OBJECTIVES

Saturday, November 11, 2023 1:07 PM

EX200V9K

- Understand and use essential tools
  - Access a shell prompt and issue commands with correct syntax
  - Use input-output redirection (>, >>, |, 2>, etc.)
  - Use grep and regular expressions to analyze text
  - Access remote systems using SSH
  - Log in and switch users in multi-user targets
  - Archive, compress, unpack, and uncompress files using tar, star, gzip, and bzip2
  - Create and edit text files
  - Create, delete, copy, and move files and directories
  - Create hard and soft links
  - List, set, and change standard ugo/rwx permissions
  - Locate, read, and use system documentation including man, info, and files in /usr/share/doc
- Create simple shell scripts
  - Conditionally execute code (use of: if, test, [], etc.)
  - Use Looping constructs (for, etc.) to process file, command line input
  - Process script inputs ($1, $2, etc.)
  - Processing output of shell commands within a script
- Operate running systems
  - Boot, reboot, and shut down a system normally
  - Boot systems into different targets manually
  - Interrupt the boot process in order to gain access to a system
  - Identify CPU/memory intensive processes and kill processes
  - Adjust process scheduling
  - Manage tuning profiles
  - Locate and interpret system log files and journals
  - Preserve system journals
  - Start, stop, and check the status of network services
  - Securely transfer files between systems
- Configure local storage
  - List, create, and delete partitions on GPT disks
  - Create and remove physical volumes
  - Assign physical volumes to volume groups
  - Create and delete logical volumes
  - Configure systems to mount file systems at boot by universally unique ID (UUID) or label
  - Add new partitions and logical volumes, and swap to a system non-destructively
- Create and configure file systems
  - Create, mount, unmount, and use vfat, ext4, and xfs file systems
  - Mount and unmount network file systems using NFS
  - Configure autofs
  - Extend existing logical volumes
  - Create and configure set-GID directories for collaboration
  - Diagnose and correct file permission problems
- Deploy, configure, and maintain systems
  - Schedule tasks using at and cron
  - Start and stop services and configure services to start automatically at boot
  - Configure systems to boot into a specific target automatically
  - Configure time service clients
  - Install and update software packages from Red Hat Content Delivery Network, a

remote repository, or from the local file system
    - Modify the system bootloader
- Manage basic networking
    - Configure IPv4 and IPv6 addresses
    - Configure hostname resolution
    - Configure network services to start automatically at boot
    - Restrict network access using firewall-cmd/firewalld
- Manage users and groups
    - Create, delete, and modify local user accounts
    - Change passwords and adjust password aging for local user accounts
    - Create, delete, and modify local groups and group memberships
    - Configure privileged access
- Manage security
    - Configure firewall settings using firewall-cmd/firewalld
    - Manage default file permissions
    - Configure key-based authentication for SSH
    - Set enforcing and permissive modes for SELinux
    - List and identify SELinux file and process context
    - Restore default file contexts
    - Manage SELinux port labels
    - Use boolean settings to modify system SELinux settings
    - Diagnose and address routine SELinux policy violations
- Manage containers
    - Find and retrieve container images from a remote registry
    - Inspect container images
    - Perform container management using commands such as podman and skopeo
    - Build a container from a Containerfile
    - Perform basic container management such as running, starting, stopping, and listing running containers
    - Run a service inside a container
    - Configure a container to start automatically as a systemd service
    - Attach persistent storage to a container

INTRO

Saturday, September 30, 2023 11:19 AM

Ls -a -> list hidden files as well
Ls -d dirname -> list dirname
Mkdir -p A/B/C -> created directory recuresively or insdie another one
Rmdir dirname -> cant use without empty directory
Rm -r dirname -> to remove even if it is not empty
Cp source1 source2 source3 destination1 -> same for mv command
Find / -name passwd - > / means root tree -nam emeans with the name of file (see the man or the use of find we can access files last accessed 20 min ago and so many stuffs)
Man find

Find . -user root -> files created byuser root
Sudo find / -name passwd -exec cp {} /mnt/copy \; -> whatever the find returns those will go to cp command and go to destination directory mentioned

Ls /etc | wc -> gives number of line number of words number of characters (-l to be specific)

Head -> command to get first 10 output

Redirect output:
Ls -l > out.txt
Ls -l 2> out.txt -> error redirection

Chmod u+x file.txt -> usergroup add execute permmission (relative permissions)
o-r (others remove read)

Absolute permissions will be based on number (read -4 write 2 execute 1)
Lets say user rw group wx others x then user -> 4+2=>6 group-> 2+1=>3 others 1 => 631 then chmod 631 out.txt

