod ugo-rwx file
echo "umask 027" >> ~/.bashrc (newly created files: 640, directories: 750)
cat > b (write your input)
tar cvf b.tar b a (compress)
tar tf (view)
tar xf (extract)
tar cvfz /root/com.gz /etc
grep -nr "pass" /etc/passwd > /mnt/pass
ls /etc/yum.repos.d
vi /etc/yum.repos.d/local.repo

ROOT PASSWORD
1. "e" on boot screen

2. after UUID section, "rd.break" , before initrd. and type ctrl+x. on vm: rw init=/bin/bash
3. chroot /sysroot
4. pwd (should be /)
4. mount -o remount,rw /
5. passwd (will be given in exam)
6. touch /.autorelabel
7. exit
8. reboot

kernel and support files are stored at different locations in the directory hierarchy, f which three locations **/boot, /proc and /usr/lib/modules.** kernel files are **vmlinuz, initramfs, config and system.**map. kernel version appended to their names. **efi** and **grub2** subdirectories under /boot hold bootloader information. **grub.cfg** and **grubevn** contain critical data such as bootable kernel information and environment information that the kernel uses.

/**proc** is a virtual memory-based file system. about processor, memory, storage, file systems, swap, processes, network interfaces, connections, routing. /proc/**cpuinfo && /proc/meminfo.** data from /proc is referenced from many system utilities: top, ps, uname, free, uptime.

Updating kernel:
by default dnf command adds new kernel to the system.
rpm -qa | grep kernel
go to rhel official page, look for kernel, download, move files to /tmp/, dnf install /tmp/kernel*,
dnf list installed kernel*, reboot, choose the latest downloaded from grub menu

/proc/cmdline -> booting arguments.
location of efi: **/boot/efi/efi**

chroot: change root directory of a process. only process can access files

RAID(redundant array of independent disks)
stripe, mirror
/etc/default/grub
sudo grub2-mkconfig -o /boot/grub2/grub.cfg (after editing /etc/default/grub issue command and restart after)
/etc/grub/ (script location)
/boot/grub/grub.cfg && /boot/grub/grubenv

chroot creates isolated environment for testing. changes root directory of a process.

making new kernel default boot kernel: **grub2-mkconfig -o /boot/grub2/grub.cfg.**
system initialisation and service management scheme: **systemd**
grub.cfg file stores the location information of th partition. chroot command changes the specified directory path to /.
uname -r , rpm -q kernel

You need to know how to boot a RHEL 8 system into a specific target from the GRUB2 menu

to modify the fstab file or reset an unknown root user password.

rd.break, chroot /, mount -o remount, rw /, passwd, touch .autorelabel, exit
uname -snovmpr (m: architecture, r: kernel version)
IPTABLES
rule based firewall scans until it finds matches
allow block specific IP addresses or ports
powerful firewall built in linux
-
—----------

5. Initialisation, logging, system tuning
systemd remounts files once autofs finishes checks. d-bus is another communication method that allows multiple services running in parallel on a system to talk to one another.
units are systemd objects used for organising boot and maintenance tasks.
sshd.service, syslog.socket, umount.target, tmp.mount
units in /run/systemd/system are created at boot and destroyed when no longer needed.
initialisation scripts: /etc/rc.d/init.d
/usr/lib/systemd/system/sshd.service (see unit, service, install)

Unit Type	Description
Automount	Offers automount capabilities for on-demand mounting of file systems
Device	Exposes kernel devices in systemd and may be used to implement device-based activation
Mount	Controls when and how to mount or unmount file systems.
Path	Activates a service when monitored files or directories are accessed
Scope	Manages foreign processes instead of starting them
Service	Starts, stops, restarts, or reloads service daemons and the processes they are made up of
Slice	May be used to group units, which manage system processes in a tree-like structure for resource management
Socket	Encapsulates local inter-process communication or network sockets for use by matching service units
Swap	Encapsulates swap partitions
Target	Defines logical grouping of units
Timer	Useful for triggering activation of other units based on timers

Table 12-1 systemd Unit Types

Target	Description
halt	Shuts down and halts the system
poweroff	Shuts down and powers off the system
shutdown	Shuts down the system
rescue	Single-user target for running administrative and recovery functions. All local file systems are mounted. Some essential services are started, but networking remains disabled.
emergency	Runs an emergency shell. The root file system is mounted in read-only mode; other file systems are not mounted. Networking and other services remain disabled.
multi-user	Multi-user target with full network support, but without GUI
graphical	Multi-user target with full network support and GUI
reboot	Shuts down and reboots the system
default	A special soft link that points to the default system boot target (multi-user.target or graphical.target)
hibernate	Puts the system into hibernation by saving the running state of the system on the hard disk and powering it off. When powered up, the system restores from its saved state rather than booting up.

Table 12-2 systemd Targets

Targets

Targets are simply logical collections of units. They are a special systemd unit type with the .target file extension. They are also stored in the same directory locations as the other unit configuration files. Targets are used to execute a series of units. This is typically true for booting the system to a desired operational run level with all the required services up and running. Some targets inherit services from other targets and add their own to them. systemd includes several predefined targets that are described in Table 12-2.

VIRTUALISATION

```
yum install qemu-kvm libvirt virt-install virt-manager
```

REPOSITORY CONFIGURATION

0. dnf install <copy given epel release url>
1. dnf repolist
2.

CONTAINER

0. container file will be given
1. dnf install podman
2. podman build -t demo .
3. podman images
4. podman -d run -p 8080:80 <image>
5.

cal 2 2025

tar cvf , tar xvf
touch command can be used with -d and -t to add sepcific date and time, a and m will enable you to change access and modification time. touch -d 2020-09-20 sec.txt .
touch -m command will reverse it to original time.
soft link can link directories hard linnk can not link directories .

Boot: BIOS-> Master Boot Record(MBR)->partition Table(PT)->Boot loader(Grub)->Kernel-> Mounting / /usr->/etc/inittab(default run level)->/etc/fstab

run levels:
run level 0 power off
run level 1: singleuser mode text mode
run level 2 multiusr text mode except NFS, NIS
run level 3: support all services including NFS and NIS(network information service) (default)
run level 4: unused
run level 5: multiuser graphical mode

User account information for local users is stored in four files that are located in the /etc directory. These files—passwd, shadow, group, and gshadow.
UIDs between 1 and 200 are used by Red Hat to statically assign them to core service accounts. UIDs between 201 and 999 are reserved for non-core service accounts, and UIDs 1000 and beyond are employed for normal user accounts.
cat /etc/login.defs
head -3 /etc/passwd
tail -3 /etc/passwd

usermod -l user2new -u 2000 -d /home/user2new -m -s /sbin/nologin user2000
useradd user4 -s /sbin/nologin
echo redhat | passwd --stdin user4
vi /et

(nologin is when user does not need login access)
userdel user3new
grep user2new /etc/passwd

tuned-adm list
tuned-adm active (check which profile is active)
tuned-adm recommend
tuned-adm profile virtual-guest

docker run --cap-add=SYS_TIME -it bb4496e662fb /bin/bash

dnf install chrony (NTP)
dnf install -y procps

docker run --privileged -d --name systemd_container -v /sys/fs/cgroup:/sys/fs/cgroup:ro 5c79fba2bcae
What is the default number of days files in /tmp are kept before they are automatically deleted if not accessed or modified?
uname -v or cat /proc/version10 days (apropos -a list directory
apropos " list directory" / man -k ext4
The udev service (part of systemd-udevd) is responsible for creating device nodes dynamically at system startup.
Wayland has replaced the X Window System as the default display protocol in RHEL 8.
RHEL supports seven types of files: regular, directory, block special device, character special device, symbolic link, named pipe, and socket.

ln original.txt hardlink.txt # Create a hard link
ls -li original.txt hardlink.txt # Check inode numbers(same in hardlinnk)

local.repo
name=baseOS
baseurl=https://xyz.server.com/baseOS
gpgcheck=0
enabled=1

name=appStream
baseurl=https://xyz.server.com/appStream
gpgcheck=0
enabled=1

subscription-manager attach –auto
subscription-manager repos –list
subscription-manager repos –enable=rhel-9-for-x86_64-baseos-rpms
subscription-manager repos –enable=rhel-9-for-x86_64-appstream-rpms
dnf repolist
dnf clean all
dnf update
ping subscription.rhsm.redhat.com
sudo subscription-manager list –consumed
subscription-manager unregister
subscription-manager clean
subscription-manager register –username=<> –password=<>

These permission bits are set user identifier bit (commonly referred to as setuid or suid), set group identifier bit (a.k.a. setgid or sgid), and sticky bit.
The setuid and setgid bits may be defined on binary executable files to provide non-owners and non-group members the ability to run them with the privileges of the owner or the owning group, respectively. The setgid bit may also be set on shared directories for group collaboration. The sticky bit may be set on public directories for inhibiting file erasures by non-owners.