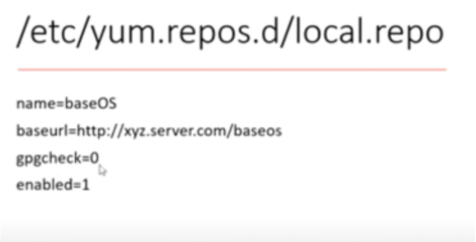
We can use umask command to specify what permissoins are not allowed when creating new files or directories

Umask 024 -> for user not restrictions for group only write not allowed and others read not allowed

Tar cvf target.tar file1 file2 (c-create archive,v-see what is happening in output,f-specify files) (in the place if files we can also use /etc or any directory to compress)
Tar tf target.tar (to view the files inside it)
Tar xf cmp.tar (to extract)

-a to auto-comporess it -j for bzip2 -J for xz -z for gzip(tar.gz)

Configuring repositories:



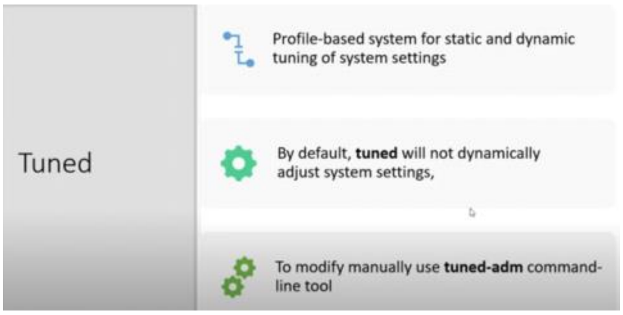
/etc/yum.repos.d/local.repo

```
name=baseOS
baseurl=http://xyz.server.com/baseos
gpgcheck=0
enabled=1
```

In this the baseurl has the packages that we can have to install them
Name=baseOS and also for appStream 4 entries for each

Need to create a file named in the screenshot and add these 4 entries for each name

Yum install packagename,yum update kernel(to update kernel)

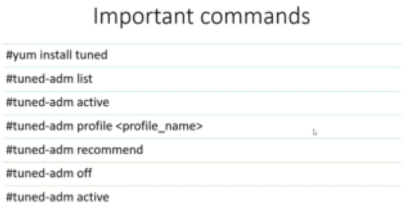


Tuned

Profile-based system for static and dynamic tuning of system settings

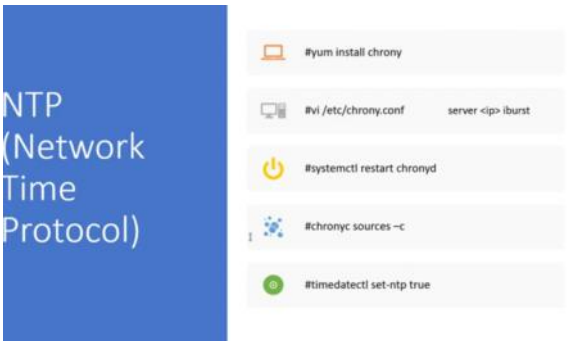
By default, **tuned** will not dynamically adjust system settings,

To modify manually use **tuned-adm** command-line tool



Important commands

```
#yum install tuned
#tuned-adm list
#tuned-adm active
#tuned-adm profile <profile_name>
#tuned-adm recommend
#tuned-adm off
#tuned-adm active
```

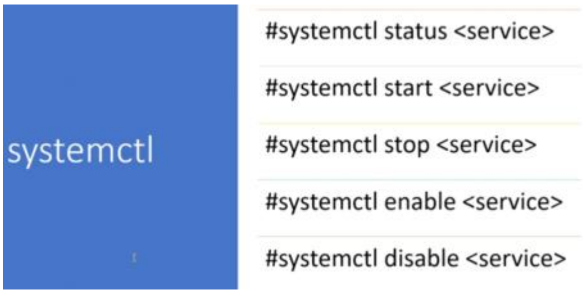


NTP (Network Time Protocol)

```
#yum install chrony
#vi /etc/chrony.conf        server <ip> iburst
#systemctl restart chronyd
#chronyc sources –c
#timedatectl set-ntp true
```

Chrony sources -c not chronyc

Uncomment the pool statement In the top of /etc/chrony.conf and give the server ip iburst



systemctl

```
#systemctl status <service>
#systemctl start <service>
#systemctl stop <service>
#systemctl enable <service>
#systemctl disable <service>
```

groupadd groupname -> creates a linux group
cat /etc/group -> contains the groups available in linux
useradd username -G groupname -> adds the user to the group(first of all with the same username a group is created this will add a secondary group with the given groupname for this user)
cat /etc/passwd -> contains the users

useradd nitin -s /sbin/nologin -> user created without login shell for the user

passwd nitin -> setup password for the user

chown nitin /var/fstab -> change the user owner for a file
Chown :Mac /var/fstab -> change groupowner (with chown -R it will change to all existing files under that directory)

setfacl -m u:username:rw- /var/fstab -> (only rw for this file)setting acl permissions for speicfic users
getfacl /var/fstab -> get the acl entries for that file

setfacl -m g:Mac:--- /var/fstab -> for group Mac should not have rwx on that fie

chmod g+s /linux -> changig the future files under /linux as a group as default

chmod +t /linux -> no user other than the user-owner is able to delte from linux folder

lsblk -> show disks

ls /dev -> lists all directories including the files which represents the disks partitions like dsa1,dsa2 nvme0n1p1,n1p2

fdisk /dev/nvme0n2 (disk for partitioning)
Command: n (add a partition)
Command: p (primary partition,e-extended)
Command: 1 (partition no)
First Sector: Enter
Last sector: +1G (the partition size)
Command: w (partition table will be saved)

# partprobe /dev/nvm10n2 (saved)

mkfs.xfs /dev/nvmen2p1 (format the partition created)
# mount /dev/partitionname /folder (mounts the partition to the folder where you need to mount)

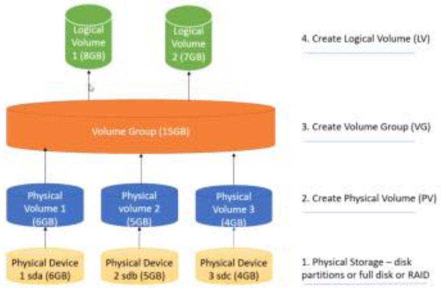
# TO PERSIST THE MOUNT #
vi /etc/fstab
/dev/partitionname TAB /foldername TAB format(xfs) TAB defaults TAB 0 TAB 0
mount -a (successfully written to fstab file)

# CREATE SWAP PARTITION #
free -m (free mem,swap available)
fdisk /dev/nvme0n2 > n > p > ENTER > +750M > t > partitionno > L > 82(linux swap) > w > partprobe /etc/vme0n2

```
mkswap /dev/nvme0n2p3  (format)
vi /etc/fstab (put the swap partition created)
/dev/partname swap swap default 0 0
swapon -a
free -m
```

It's a strategy to mitigate memory starvation when a system is running out of memory. To employ it, Linux moves or swaps out blocks of non-critical memory to disk and swaps them back in on demand

# LOGICAL VOLUME MANAGEMENT #


Logical Volume 1 (8GB)
Logical Volume 2 (7GB)
4. Create Logical Volume (LV)
Volume Group (15GB)
3. Create Volume Group (VG)
Physical Volume 1 (6GB)
Physical volume 2 (5GB)
Physical Volume 3 (4GB)
2. Create Physical Volume (PV)
Physical Device 1 sda (6GB)
Physical Device 2 sdb (5GB)
Physical Device 3 sdc (4GB)
1. Physical Storage – disk partitions or full disk or RAID

```
lsblk -> lists the physical disks
```

2. Create PV

```
pvcreate /dev/nvme0n3
pvcreate /dev/nvme0n4
pvcreate /dev/nvme0n5

pvs -> lists PV's
```

3. Volume Group

```
vgcreate VG1 /dev/nvme0n3 /dev/nme0n4 -> club multiple PV into VG

vgs -> list VG's
```

4. Logical Volume

```
lvcreate -L 8Gb -n LV1 VG1 -> LV of size 8gb with name LV1 from VG1

lvs -> lists LV's

vi /etc/fstab > /dev/VG1/LV1 /mountpointdir xfs defaults 0 0
mkdir mountpointdir
mkfs.xfs /dev/VG1/LV1
mount -a
```

EXTEND LV :
lvextend -r(installs the file system on new space) -L +2Gb /dev/VG1/LV1

EXTEND VG:

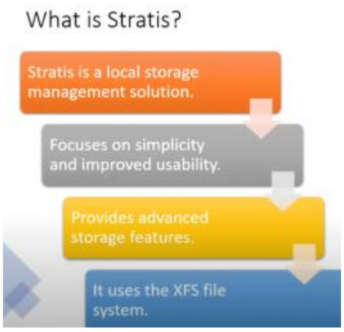
```
vgextend VG1 /dev/nvme0n6
lvextend -r -L +2Gb /dev/VG1/LV1

vgdisplay -> displays the infor of VG's which has the PE size
```