A common example is the su command that is owned by the root user. T

timedatectl set-ntp true

groupadd newgroup (should be listed in /etc/group)
useradd harsh -G newgroup
useradd nolog -s /sbin/nologin
passwd nolog (set password to redhat)
setfacl -m u:natasha:rw /var/fstab (m for modify)
getfacl /var/fstab
setfacl -m g:Mac:--- /var/fstab
4: read , 2: write 1 : execute
chmod u-x testfile -v (verbose)
chmod go+w testfile -v

chown :Mac /linux
groupadd blue
chgrp blue
chmod g+s . (now all the files will belong to group blue)
chmod +t (Sticky Bit: ensure only linux group can delete files)
chmod g-w, u+r testfile -v (from group remove writing, to user add read)
chmod -v u+s /usr/bin/su (adding user id to /usr/bin/su)

systemctl not default on container env. should enable cgroups while running:
(i. e. : docker run --privileged -d --name systemd_container -v /sys/fs/cgroup:/sys/fs/cgroup:ro centos:8 /usr/sbin/init)

DISK PARTITION:
lsblk (check devices)
fdisk /dev/sda2 (+1G for 1 GB partition, then n, p, w for write)
partprobe /dev/sda2
lsblk(check the newly created partition)
mkdir newdisk
mkfs.xfs /dev/sda2p1
mount /dev/sda2p1 /newdisk (not persistent yet)
vi /etc/fstab (write in the file: /dev/sdap1 /newdisk defaults 0 0)
mount -a

SWAP MANAGEMENT:
free -m
fdisk nvmep2 (from the input menu type:
0.(type letters, m for help)
1. n(new), p(primary), w(write)
2. partition number default
3. first sector: default
4. last sector: +750M

5. type: t
6. partition number: 3
7. L (get hex codes, 82 for linux swap) (change partition type to linux swap)
8. partprobe nvmep2 (will make changes permanent)
9. mkswap /dev/nvmep2
10. vi /etc/fstab (/dev/nvmep2 swap swap defaults 0 0)
11. swapon -a (if no error, check with free -m, see swap has has increased by 750Mb)

LVM: (create logical volume, give size, extend existing logical volumem)
1. Create Physical Volume (pv)
2. Create Volume Group (vg)
3. Create Logical Volume (lv)

pvcreate nvme0v3
pvcreate nvme0v4
pvcreate nvme0v5
pvs (show)
vgcreate vgtest /dev/nvme0v3 nvme0v4 nvme0v5 (create volume groups)
vgs (show)

(linear, striped, mirrored volumes)
lvcreate -L 8Gb -n lvtest vgtest
lvs
vi /etc/fstab (/dev/vgtest/lg1/ /lv xfs defaults 0 0)
mkdir /lv
mkfs.xfs /dev/vgtest/lvtest
mount -a

LVM Extension:
vgs
lvextend -r -L +2Gb /dev/vg1/lv1
vgextend vg1 /dev/nvme0v5
vgs
lvremove /dev/vg1/lv1 (you'll get warning filesystem is in use)
vim /etc/fstab (comment vg1/lv1)
unmount /lv
lvchange -an /dev/vg1/lv1
lvremove /dev/vg1/lv1
lvs
vgremove vg1
vgs
vgcreate -s 8M vg1 /dev/nvme0v3
lvcreate -l 10 -n lv2 /dev/vg1 (creating 80M logic volume 10 times)

STRATIS:
blockdev:minsize 1 gb
pool (combined block devices to create pool)
filesytem(no fixed size for filesystem, automatically grows)

dnf install stratisd stratis-cli
systemctl start stratisd
systemctl enable stratisd
stratis pool create pool1 /dev/nvme0n5 (create pool)
stratis pool list
stratis pool add-data pool1 /dev/nvme0n4 (extend pool)
stratis filesystem create pool1 fs1 (create filesystem)
stratis filesystem list
stratis filesystem create pool1 fs2
stratis filesystem list
mkdir /fs1
vi /etc/fstab (copy UUID from filesystem list output /fs1 xfs defaults,x-system.requires=stratisd.service 0 0)
mount -a

VIRTUAL DATA OPTIMISER (VDO)-deprecated.
New one: lvmvdo:
compression, thin provisioning, deduplication

dnf install lvm2 kmod-kvdo vdo

vdo create --name vdo1 --device /dev/nvme0n2 --vdoLogicalSize=50G
vdo list
mkfs.xfs /dev/mapper/vdo1
mkdir /vdo1
vi /etc/fstab (/dev/mapper/vdo1 /vdo1 auto defaults,x-systemd.requires=vdo.service 0 0)
mount -a
man vdo

dnf module info postresql:10
dnf module install -y postgresql:10

you can only have one module installed at a time.

CRON:
execute command /usr/local/bin/backup at 10:00 am on Feb 4th every year.
crontab -e (0 10 4 2 * /usr/local/bin/backup)

configure cron job for a user jiu at 12:08 every Thursday execute /bash/echo hello
crontab -u jiu -e (08 12 * * THU /bash/echo hello)

GREP:
grep -i "root" /etc/group
grep -i "sbin" /etc/passwd > /tmp/pass

CH ROOT PASSWD:
press "e" boot screen
put "rd.break" after word quiet --
mount -o remount,rw /sysroot --
chroot /sysroot --
passwd
touch /.autorelabel --
exit
reboot

NETWORKING/HOSTNAME

ip addr show ens160
nmcli con add con-name "Default" type ethernet ifname ens160
nmcli con show
nmcli con add con-name "Default1" type ethernet ifname ens160 ipv4 192.168.1.1/24 gw4 192.168.1.2
nmcli con up "Default1"
nmcli con show Default1
nmcli con mod "Default1" connection.autoconnect yes
nmcli con show Default1
nmcli con mod Default1 ipv4.addresses 192.168.2.2/16
nmcli con mod Default1 ipv4.dns 172.2.2.2
nmcli con mod Default1 ipv4.addresses 192.168.3.3/24 (multiple ip addresses can be added)
nmcli con add "Net" type ethernet ifname eth0 ipv4.addresses 200.0.0.12/16 gw4 20.0.0.1
nmcli con mod Net ipv4.dns 8.8.8.8
nmcli con show Net
nmcli con up Net

nmcli con add "net2" type ethernet ifname ens160 ipv4.addresses 172.24.5.10/24 gw4 172.24.5.48

nmtui (alternative to nmcli)
/etc/hostname
hostnamectl status
hostnamectl set-hostname server

SELINUX
touch /var/www/html
ls -ld /var
mkdir /new
touch /new/index.html
ls -ls /new/index
vi /etc/httpd/conf/httpd.conf (check DocumentRoot="/var/www/html"
vi /new/index "DocumentBoot "/var/www/html"
ls -lZ /var/www/html/index.html (httpd_sys_content_t) is the content
ls -lZ /new/index (default) is the content

<Directory "/new">
    AllowOverride All

```
    #Allow open access
    Require all granted
</Directory>
```

--> add above to /etc/httpd/conf/httpd.conf

SEMANAGE:
semanage fcontext -a -t httpd_sys_content_t "/new(/.*)?"
restorecon -Rv /new

SElinux modes: disabled permissive(0) enforced(1)
getenforce
setenforce 0 | 1
/etc/sysconfig/selinux (modify file to disable, reboot is required)
getsebool -a | grep httpd_enable (policy bool)
setsebool -P httpd_enable_homedirs on (-p flag permananent change)

selinux modes, booleans, context, port

httpd is able to access home dir: (getsebool | grep httpd_enable_homedirs)
system is not able to access httpd on port 82
semanage port -a -t http_port_t -p tcp 82 (systemctl restart httpd is required)

ensure httpd is able to access files at test directory.

PODMAN:
podman login registry.redhat.io (optional)
podman search httpd
podman pull docker.io/registry/httpd
podman rmi <image>
podman run -d --name httpd -p 8080:80 <imageID>
podman ps
curl localhost:8080 (outside the httpd container, see It works!)
podman stop <container>
podman rm <container>
podman run --d it <imageID> /bin/bash
podman exec -it <containerID> /bin/bash (inside the container check find . -name index.html see: /usr/local/apache2/htdocs)
mkdir /web && touch /web/mypage.html && vi mypage.html (add some context)
podman run -d --name web1 -p 8080:80 -v /web:/usr/local/apache2/htdocs/:Z <imageid>
after mapping
semanage fcontext -a -t httpd_sys_content "/web(/.*)?"
restore con -Rv /web
curl localhost:8080/mypage.html

QUADLET
podman info --debug | grep "rootless"
systemctl --user enable podman-auto-update-timer

systemctl --user daemon-reload

/etc/containers/systemd/sleep.container
[Unit]
Description=A minimal container

[Container]
Image=centos
Exec=sleep 60

[Service]
Restart=always

systemctl daemon-reload
systemctl start sleep.service
systemctl start sleep.service (auto restart once system rebooted)
systemctl status sleep.service

Quadlet files to be stored in either:
user: /usr/share/containers/systemd/
system wide: /etc/containers/systemd/

https://dokumen.pub/qdownload/rhcsa-red-hat-enterprise-linux-8-training-and-exam-preparation-guide-ex200.html

FILESYSTEM:
fsck.ext4 /dev/sda1
xfs-repair -L /dev/sda1

GROUP QUOTA:
mount -o remount, usrquota, grpquota /dev/sdb2/ /quota

/dev/sdb2 /quota ext4 ->/etc/fstab

SCHEDULING AND PROCESS ADMIN
at 10:03am today
> command
ctrl+D
atq

CPIO: occupies less space compared to tar.
cpio -icvf -I /root/backup

FTP.