REMOVE:
Comment the /etc/fstab line for this and do unmount /lv,lvchange -an /dev/VG1/LV1

```
lvremove /dev/VG1/LV1
vgremove VG1

vgcreate -s 8M VG1 /dev/nvme0n3
```

The PE size is the basic unit used to create logical volumes. It defines the minimum size of a volume and the possible increments.
For example, if the PE size is 4 MB, the minimum volume size is 4 MB and it can be grown in 4 MB increments.
 The default PE size of LVM is 4 MB

Lvcreate -l 10 -n LV2 /dev/VG1 (in the previous command we created the extend size and here we give the extend increments so 8* 10 => 80 Mb SIZE)



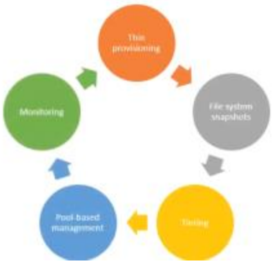
What is Stratis?

Stratis is a local storage management solution.

Focuses on simplicity and improved usability.

Provides advanced storage features.

It uses the XFS file system.



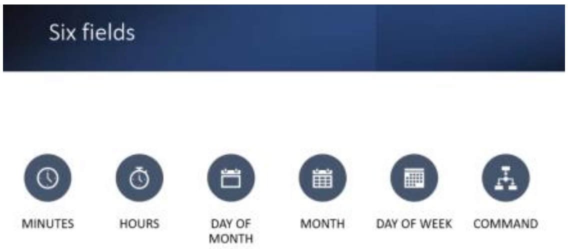
Components of a stratis volume:

1. Blockdev (min size is 1Gb)
2. Pool
3. Filesystem

```
# yum install stratis-cli stratisd
# systemctl start,enable stratisd
# stratis pool create poo1 /dev/nvme0n5
# stratis pool list
# stratis pool add-data pool1 /dev/nvme0n6
# stratis filesystem create pool1 fs1
# stratis filesystem list (get the UUID from the output to put this in fstab file_
# mkdir /ds1
# vi /etc/fstab > UUID=uuidoffs /fs1 xfs defaults,x-systemd.required=stratisd.service 0 0
# mount -a
```

```
# VIRTUAL DATA OPTIMIZER # (provides thin provisioning ,deduplication etc.. )

# yum install vdo kmod-kvdo
# 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 default,x-systemd.required=vdo.service 0 0  > mount  -a
```

CRON:



```
# crontab -e
0 10 4 2 * /usr/local/bin/backup
(save it)

# crontab -e -u username (for particular username)
08 12 * * THU /bash/echo hello
(save it)
```

GREP:

```
# grep -i "root" /etc/groups ->(-i is for case insensitive)ytho
```

```
# RESET ROOT PASSWORD #
After login to the vm need to press down arrow to stop booting and press e for the first option
go to the line where linux is the starting one and go to its end and type rd.break
Ctrl+x to start > # mount -o remount,rw /sysroot > # chroot /sysroot > # passwd (set the new password)
Ch-4# touch /.autorelabel > exit > exit
```

```
#useradd test > # passwd test (change password for test user)
```

```
# NETWORKING #
# ip addr show (gives you interfaces and its ip)
# ip addr show ens160 (interface)
```

CONFIGURING NEW CONNECTION:

```
# nmcli con add
```

MODIFY EXISTING CONNECTION:

```
# nmcli con mod
# nmcli con show (also use the interface name at last for detail info on that)
```

Parammeters used:

```
# con-name,type,ifname,autoconnect,ip4,gw4
```

```
# nmcli con add con-name "Default" type Ethernet ifname ens160 ip4 192.168.1.1/24 gw4 192.168.1.2
# nmcli con up "Default"
( if previously there was any con of type ethernet (it will be ) it will be deactivated and this will be activated)

# nmcli con mod "Default" connection.autoconnect yes (after restart autoconnect to this network (ipv4.addresses when creating we
use ip4 but when modification we use the con show command and get the argument that we can use for modification)
```

To add multiple ip addresses use +ipv4.addresses 192.168.3.3/24 on con mod

```
# CHANGING THE HOSTNAME #
# hostnamectl (current hostname of the system)
# hostnamectl set hostname Ram
```

# SELINUX CONTEXT #

```
#ls -lZ /var/www/html/index.html
```

(shows the long listign with the SELinux context)
Lsts say the apache server is only able to access within /var/www/html is because the selinux policies are applied to only that folder so if we change the selinux context on other folder it will be able to access

(for apsche the default configuration ies here /etc/httpd/conf/httpd.conf)

```
# semanage fcontext -a -t httpd_sys_content_t(apache selinux context) "/NEW(.*)?"
# restorecon -Rv /NEW
```

MODES:

1. Disables
2. Permissive
3. Enforcing

```
# getenforce
# setenforce 0(permissive) 1(enforcing)
Vi /etc/sysconfig/selinux > SELINUX=disabled
```

```
# getsebool -a (all the selinux policy booleans(lists))
# setsebool -P(permissive) httpd_enable_homedirs on (this is one of the selinux policy)
```

To make sure httpd service is accessible at port 82 and should start at boo time we can use this

```
# semanage port -a -t http_port_t -p tcp 82
```

# CONTAINERS USING PODMAN #

```
# yum install podman @container-tools
# podman login registry.rehat.io > username > password
# podman search httpd
# podman pull docker.io/library/httpd
# podman images (lists images pulled/downloaded)
# podman rmi httpd ( remove images)
# podman run -d(detached mode) --name web1(name of container) imageid ( -p 8080:80 )
# podman ps (lists containers running)
# podman stop web1(containername)
# podman rm web1(containername)
# podman run -it imageid /bin/bash (run commands inside the container)
```

```
# mkdir /web
# touch /web/mypage.html
# podman run -d --name web4 -p 8080:80 -v /web:/usr/local/apache2/htdocs imageid
```

(to map local directory to container's directory)

```
# podman generate systemd web5 > /etc/systemd/system/web5-container.service
```

(this will create container as a service

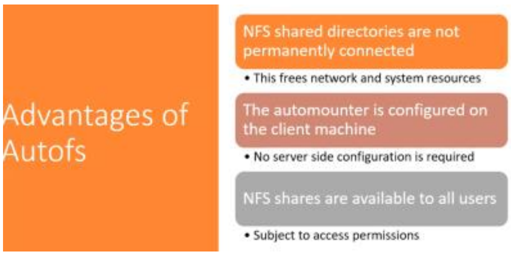
```
# systemctl daemon-reload
# systemctl start web5-container
```

(to login to a users using ssh)

```
# ssh username@localhost
# mkdir -p ~/.config/systemd/user
# podman generate systemd web5 > ~/.config/systemd/user/servicename.service
```

(inside that file on las on [Install] section WantedBy=default.service)

```
# systemctl --user daemon-reload
# systemctl --user start servicename
```



Advantages of Autofs

NFS shared directories are not permanently connected

- This frees network and system resources

The automounter is configured on the client machine

- No server side configuration is required

NFS shares are available to all users

- Subject to access permissions

(server)

```
# yum install nfs*
# mkdir /shared
# vi /etc/exports > /share ipofclient(ro,sync)
# exportfs -avr
# firewall-cmd --add-service=(nfs,mountd,rpc-bind) --permanent
# firewall-cmd --reload
```

(client)

```
# yum install nfs-utils autofs
# showmount -e serverip
# vi /etc/auto.master > /auto-mount /etc/auto.misc --timeout=20
# vi /etc/auto.misc > mountpoint -rw,soft,intr serverip:/share (mountpoint is like within the access folder another folder)
# systemctl enable autofs --now
```

```
# yum install openssh-server (on both server and client)
# systemctl status sshd
# ssh username@userip
```

# SHELL SCRIPTING #

```
# date
# cal(calendar)
```

```
For((i=1;i<=10;i++))
Do
Echo "hello"
Done
```

```
# chmod u+x scriptfile
# ./scriptfile
# bash scriptfile (without changing the permissions)
```

# SHE-BANG #

```
#!/bin/sh
#!/bin/bash
```

(tells to what to use interpretor)

```
Var=10
Echo "sss: $var1" (var1 value is replaced in echo)
```

$HOME,$SHELL are environment variables

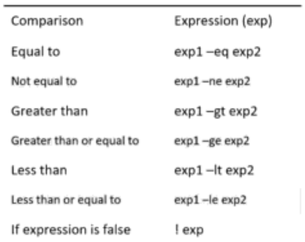
```
echo "enter the name"
read name name1
touch $name $name1
```

(it reads the user input and created a touch file)