AUTOFS:
SERVER
dnf install -y nfs* nfs-utils autofs
mkdir /share
touch /share/f1 /share/f2
chmod 777 /share
/share <clientIP>(ro,sync) >> /etc/exports
exportfs -avr
firewall-cmd --add-service={nfs,mountd,rpc-bind} --permanent
firewall-cmd --reload

CLIENT -in exam responsiblity only client part-
dnf install nfs-utils autofs
showmount -r <clientIP>
/auto_mount /etc/auto_misc --timeout=60 >> /etc/auto.master
access --rw,soft,intr <serverIP>:/share >> /etc/auto.misc
systemctl enable autofs --now (after this command /auto_mount or /afs directory should be created)
cd auto_mount && cd access && ls (should see files f1 and f2)

SHELL
if [$? -eq 0]; then
    echo "succesfull exit"

$0 -> script's name
$1 -> firstarg
$# -> number of args

SSH
systemctl status sshd
firewall-cmd --zone=public --permanent --add-service=ssh

A list of the users who have successfully signed on to the system with valid credentials can be printed using one of the two basic Linux tools: who and w.
last
last user1000
last root
lastb
lastlog
id
id user1000
groups user1000

Service accounts take care of their respective services, which include apache, ftp, mail, and chrony.

3 main: networking | storage | manage groups

find -mmin -300 -exec file {} \;

find / -type p -o -type s 2>/dev/null
find /usr -atime -100 -size -5M -user root
setfacl -x u:user2000 /tmp/testfile

in vi %s/tes/fes/
sed -i 's/globe/earth/g'

/etc/nologin.txt (custom no login test if -s /sbin/nologin is defined when creating useradd user5 -s /sbin/nologin)
useradd -D default login settings
useradd -D -s /bin/sh -b /custom/home: this would set the default shell to /bin/sh and home directory to /custom/home for all future users.
Name the four local user authentication files. /etc/passwd, /etc/group, /etc/shadow, /etc/gshadow
The who command in Linux consults the /var/run/utmp file to display information about currently logged-in users.
/var/log/wtmp: Keeps a history of all logins and logouts.
/var/log/btmp: Logs failed login attempts.
/etc/shadow- is the backup for /etc/shadow
/etc/nologin is a special file in Linux. When it exists, it prevents all non-root users from logging into the system.
who, w, id, groups

The lastlog command in Linux displays the most recent login information of all system users.

password aging is a secure mechanism to control user passwords in Linux
~

PACKAGES
rpm -q vsftpd
rpm -q createrepo
rpm -qf /bin/bash : Queries which installed RPM package provides the file `/bin/bash`.

NFS Server:
yum install nfs*
(remote server)
vi /etc/exports -> /remote 192.168.11.8(rw, sync)
exportfs
(on the client)
ifconfig -a (check IP can reach to remote server
show mount -2 192.168.11.7
mount -t nfs 192.168.11.7:/remote /nfs (change IP of client to 11.7 if it does not mount)
service network restart
df -h (check mount)
vi /etc/fstab -> 192.168.11.7:/remote /nfs defaults 0 0
yum list autofs
vi /etc/auto.master

vi /etc/auto.misc -> ram -fstype:nfs 192.168.11.7 (config for mount point)

SAMBA Server(device and file share across heterogenous OSes.
Ports: 137(name), 138(datagram), 139(session)
vi /etc/samba/smb.conf
cd /etc/samba && grep "log" *
service smb restart
(on the client)
smbclient -L //182.168.11.7/ -N
(exercise: secure shared shares in samba server)(disable printer sharing)

DHCP server:
PORTS: 67-bootp, 68-dhcp
yum install dhcp -y
cd /usr/share/doc/dhcp_server
cp dhcpd.conf.example /etc/dhcpd/dhcpd.conf
vi /etc/dhcpd.conf (edit subnet and range according to IP)
service dhcpd start

NETWORK INFORMATION SERVICE a.k.a YellowPages (NIS)

DNS RECORDS
CNAME. files.example.org alias hostname
A RECORD IP address of the domain. maps hostname to IPv4 address to be saved in icann.net
MX RECORD maps domain name to mailexchange server. host can have multiple MX
PTR RECORD maps ipv4 to the canonical name for the host. adds 192.168.1.10.in-addr.arpa (reverse address)
NS record maps domain name to list of DNS servers authoritative for that domain.
named-checkconf /etc/named.conf
named-checkconf /etc/rfc1912.zones
named-checkzone example.com example.for
named-checkzone 192.168.1.11.in-addr.arpa

APACHE:
port: 80
yum install http*

Additionally, certain configuration options have been deprecated or removed in recent BIND 9 releases. For instance, the `auto-dnssec` configuration statement was removed, and users are advised to use `dnssec-policy` or manual signing instead.
BIND 9 Documentation
It's advisable to consult the release notes of the specific BIND version you're using to stay informed about any changes to configuration options.

yum install system-config-kickstarter
system-config-kickstart(will bring gui)
add
root password

installation method: ftp: server 192.168.10.7, ftp: directory: pub
partition info: add 10GB mount boot filetype ext4,add swap 2048
network config: eth0: dhcp.
authentication: keep default settings
firewall: SElinux disabled
installation:
vmlinuz initrd.initrd.img repo=ftp://192.168.10.7/

Basic sendmail (deprecated)
yum install sendmail*
vi /etc/mail/sendmail.mc (nothing to be added)
make -C /etc/mail
vi /etc/mail/sendmail.mc
service sendmail start
service sendmail status
sendmail -v -s “Test Email” user@server.example.org

POSTFIX:
dnf remove sendmail
dnf install postfix
dnf install chkconfig
chkconfig postfix off
systemctl status postfix
vi postfix.sh …
vi /etc/rc.local -> /root/postfix.sh
LVM

redirection
umask 022
chmod ugo
tar xvf gz
chmod g-s /usr/bin/write -v (gid)

TARGETS:

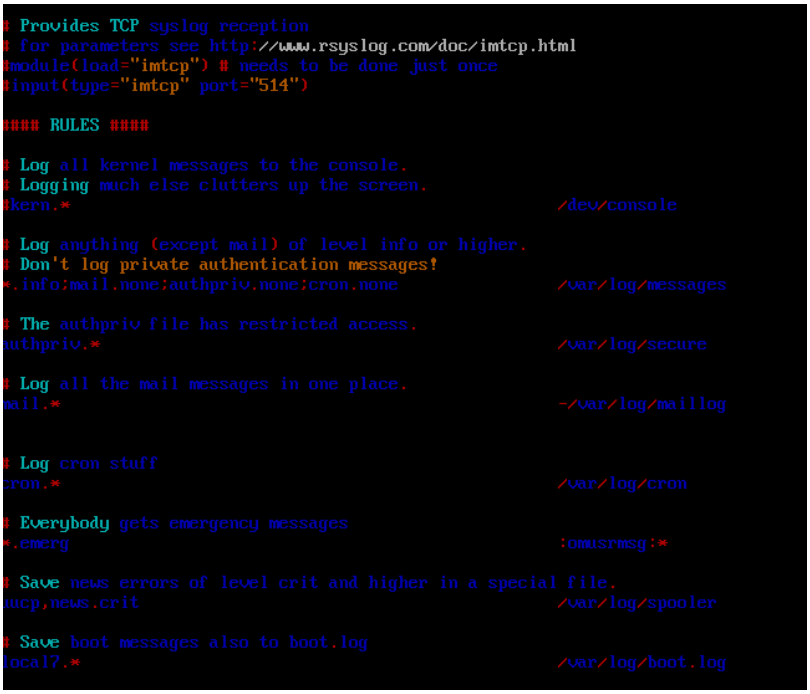
Subcommand	Description
daemon-reload	Re-reads and reloads all unit configuration files and recreates the entire user dependency tree.
enable (disable)	Activates (deactivates) a unit for autostart at system boot
get-default (set-default)	Shows (sets) the default boot target
get-property (set-property)	Returns (sets) the value of a property
is-active	Checks whether a unit is running
is-enabled	Displays whether a unit is set to autostart at system boot
is-failed	Checks whether a unit is in the failed state
isolate	Changes the running state of a system
kill	Terminates all processes for a unit
list-dependencies	Lists dependency tree for a unit
list-sockets	Lists units of type socket
list-unit-files	Lists installed unit files
list-units	Lists known units. This is the default behavior when systemctl is executed without any arguments.
mask (unmask)	Prohibits (permits) auto and manual activation of a unit to avoid potential conflict
reload	Forces a running unit to re-read its configuration file. This action does not change the PID of the running unit.
restart	Stops a running unit and restarts it
show	Shows unit properties
start (stop)	Starts (stops) a unit
status	Presents the unit status information