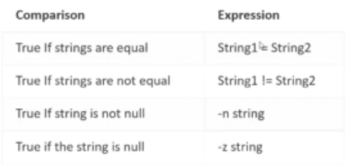
$0 -> name of the script
$1,$2,$3 -> first,second,third argument
$# -> total no of arguments
$* -> value of all arguments

```
If [ condition ]
Then
        Statement1
        …
        Statement n
Elif [condition2]
Then
        statements
Else
        Statement 1-n
Fi
```

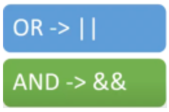
```
If [ $number -gt 10 ]
Then
        Echo ""
Else
        Echo ""
Fi
```

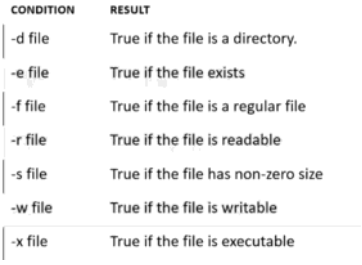


Comparison	Expression (exp)
Equal to	exp1 –eq exp2
Not equal to	exp1 –ne exp2
Greater than	exp1 –gt exp2
Greater than or equal to	exp1 –ge exp2
Less than	exp1 –lt exp2
Less than or equal to	exp1 –le exp2
If expression is false	! exp



Comparison	Expression
True If strings are equal	String1 = String2
True If strings are not equal	String1 != String2
True If string is not null	-n string
True if the string is null	-z string





CONDITION	RESULT
-d file	True if the file is a directory.
-e file	True if the file exists
-f file	True if the file is a regular file
-r file	True if the file is readable
-s file	True if the file has non-zero size
-w file	True if the file is writable
-x file	True if the file is executable

```
If [ -f $1 ] || [ -d $1 ]
Then
        Echo ""
Else
        Echo ""
Fi
```

```
For variable in values
Do
        Statements
Done
```

```
For x in 12 13 14
Do
        Touch $x
Done
```

```
For ((initialiazation;condition;dec/inc))
Do
        Statements
done
```

```
For ((i=0;i<10;i++))
Do
        Date
        sleep 1
done
```

```
Count=0
For x in $*
Do
        If [ -f $x ]
        Then
                Count=$(($count+1))
        Fi
Done
Echo "total: $count"
```

```
x=$(date +"%m")
If [ $x -eq 10 ]
Then
        Echo "month is oct"
Else
        Echo "not"
fi
```

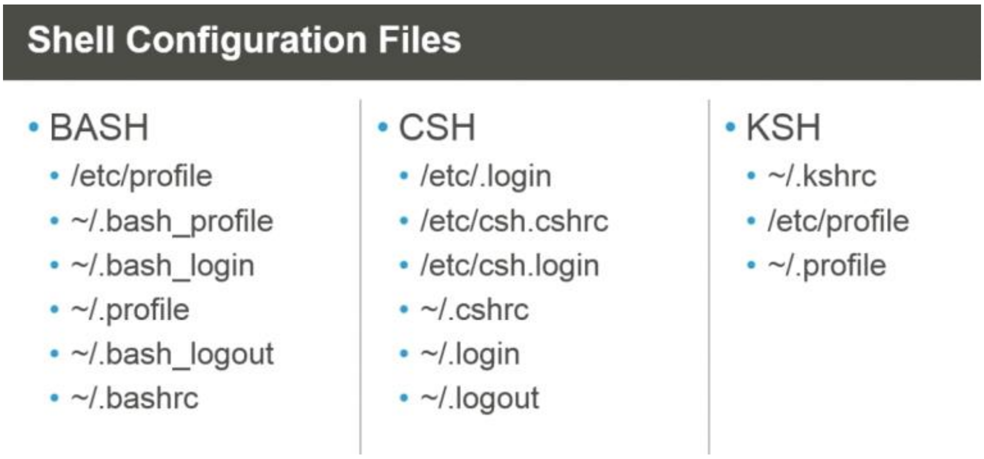
```
For x in * (all the files in current working directory)
Do
        If [ -f $x ] && [ -r $x ] (is a file and readable)
                Echo ""
        Fi
Done
```

VDO ON Lvm:

```
# yum install vdo  kmod-kvdo lvm2
Create PV,VG

# lvcreate --type vdo --name VDO1 --size 5GB --virtualsize 50GB VG1(physical:logicalsize => 1:10)
# mkfs.xfs -K /dev/VG1/VDO1
# mkdir /vdo_m
# vi /etc/fstab > /dev/VG1/VDO1 /vdo_m cfs defaults 0 0
# mount -a
```

COURSE 1 - Part1

Saturday, November 11, 2023 4:13 PM



Shell Configuration Files

- BASH
  - /etc/profile
  - ~/.bash_profile
  - ~/.bash_login
  - ~/.profile
  - ~/.bash_logout
  - ~/.bashrc
- CSH
  - /etc/.login
  - /etc/csh.cshrc
  - /etc/csh.login
  - ~/.cshrc
  - ~/.login
  - ~/.logout
- KSH
  - ~/.kshrc
  - /etc/profile
  - ~/.profile

The /etc/profile > .bash_profile is the order executed when we switch to a user
If we echo something in these two files and exxecute su - username(it will print and will not if - is not given which means - executes the startup scripts)

(prompt modification statement)
PS1="\[\033[34m[\$(date +%H:%M)]\u@\h:\w "
(changing the color of the terminal and format in whci it is beign displayed)

If this needs to be permanent then need to edit this in profile

The .bash_history has command line history (use history command)
!43 (run command 43 from history)

ls -R to recursively list

ls -lt (long listing with modification time)
ls -lS (sort from largest to smallest)

rm -f ~/Downloads/filename (force remove)

Ls -ld /etc (shows the directory itself as a list else it will show all the contents in the directory)

Mkdir -p dir1/dir2/dir3 (create directories within directories)

Rm -r /dir (can use f to force remove directory)

In linux there is something called runlevel where runlevel 5 automatically starts the graphical environment whereas runlevel 3 does not
(runlevel) -> command(give output as 3 5)previos

runlevel currentrunlevel)

gedit to open editor

If we put a & to any command it wil start a backgorund process

jobs -> lists the processes we ran
kill pid -> terminated the process

Man commandname -> gives the manual page for commandname

Man 5 commandname -> (the configuration file man pages)

Info commandname -> infomration of that command

Cat command is to view the contents instead tac command is to view the contents in reverse
More command for stop after screen by screen
Less command to view line by line using down and up arrows

Diff file1 file2 (shows difference in files)

ALIASES, LINKS:

Grep inet -> grep contents containing inet
Grep -v inet6 -> grep contents excluding inet6

Alias is like a command line that we create which executes a command that we give(temp)
Ifconfig | ip add | ipconfig

alias ipadd='ifconfig eth0 | grep inet'
Ipadd

If want to become permanent give this statement in bash profile

LINKS: (like nfs file system)
Linux tracks file using an inode (ls -li -> listing with inode(metadata of file) info,gives info on hard links of the files in 3rd section)
HARD link:
>ln filename ~/foldername/filename
SYMBOLIC LINK:(soft link)
>ln -s filename /dir/filename
(for soft links same inode umber is not given it is seprate inode which means the link file's permissions owning user,group can be changed which cant be done in hard links)
(can also do direcotry links)

For defining a variable permenanetly first VAR="VALUE"
And then export VAR and add it to bash profile file also
can use setenv(setenv VAR "value" command which only
works in specific shells(c shell)
csh -> start c shell

Echo -e "Logged in users \n " $(who) -> here -e is used to
allow \n stuffs

Who > loggedinusers

Sort < filename (sorts the contents given a filename)

Sort < filename > filenames.txt (get the content from
filename and output into filenames)

tr -s " " ":" -> translate -s squeeze the spaces with :
cut -f3 -d ":"-> cut field 3 given a delimiter

Ls *201[34]* -> 2013 or 2014 filenames if 3,4 then it is and

Ls *{2013..2015} -> filenames having from 2013 to 2015

In redhat linux the TTY terminal 1 is used for graphics
when linux bootswith it

:0 is the graphical env that root is logged into
Pts/0 pseudo terminal 0 command prompt itself in GUI
Tty2 for users

```
[Sun Nov 12 ec2-user ~]-->>who
ec2-user pts/0        2023-11-12 09:39 (117.221.15.70)
ec2-user pts/1        2023-11-12 09:39 (117.221.15.70)
```

When logged in via two cmd in windows

Find -name "Project*" -> find names start with Project
(recursively)

Find . -type f -print | wc -l -> find in the current directory
for files alone

Find -user blackwell

Find -cmin -3 (files created within 3 min)

Find -user vlackwell -name"Project*" -exec ls -l {} \;
(execute command after find command)

\; is termination and {} is the placeholder like mv {}

Where in place of {} the filenamewould have come

Find -size +500c -> with size 500c size

Which command to locate the binary program

Which who -> /usr/bin/who
Which ifconfig
/usr/sbin/ifconfig

Locate -S -> shows info on what database and about files directories and stuffs

Locate -ic (case insensitive and count)filename -> gives out the file paths which is there in db

vim
:w /userfile/subnets inside a vim editor to write the content to another file

dd -> remove a line
yy -> copy
p -> paste
:%s/findword/replacword/i -> replace a content case insensitive