Table 12-3 systemctl Subcommands

systemctl list-unit-files
systemctl –failed
systemctl status atd
systemctl -t target (list all targets)
systemctl list-dependencies multi-user.target (list dependencies for multi-user target)
systemctl isolate multi-user (`systemctl isolate` command is used to change the current system's state by **isolating a specific target)**

SYSLOG: *rsyslogd* ***modern systems:*** *journalctl*

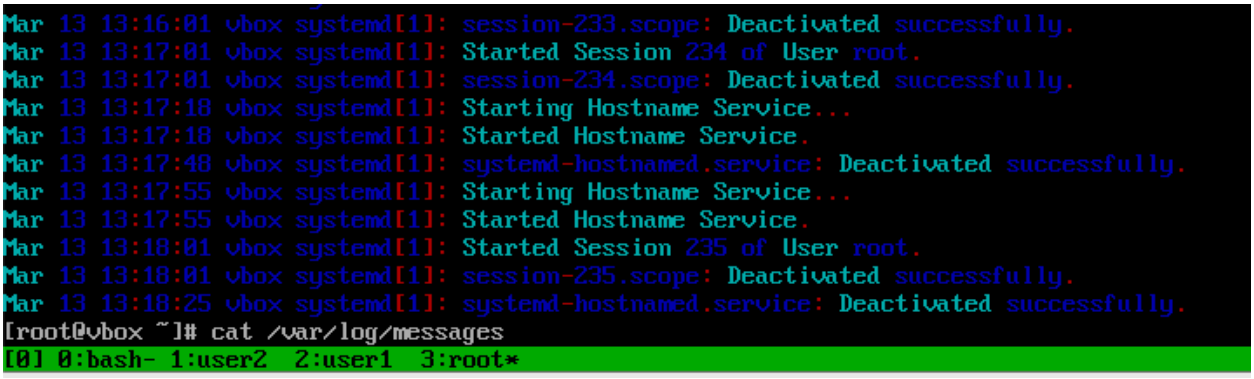
/var/log are the default location where the log files are stored. rsyslog service is modular, allowing modules listed in its configuration file to be dynamically loaded in the kernel as and when needed. each module brings new functionality to the system upon loading. ***systemctl start/stop/status rsyslog***
/etc/rsyslog.conf *:* configuration file. Rules section has **selectors**(left) **and action**(right), **facility**(left) and **priority**(right)**.**
rsyslogd logs messages based on priorities: emerg, alert, crit, error, warning, notice, info, debug. *it will keep for target level and higher levels.*



/var/log/messages | /var/log/boot | ls -l /var/log

files under /var/log can be filled very quickly. To prevent this a script called logrotate under /etc/cron.daily invokes the *logrotate* command. /etc/logrotate.conf

– each time a log file is rotated, an empty replacement file is created with the date as a suffix to its name, and logging restarted. services have each different logrotate configuration. /*etc/logrotate.d/****script has option for postrotate (such as gzip) the log files. latest system messages: **/var/log/messages (tail -f).** It is helpful to tail the messages file when starting or restarting the service.



journalctl:

addition to rsyslog, systemd implements logging service for the collection of logs. this is implemented via **systemd-journald** daemon. *journalctl* command will print messages. **journalctl -o verbose**, **journalctl -b** (from last systemboot). -0: since the last system boot, -1: since previous system boot, -2: since two previous system boot.

journalctl -kb0 (only kernel generated alerts since last reboot)

journalctl -n5 (list only last 5 lines)

journalctl /usr/sbin/crond (see logs generated by crond)

journalctl _*SYSTEMD*_UNIT=sshd.service

journalctl _PID=$(pgrep chronyd) / journalctl _PID=$(pgrep sshd)

journalctl –since 2019-10-10 –until 2019-10-16 -p err
journalctl –since today -p warning -r
journalctl -f *(real time viewing, same as tail -f, logger to send messages to journal, “write root/user”to print msg on other logged in users)*

logs are stored in /run/log/journal and its transient. options: volatile(stores in memory only), persistent(permanent under **/var/log/journal)**, **auto**(similar to persistent but does not create /var/log/journal), none(disables both volatile and persistent storage, not recommended).
file is rotated once a month. settings at **/etc/systemd/journald.conf.**

/etc/machine-id : where the system's machine ID is kept.

SYSTEM TUNING:
tuned: monitor storage, networking, audio, video and a variety of other connected devices. adjust parameters for better performance. there are several predefined tuning profiles, that may be activated statically or dynamically. for example during large file transfer network connection use increases. 9 profiles default, can create custom and save it under **/etc/tuned.** predefined profiles at **/usr/lib/tuned.**
profile management: tuned-adm (**tuned-adm list**)
systemctl –now enable tuned (will stick tuned service to auto-start)
tuned-adm profile virtual-guest | tuned-adm active | tuned-adm recommend | tuned-adm off

Profile	Description
Profiles Optimized for Better Performance	
Desktop	Based on the balanced profile for desktop systems. Offers improved throughput for interactive applications.
Latency-performance	For low-latency requirements
Network-latency	Based on the latency-performance for faster network throughput
Network-throughput	Based on the throughput-performance profile for maximum network throughput
Virtual-guest	Optimized for virtual machines
Virtual-host	Optimized for virtualized hosts
Profiles Optimized for Power Saving	
Powersave	Saves maximum power at the cost of performance
Balanced/Max Profiles	
Balanced	Preferred choice for systems that require a balance between performance and power saving
Throughput-performance	Provides maximum performance and consumes maximum power

Table 12-5 Tuning Profiles

systemctl set-default graphical-target
systemctl get-default

Master Boot Record vs. GUID Partition | VDO | MBR Disk | GPT Disk
data is stored on disks that are logically divided into partitions.

VDO: newer storage management solutions capitalize on thin provisioning, deduplication and compression to conserve storage space and improve data throughput.
a disk is stored on the disk in a small region, which is read by the operating system at boot time. region is referred to as MBR on the BIOS, and GUID Partition Table(GPT) on UEFI. store disk partition information and the boot code.

- **MSDOS (MBR)** and **GPT** are the two most common partition schemes in use today.
- **Apple Partition Map (APM)** was used in older Macs.
- **BSD Disklabel** is used primarily by BSD-based operating systems.
- **LVM** provides flexible disk management and works on top of MBR or GPT.
- **Solaris Partition Table** was used by older versions of Solaris before GPT.
- **RAID** is a technology for combining multiple disks but isn't a partitioning scheme in itself.
- **BIOS Boot Partition** is part of GPT and needed for BIOS-based booting.
- **Logical Partitions** are a feature within MBR, allowing more than four partitions on a disk.

storage management tools: **parted, gdisk, vdo, lvm, stratis.**

parted is a simple tool that understands both MBR and GPT formats. **gdisk** is designed to support the GPT format only, it may be used as replacement of parted. **VDO** is a disk optimizer software that takes advantage of certain technologies to minimize the overall data footprint on storage devices. **LVM** is a feature rich logical volume management solution that gives flexibility in storage management. **Stratis** capitalizes on thin provisioning to create volumes much larger in size than the underlying storage options.

thin provisioning(lvm): economical allocation and utilisation of storage space by moving arbitrary data blocks to contiguous locations. i.e LVM, VDO, Stratis can create *thin pool* space and assign volumes much larger than the physical capacity of the pool. When a preset custom threshold(i.e 80%) on the actual consumption of the storage is reached, expand the pool dynamically by adding more storage. after creating the partition, **/proc/partitions** is also updated. repair: journalctl -xb, xfs_repair, smartctl -a /dev/sda

add 100mb.vdi: /dev/sdb

Subcommand	Description
print	Displays the partition table that includes disk geometry and partition number, start and end, size, type, file system type, and relevant flags.
mklabel	Applies a label to the disk. Common labels are gpt and msdos.
mkpart	Makes a new partition
name	Assigns a name to a partition
rm	Removes the specified partition

Table 13-1 Common parted Subcommands

parted provides commands for viewing, labeling, adding, naming and deleting partitions.

*sudo **parted** /dev/sdb **print** (see unrecognised label, disk must be labelled before usage)*
*sudo **parted** /dev/sdb **mklabel** msdos*
sudo parted /dev/sdc mklabel gpt
sudo parted /dev/sdb print (see label)
*sudo parted /dev/sdb **mkpart** primary 1 50m*
sudo parted /dev/sdc mkpart primary 1 50m
sudo parted /dev/sdd mkpart pri 1 200m
*sudo parted /dev/sdd **rm** 1 (will delete partition)*

gdisk utility partitions disks using the gpt format. gdisk can create up to 128 partitions on a single disk on systems with UEFI firmware.

*sudo **gdisk** /dev/sdd*
o: delete all partitions
p: print partitions(1,
n: new partition(+20M)
w: write changes to disk
d1: delete partition

write root (will start message sharing)

VDO
1. making use of the thin provisioning technology to identify and eliminate empty(zero-byte) data blocks, referred as *zero-block elimination.* removing randomization of data blocks by moving in-use data blocks to contiguous locations on the storage device -
2. keeping an eye on the data being written on the disk. If data is redundant vdo skips writing.(kernel module: universal deduplication service).
3. *kvdo* compresses the residual data blocks and distributes them on a lower number of blocks.

vdo runs in the background and processes inbound data through the three stages on vdo-enabled volumes. vdo is not cpu/memory intensive process, consuming a low amount of resources.

vdo can be initialised just like disk partitions or they can be used as lvm physical modules.

Subcommand	Description
create	Adds a new VDO volume on the specified block device
status	Returns the status and attributes of VDO volumes
list	Lists the names of all started VDO volumes
start	Starts a VDO volume
stop	Stops a VDO volume

Table 13-2 vdo Subcommands

dnf install vdo kmod-kvdo -y
systemctl –now enable vdo

systemctl status vdo
systemctl list-units –type=service | grep vdo
modprobe vdo (if vdo is not loaded on kernel)

vdi is not treated as raw storage. vdi adds an extra layer. but, vdo should contact directly with the storage. Linux-based VMs, you can create a ***logical volume*** *(LV) with LVM and then use VDO on top of it. This allows VDO to manage storage optimization directly on the block device. raw block devices(VirtIO, SCSI)*

vdo create –name vdo-vol1 –device /dev/sdc –vdoLogicalSize16G –vdoSlabSize 120
vdo list | vdostats –hu | vdostats –verbose | vdo status –name vdo-vol1
vdo remove –name vdo-vol1

exam vm will have no outside connection. refer to the manual pages and the documentation under /usr/share/doc directory.

hostnamectl set-hostname rhcsa2.example
nmcli con mod enp0s3 *ipv4.addresses* 192.168.0.242/24
nmcli con mod enp0s3 *ipv4.gateway* 192.168.0.1
nmcli con mod enp0s3 *ipv4.dns* 192.168.0.1
nmcli con *down* enp0s3 && nmcli con *up* enp0s3 && nmcli con *show* enp0s3

useradd -u 9000 -c “this is user90” -e $(date -d “+4 days” + %Y-%m-%d) user90
useradd -m -s /sbin/nologin user50 (create user with no login)
passwd -l user90 (lock user90 fro logging)

setfacl -m u:user10:rw /tmp/file
setfacl -d -m u:user10 /tmp

mkdir -p /dev/dvdrom
mount /dev/cdrom /dev/dvdrom
vi /etc/fstab (/dev/cdrom /dev/dvdrom iso9660 defaults 0 0)

vi /**etc/yum.repos.d/rhel9-local.repo**

[rhel9-local]
name=RHEL 9 Local Repo
baseurl=file:///mnt/dvdrom/AppStream/
enabled=1
gpgcheck=1
gpgkey=file:///mnt/dvdrom/RPM-GPG-KEY-redhat-release

PHYSICAL VOLUMES, VOLUME GROUPS AND LOGICAL VOLUMES
Volume Groups (VGs) are part of the Logical Volume Manager (LVM), which provides a flexible and dynamic

way to manage disk storage. Here are the primary use cases for Volume Groups:

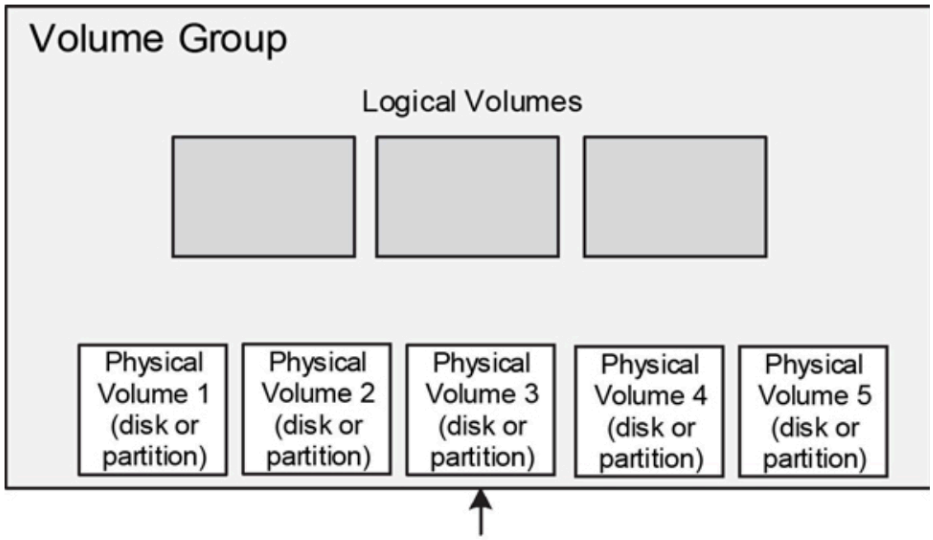


Figure 14-1 LVM Structure

*sudo **pvs***
*sudo **vgs***
*sudo **vgdisplay** (see PE: physical extent, LE: logical extent)*

logical extent can be smaller or larger than the physical extent.
*sudo **lvdisplay***
***sudo gdisk /dev/sdc** (n, +10m, w)*
sudo parted /dev/sdb set 1 lvm on
sudo parted /dev/sdd set 1 lvm on
sudo parted /dev/sdd set 2 lvm on
sudo parted /dev/sdc set 2 lvm on
sudo vgcreate -vs 1 vgnew /dev/sdd2
***sudo vgcreate -vs 1 vgnew2 /dev/sdc2** (add sdc2 to vgnew2)*

sudo pvcreate /dev/sdc1
sudo vgcreate vg100 –physicalextentsize 16M /dev/sdc1 (by default PE is 4MB)
sudo lvcreate -n lv100 -L 10M vg100
sudo lvremove /dev/vg100/lv100
sudo vgremove /dev/vg100

sudo pvcreate /dev/sdd1 (initialize pv)
sudo pvcreate /dev/sdb1
sudo pvs -v
sudo pvdisplay /dev/sdc1

sudo vgextend vgnew /dev/sdb1 *(will add sdb1 to vg(rhel))*
sudo vgextend vgnew2 /dev/sdc1
sudo vgs (show actual virtual group)
sudo vgdisplay -v vgnew
sudo lvcreate -l 1 -n lvnew vgnew
sudo lvcreate -l 1 -n lvnew2 vgnew2
sudo lvs
sudo lvdisplay /dev/vgnew/lvnew
sudo lvextend -L +10m /dev/vgnew/lvnew
sudo lvrename vgnew lvnew lvneww
sudo lvreduce -L 5m /dev/vgnew/lvneww
sudo lvresize -L 2m /dev/vgnew/lvneww
sudo pvmove /dev/sdc2 /dev/sdc3 (to another allocation in same vg)
sudo vgreduce vgnew2 /dev/sdc2 (before removal /dev/sdc2 data should be moved)
sudo pvremove /dev/sdc2

Command	Description
Create and Remove Operations	
pvcreate/pvremove	Initializes/uninitializes a disk or partition for LVM use
vgcreate/vgremove	Creates/removes a volume group
lvcreate/lvremove	Creates/removes a logical volume
Extend and Reduce Operations	
vgextend/vgreduce	Adds/removes a physical volume to/from a volume group
lvextend/lvreduce	Extends/reduces the size of a logical volume
lvresize	Resizes a logical volume. With the -r option, this command calls the resize2fs command and resizes the underlying file system as well. Applies to Ext2/Ext3/Ext4 file system types only.
Rename Operations	
vgrename	Renames a volume group
lvrename	Renames a logical volume
List and Display Operations	
pvs/pvdisplay	Lists/displays physical volume information
vgs/vgdisplay	Lists/displays volume group information
lvs/lvdisplay	Lists/displays logical volume information

Table 14-1 Common LVM Operations and Commands

sudo parted /dev/sdc set 1 lvm on (should update the flags part, see with print)
sudo pvcreate /dev/sdd1 /dev/sda -v (initialisation)
sudo vgcreate -vs -16 vgbook /dev/sdd1 /dev/sda
sudo vgdisplay -v vgbook
sudo pvdisplay /dev/sdd1

lv(pvcreate , vgcreate, lvcreate) and then apply vdo.

STRATIS(storage pool):
stratis capitalizes three matured storage comp: ***device mapper(dm),*** *kernel driver, the lvm solution and the xfs file system.*

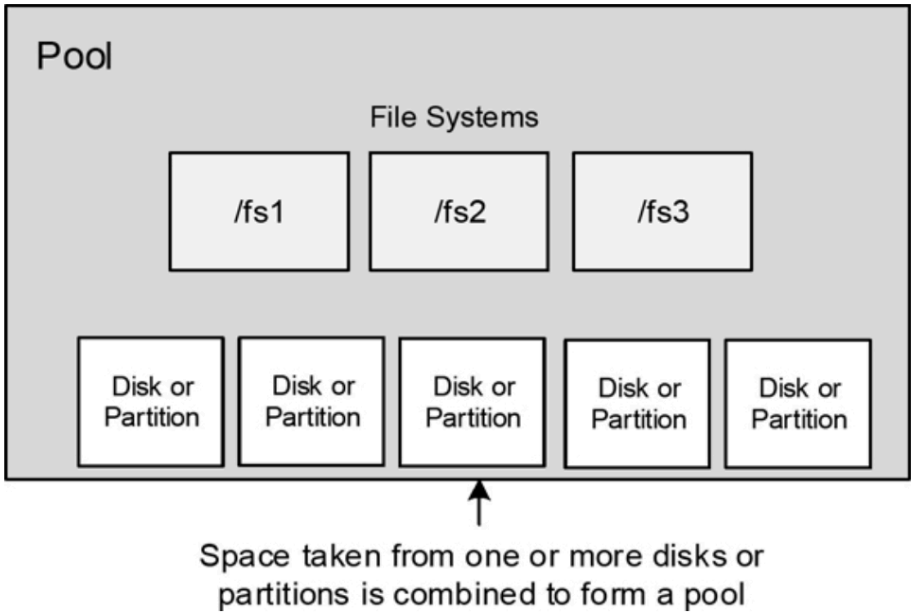


Figure 14-2 Stratis Structure

actual size grows with data stored. startis handles everything automatically so nothing should be manually configured.

Command	Description
pool	Administers storage pools. Subcommands are available to list, create, rename, expand, and destroy a pool.
blockdev	Lists block devices
filesystem	Administers file systems within storage pools. Subcommands are available to list, create, rename, and destroy a file system.

For each pool created, stratis creates a subdirectory under the /stratis directory. then creates symbolik link for each file system under the directory and keep actual files in /dev directory. **min disk size: 2gb for Stratis2.0**

stratis pool create poolnew /dev/sdb
stratis pool list
stratis blockdev list poolnew
stratis pool add-data poolnew /dev/sde
stratis filesystem create poolnew fsnew
stratis filesystem list
stratis filesystem destroy poolnew fsnew (remove fs from pool)
stratis pool rename poolnew poolneww
stratis pool destroy poolneww (remove pool)
stratis filesystem list (confirm removal)

mkdir /newfs
mount /stratis/newpool/newfs /newfs
umount /newfs
lsblk

LVM and Stratis can exist on the same system as long as they manage different partitions.
Stratis typically manages **pools** of storage, while **LVM** works with **volume groups** and logical volumes.

FILESYSTEMS
logical container that stored files and directories. Each file system is created with a discrete partition, vdo volume, logical volume or stratis pool. production RHEL has */ and /boot* during installation, additionally other file systems created during boot are: */home, /opt, /tmp, /usr, /var.*

RHEL supports three basic groups: *disk-based, network-based, memory-based.*
disk-based: SATA, USB, Fibre and other, network: shared, memory: virtual.
EXT is divided into two: first is the metadata(inode, permission, ownership, access/creation times), second is the actual data.
XFS high performing 64bit extent based journaling file system type. it does not run file system checks at system boot, relies on the user to use *xfs_repair.* Only caveat to use is that XFS's inability to shrink. XFS uses *sophisticated techniques in its architecture for speedy input/output performance.* ***It can be snapshot in a mounted active state, and then be used for backup or other(migration) purposes.***
VFAT: FAT16, introduced early versions of ms-dos.
ISO9660: optical disc media(cd/dvd), originally high-sierra file system(hsfs)

Command	Description
Extended File System	
e2label	Modifies the label of a file system
mke2fs	Creates a file system. Can also be invoked as mkfs.ext3, mkfs.ext4, mkfs -t ext3, and mkfs -t ext4.
resize2fs	Resizes a file system. This command is automatically invoked when the lvresize command is run with the -r switch.
tune2fs	Tunes or displays file system attributes
XFS	
mkfs.xfs	Creates a file system. Can also be invoked as mkfs -t xfs.
xfs_admin	Tunes file system attributes
xfs_growfs	Extends the size of a file system
xfs_info	Exhibits information about a file system
VFAT	
mkfs.vfat	Creates a file system. It is equivalent to using mkfs -t vfat.
General File System Commands	
blkid	Displays block device attributes including their UUIDs and labels
df	Reports file system utilization
du	Calculates disk usage of directories and file systems
lsblk	Lists block devices and file systems and their attributes including their UUIDs and labels
mount	Mounts a file system for user access. Displays currently mounted file systems.
umount	Unmounts a file system

df -h
du -sh

Table 15-2 File System Management Commands

mount -t xfs *(getting information on the xfs file system)*

mount command requires sudo and absolute path. mount point should be empty. after mount, the kernel places an entry on the **/proc/self/mounts**. (file structure is similar to /etc/fstab).
remount is used to enable/disable an option.-

Option	Description
acl (noacl)	Enables (disables) the support for ACLs
auto (noauto)	Mounts (does not mount) the file system when the -a option is specified
defaults	Mounts a file system with all the default values (async, auto, rw, etc.)
_netdev	Used for a file system that requires network connectivity in place before it can be mounted. VDO and NFS are examples.
remount	Remounts an already mounted file system to enable or disable an option
ro (rw)	Mounts a file system read-only (read/write)

UUID: **xfs-admin -u /dev/sda1 /**
tune2fs (ext4)
blkid (all)
- UUID which is assigned to the file system remains persistent across system reboots.
- RHEL attempts to mount all file systems that are listed in /etc/fstab over its UUID or label.
- with the unmount kernel removes the corresponding file system entry from **/proc/self/mounts**.
monitoring disk: **df -h** (usage information) **du -sh** (amount of space a file or directory occupies)
-T to add the file system type to the output (example: **df -hT**)
-x to exclude the specified file system type from the output (example: **df -hx tmpfs**)
-t to limit the output to a specific file system type (example: **df -t xfs**)
-i to show inode information (example: **df -hi**)

Labelling:
xfs-admin -l /dev/sda1 , *lsblk -f /dev/sda1 (to view label)*
to create a label target must be unmounted.
umount /boot
xfs_admin -L bootfs /dev/sda1
mount /boot
e2label for ext filesystem, label is not recommended for VDO and LVM volumes. It is self-managed.

/etc/fstab file needs only one of four: block device, uuid, label, mount point. any missing or invalid entry may make the system unbootable, and should be booted in emergency mode to fix the file. find the line start with linux and add: **systemd.unit=emergency.target**

1: **path or UUID** 2: mount-point(**/, /boot, none, swap**) 3: filesystem: **Ext4, XFS, VFAT, ISO9660**, **swap** 4: comma-seperated options(**defaults**) 5: dump-check(**0**: disables) 6: sequence number runs e2fsck(repair utility) **0** for memory-based, remote and removable filesystems, **1** for /, and **2** for /boot and other physical systems. only valid for Ext.

UUID /boot xfs defaults 0 0

EX15.1
parted /dev/sdb mklabel msdos
parted /dev/sdb mkpart primary 1 20m
parted /dev/sdb mkpart primary 21 40m
mkfs -t ext4 /dev/sdb1

mkfs -t vfat /dev/sdb2
parted /dev/sdb print
mkfs.xfs /dev/sdd -f (parititon > 300mb, -f fore removal of old partition)
lsblk -f /dev/sdb /dev/sdb >> /etc/fstab (determine UUID) organise fstab after
mkdir /ext4fs1 /vfatfs1 /xfsfs1
mount -a
systemctl daemon-reload
df -hT

15.2 EXT with VDO

15.3 XFS with LVM

parted /dev/sde mklabel gpt
parted /dev/sde mkpart primary 1 160m
pvcreate /dev/sde1
vgcreate -s 16 vgfs /dev/sde1
lvcreate -n ext4vol -L 20 vgfs
lvcreate -n xfsvol -L 20 vgfs
mkfs.ext4 /dev/vgfs/ext4vol
mkfs.xfs /dev/vgfs/xfsvol -f
lsblk -f /dev/sde >> /etc/fstab
mkdir /ext4fs2 /xfsfs2
mount -a
df -hT | grep fs2

15.4 Resize Ext4 and XFS

pvcreate /dev/sde2
vgextend vgfs /dev/sde2
vgdisplay vgfs (check pv count:2)
lvextend -L +10m /dev/vgfs/ext4vol (adds 10m to ext4vol)
lvdisplay /dev/vgfs/ext4vol
lvextend -L +10m /dev/vgfs/xfsvol (adds 10m/one extent(16m) to xfsvol)
lvdisplay /dev/vgfs/xfsvol
lvresize -r -L +10m /dev/vgfs/xfsvol
lvresize -r -L +10m /dev/vgfs/ext4vol

15.5 Create, Mount, Expand with Stratis

stratis pool create /dev/sdf
stratis pool list
stratis blockdev list strpool
stratis filesystem create strpool strfs2
stratis filesystem list (UUID can be found)
lsblk /dev/stratis/strpool/strfs2 -o UUID (or such)
>/etc/fstab
“UUID” /strfs2 xfs **x-systemd.requires=stratisd.service** 0 0
mkdir /strfs2
mount -a
stratis pool add-data strpool /dev/sdg (add new pv to stratis pool)

SWAP

an independent region on the physical disk used for holding idle data until it is needed. Physical memory is split into small chunks: **pages** and maps their physical location to virtual locations on swap. Mapping

information is stored in the *page table*, maintained by the kernel. Paging data out and in is known as demand paging. Excessive amount of paging causes system degradation called *thrashing.* When trashing begins, the system deactivates new processes to launch until the system reaches the threshold value back.
Swap may be twice or larger than the physical memory or smaller. *free -ht*

mkswap: create swapon: activate swapoff: deactivate swapon -s: list

15.6 Create and Activate Swap

parted /dev/sdb mkpart primary 1 80
parted /dev/sdb mkpart primary 80 160
parted /dev/sdb mkpart primary 160 240
vgcreate vgfs /dev/sdb1
lvcreate -n swapvol -L 80
mkswap /dev/vgfs/swapvol
mkswap /dev/sdb2
swapon -a
swapon
free -ht

```
[tm@vbox ~]$ free -ht
               total        used        free      shared  buff/cache   available
Mem:           1.7Gi       1.2Gi        76Mi        20Mi       637Mi       543Mi
Swap:          2.1Gi          0B       2.1Gi
Total:         3.8Gi       1.2Gi       2.2Gi
```

lsblk -f >> /etc/fstab
UUID="<>" swap swap pri=1 0 0
/dev/vgfs/swapvol swap swap pri=2 0 0
cat /proc/swaps

```
[tm@vbox ~]$ cat /proc/swaps
Filename                                Type            Size            Used            Priority
/dev/dm-1                               partition       2097148         0               -2
/dev/sdb3                               partition       59388           0               1
/dev/dm-2                               partition       40956           0               2
```

cat /proc/mounts

rhel99_1.vdi	399,66 MB	2,00 MB
rhel99_2.vhd	399,66 MB	3,00 KB
rhel99_3.vmdk	399,66 MB	64,00 KB
rhel99_4.vdi	399,66 MB	2,00 MB

VMDK (VMware Virtual Machine Disk):

- VMware uses a specific format for its virtual disks, and one key component of a VMDK file is the way it stores metadata for the virtual disk. The 64KB file size you're referring to is often related to the block size or the minimum allocation for a virtual disk in VMware. VMware may reserve small blocks of space in the virtual disk for things like metadata, snapshots, and disk management. Thus, a VMDK file might show up as 64KB, even if the actual disk content is smaller.

VHD (Virtual Hard Disk - used by Hyper-V):

- VHD files are also formatted in a way that has space reserved for disk management structures and metadata. The 3KB "real" size you mentioned may be the actual size of the metadata block or the overhead associated with tracking the virtual disk's state.

VDI (VirtualBox Disk Image):

- Similarly, VirtualBox's VDI format has a slightly larger size (2MB real size) because of the way it organizes blocks and metadata. VirtualBox manages virtual disks in a way that can result in some overhead storage for file system structures, although it's typically more efficient in terms of block size compared to VMware or Hyper-V.

15.1 Create VFAT, Ext4, XFS File Systems in partitions and mount persistently

```
NAME                FSTYPE      FSVER    LABEL UUID                                   FSAVAIL FSUSE% MOUNTPOINTS
sda
├─sda1              xfs                       0cdd878c-ed88-4631-9ed7-1df8a52c91c6    610.4M    36% /boot
└─sda2              LVM2_member LVM2 001      3KsWDq-77cJ-CIpv-SsST-FcJ6-x1If-h1ubyJ
  ├─rhel_vbox-root  xfs                       e1fd8d5c-58a0-49ce-bad7-2fee1450483a     12.2G    28% /
  └─rhel_vbox-swap  swap        1             a22b1e7b-52d6-42fb-bc94-c65c20f00f83
sdb
├─sdb1              vfat        FAT16         782A-1688                                74.8M     0% /vfatfs5
├─sdb2              ext4        1.0           db7ee7fa-d24e-4d6d-8622-59c0322a0d1a     60.4M     0% /ext4fs5
└─sdb3              xfs                       c090dd64-208d-49af-b62d-60adae8fef51     65.2M     6% /xfsfs5
```

```
[root@vbox tm]# df -h
Filesystem                  Size  Used Avail Use% Mounted on
devtmpfs                    4.0M     0  4.0M   0% /dev
tmpfs                       888M     0  888M   0% /dev/shm
tmpfs                       355M  5.7M  350M   2% /run
/dev/mapper/rhel_vbox-root   17G  4.8G   13G  28% /
/dev/sda1                   960M  350M  611M  37% /boot
tmpfs                       178M   96K  178M   1% /run/user/1000
/dev/sdb1                    75M     0   75M   0% /vfatfs5
/dev/sdb2                    66M   14K   61M   1% /ext4fs5
/dev/sdb3                    70M  4.5M   66M   7% /xfsfs5
```

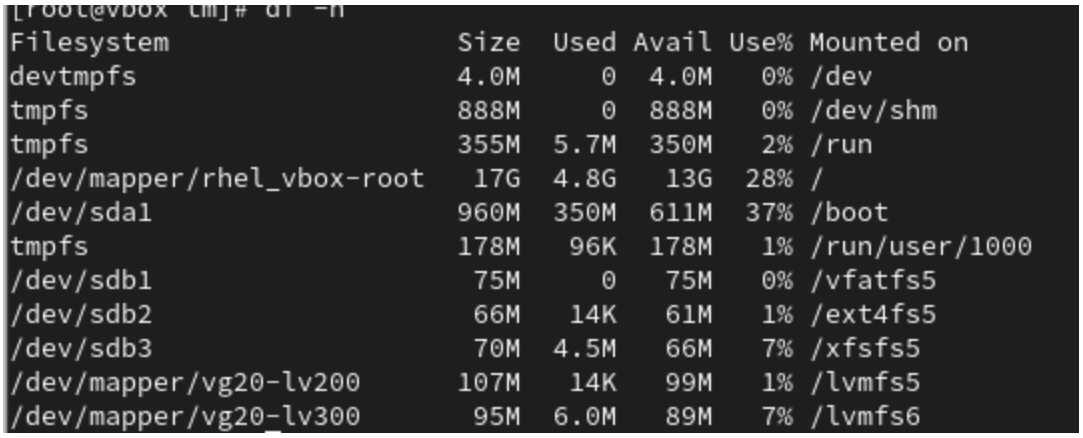
15.2 Create XFS in VDO volume and mount persistently

sudo subscription-manager repos --enable=rhel-9-for-x86_64-baseos-rpms

--enable=rhel-9-for-x86_64-appstream-rpms

vdo create —name=vdo5 —device=/dev/sdc —size=1G —slab_size=1M
vdostatus
vdostats
mkfs.xfs /dev/mapper/vdo5
mkdir /vdofs5
nlkid /dev/mappers/vdo5
UUID="<>" /vdofs5 xfs defaults,_netdev 0 0
umount /vdofs5
mount -a
df -h

15.3 Create ext4 and xfs in lvm
pvcreate /dev/sdc
vgcreate vg20 /dev/sdc
lvcreate -L 120M -n lv200 vg20
lvcreate -L 100M -n lv100 vg20
mkfs.ext4 /dev/vg20/lv200
mkfs.xfs /dev/vg20/lv300
mkdir /lvmfs5 /lvmfs6
mount /dev/vg20/lv200 /lvmfs5
mount /dev/vg20/lv300 /lvmfs6
blkid /dev/vg20/lv200 >> /etc/fstab
blkid /dev/vg20/lv300 >> /etc/fstab
umount /lvmfs5 /lvmfs6
mount -a
df -h



```
[root@vbox tm]# df -h
Filesystem                  Size  Used Avail Use% Mounted on
devtmpfs                    4.0M     0  4.0M   0% /dev
tmpfs                       888M     0  888M   0% /dev/shm
tmpfs                       355M  5.7M  350M   2% /run
/dev/mapper/rhel_vbox-root   17G  4.8G   13G  28% /
/dev/sda1                   960M  350M  611M  37% /boot
tmpfs                       178M   96K  178M   1% /run/user/1000
/dev/sdb1                    75M     0   75M   0% /vfatfs5
/dev/sdb2                    66M   14K   61M   1% /ext4fs5
/dev/sdb3                    70M  4.5M   66M   7% /xfsfs5
/dev/mapper/vg20-lv200      107M   14K   99M   1% /lvmfs5
/dev/mapper/vg20-lv300       95M  6.0M   89M   7% /lvmfs6
```

15.4 Extend ext4 and xfs in lvm

pvcreate /dev/sdd

vgextend vg20 /dev/sdd
lvextend -L +80m /dev/vg20/lv200 (additional)
lvextend -L 300m /dev/vg20/lv300 (final)
/etc/fstab modification is not necessary

```
[root@vbox tm]# lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda                   8:0    0    20G  0 disk
├─sda1                8:1    0     1G  0 part /boot
└─sda2                8:2    0    19G  0 part
  ├─rhel_vbox-root  253:0    0    17G  0 lvm  /
  └─rhel_vbox-swap  253:1    0     2G  0 lvm
sdb                   8:16   0 312.2M  0 disk
├─sdb1                8:17   0    75M  0 part /vfatfs5
├─sdb2                8:18   0    76M  0 part /ext4fs5
└─sdb3                8:19   0    75M  0 part /xfsfs5
sdc                   8:32   0 399.7M  0 disk
├─vg20-lv200        253:2    0   320M  0 lvm  /lvmfs5
└─vg20-lv300        253:3    0   300M  0 lvm  /lvmfs6
sdd                   8:48   0 399.7M  0 disk
├─vg20-lv200        253:2    0   320M  0 lvm  /lvmfs5
└─vg20-lv300        253:3    0   300M  0 lvm  /lvmfs6
sde                   8:64   0 399.7M  0 disk
sdf                   8:80   0 399.7M  0 disk
sr0                  11:0    1  1024M  0 rom
```

15.5 create xfs in stratis volume
stratis pool create strpool5 /dev/sdg
stratis filesystem create strfs5
stratis pool list
stratis filesystem list
mkdir /strfs5
mkfs.xfs /dev/stratis/strpool5/strfs5
mount /dev/stratis/strpool5/strfs5 /strfs5
blkid /dev/stratis/strpool5/strfs5 >> /etc/fstab

```
[root@vbox tm]# mount /dev/stratis/strpool5/strfs5 /strfs5
mount: (hint) your fstab has been modified, but systemd still uses
       the old version; use 'systemctl daemon-reload' to reload.
[root@vbox tm]# systemctl daemon-reload
[root@vbox tm]# lsblk
NAME                                                                                  MAJ:MIN RM   SIZE RO TYPE    MOUNTPOINTS
sda                                                                                     8:0    0    20G  0 disk
├─sda1                                                                                  8:1    0     1G  0 part    /boot
└─sda2                                                                                  8:2    0    19G  0 part
  ├─rhel_vbox-root                                                                    253:0    0    17G  0 lvm     /
  └─rhel_vbox-swap                                                                    253:1    0     2G  0 lvm     [SWAP]
sdb                                                                                     8:16   0 312.2M  0 disk
├─sdb1                                                                                  8:17   0    75M  0 part    /vfatfs5
├─sdb2                                                                                  8:18   0    76M  0 part    /ext4fs5
└─sdb3                                                                                  8:19   0    75M  0 part    /xfsfs5
sdc                                                                                     8:32   0 399.7M  0 disk
├─vg20-lv200                                                                          253:2    0   320M  0 lvm     /lvmfs5
└─vg20-lv300                                                                          253:3    0   300M  0 lvm     /lvmfs6
sdd                                                                                     8:48   0 399.7M  0 disk
├─vg20-lv200                                                                          253:2    0   320M  0 lvm     /lvmfs5
└─vg20-lv300                                                                          253:3    0   300M  0 lvm     /lvmfs6
sde                                                                                     8:64   0 399.7M  0 disk
sdf                                                                                     8:80   0 399.7M  0 disk
sdg                                                                                     8:96   0   1.1G  0 disk
└─stratis-1-private-d029a98f8495483581bc1316909e58bf-physical-originsub               253:4    0   1.1G  0 stratis
  ├─stratis-1-private-d029a98f8495483581bc1316909e58bf-flex-thinmeta                  253:5    0     4M  0 stratis
  │ └─stratis-1-private-d029a98f8495483581bc1316909e58bf-thinpool-pool                253:8    0   636M  0 stratis
  │   └─stratis-1-d029a98f8495483581bc1316909e58bf-thin-fs-d484bffb8ccc4117a644dcb18442e5b7 253:9    0     1T  0 stratis /strfs5
  ├─stratis-1-private-d029a98f8495483581bc1316909e58bf-flex-thindata                  253:6    0   636M  0 stratis
  │ └─stratis-1-private-d029a98f8495483581bc1316909e58bf-thinpool-pool                253:8    0   636M  0 stratis
  │   └─stratis-1-d029a98f8495483581bc1316909e58bf-thin-fs-d484bffb8ccc4117a644dcb18442e5b7 253:9    0     1T  0 stratis /strfs5
  └─stratis-1-private-d029a98f8495483581bc1316909e58bf-flex-mdv                       253:7    0   512M  0 stratis
sr0                                                                                    11:0    1  1024M  0 rom
```

15.6 create swap in partition and lvm volume and activate persistently

parted /dev/sde mklabel msdos
parted /dev/sde mkpart primary 1 100
parted /dev/sde mkpart primary 101 200
mkswap /dev/sde1 -L swapp1
lsblk -f /dev/sde >> /fstab

swapon -a && swapon -s

pvcreate /dev/sde2
vgextend vg20 /dev/sde2
lvcreate -L 80m -n swapvol vg20
mkswap /dev/vg20/swapvol
mkswap /dev/sde2
lsblk -f /dev/sde2 >> fstab (none swap defaults 0 0)
swapon -a && swapon -s

Glossary:
block: collection of bytes of data transmitted as a single unit.
block device file: file associated with devices that transfer data randomly in blocks(disk).
chrony: implementation of Network Time Protocol for time synchronization on network devices.
CIDR: classless inter domain routing
DAC(SELinux): Discretionary Access Control(set of traditional access controls)
De-deplication: removal of redundant data blocks
De-encapsulation: Reverse of encapsulation
Defunct process: zombie process
firewalld: a dynamic firewall manager
firewalld zone: a method of segregating incoming network traffic
globbing: regex
GPG: Gnu privacy guard. open source implementation of PGP(pretty good privacy).
host table: maintains UP and hostname mappings.
index node: hold's file properties: permissions, size, creation/modification time pointer to the data blocks that is stored
IPC: inter-process comm(data sharing, synchronization, resource sharing). Types: Pipes, message queue, shared memory, semaphores, sockets, signals, remote procedure calls.
job: process started in the background
journaled file system: file system that uses the journaling mechanism for a swift recovery after crash.
kerberos: authentication over unsecure networks.
logical extent: a unit of space allocation for logical volumes in lvm
logical volume: logical container in lvm that holds a file system or swap
masquerading: SNAT
mbr(master boot record): small region that stores disk partition information.
named group: specific group that receives ACLs.
named pipe: two unrelated process (on a same/separate system) exchange data.
named user: specific user that receives ACLs.
NDP: neighbor discovery protocol
nftables: packet classification framework to monitor network traffic.
octal mode: method for setting permissions on a file or directory using octal numbering system.

on-demand activation: systemd way of activation of a service when needed.
orphan process: alive child process of a terminated parent process.
package integrity: complete and error-free package
paging: process of transferring data between memory and swap space.
pam: :pluggable authentication module
pattern matching: regex. metachars: chars that have special meaning for machines.
public key encryption: asymmetric encryption.
forward proxy: content filtering, anonymity, internet access behind firewalld (shodowsocks, squid, privoxy, tor, proxychains)
reverse proxy: load balancing, ssl termination, security caching
sticky bit: disable non-owners to delete files.
symmetric encryption technique : technique that employs secret key for private communication between two network entities.
target: logical collection of systemd units. all units within a target are treated as a single entity.
unit: systemd object to organize service startups, socket creation etc.
thin pool: pool of storage allows to create volumes larger than actual size.
thin provisioning: economical technique of storage allocation and utilization.
trashing: excessive amount of paging.
udevd: dynamic device management service
vfat: virtual file allocation table
wayland: superior networking protocol that has replaced the x window system.
workload: any application that runs on the system.
zero-block elimination: technique removing empty data blocks from storage.
hardware-assisted virtualization will have either the **vmx** (Intel) or **svm** (AMD) flags in the */proc/cpuinfo*.