Emacs filename-> (emacs editor like gedit)
Ctrlx+ctrlw to save and write
Ctrlk -> cut
Ctrly -> paste
Ctrlx + ctrlc -> close and quit

Sed -n(suppresses automatic printing) '1~2p' (start at line 1 in the file and print every second line) subnets(filename)

Sed '1~2d'(d for delete and p for print) filename

Sed 's/findword/replaceword/g' filename

Sed 's/^.*0//' filename-> start of the beginning of line and any number of characaters upto 0 and replace it with nothing)

Bash scripting:

#!/bin/bash -> which shell

PATH=$PATH:/userfiles/scripts -> whenever we type script,sh which is under the /userfiles/scripts directory the linux will interpret it as a command and execute it

Command & -> background

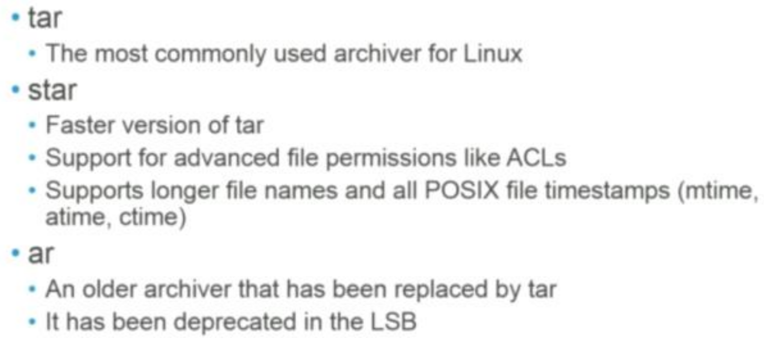
Fg pid -> foreground
Bg pid -> continure the process
ctrlZ -> when foreground to stop

Cat file 2> (error rediredction)

2>&1
This command helps in redirecting the stdout and stderr
in the same file.

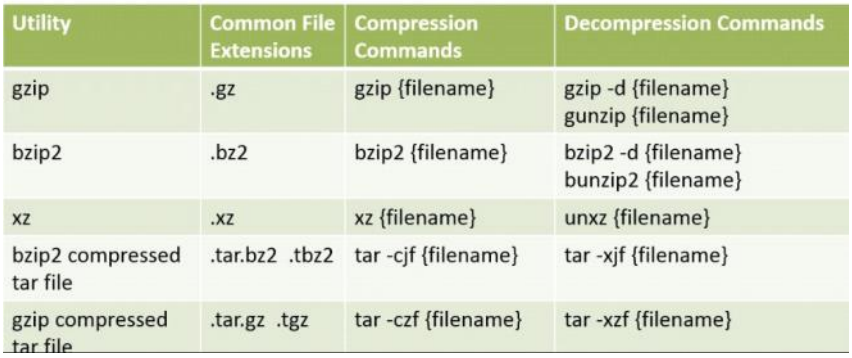
Bash -> spawn a new subshell

Archiving:



- tar
  - The most commonly used archiver for Linux
- star
  - Faster version of tar
  - Support for advanced file permissions like ACLs
  - Supports longer file names and all POSIX file timestamps (mtime, atime, ctime)
- ar
  - An older archiver that has been replaced by tar
  - It has been deprecated in the LSB

Gzip/gunzip,bzip2,rar,xz,7z,arc,xar,zip



Utility	Common File Extensions	Compression Commands	Decompression Commands

gzip	.gz	gzip {filename}	gzip -d {filename}
			gunzip {filename}

bzip2	.bz2	bzip2 {filename}	bzip2 -d {filename}
			bunzip2 {filename}

xz	.xz	xz {filename}	unxz {filename}
bzip2 compressed tar file	.tar.bz2 .tbz2	tar -cjf {filename}	tar -xjf {filename}
gzip compressed tar file	.tar.gz .tgz	tar -czf {filename}	tar -xzf {filename}

```
>gzip filename (adds .gz extension)
>gunzip filename.gz(uncompresses the file)
```

```
>bzip2 * (adds an .bz2 extension)
>bzip2 -d * (decompress)
>bzip2 -v(verbose) -9(best possible compression) *.sh
>bunzip2 *.bz2 (decompress)
```

```
>tar -cvf(create,verbose,filename) users.tar /userfiles (recursively under the directory given)
>tar -cvzf(gzip compression) users.tar.gz /folder
>tar -tvf(view in archive,verbose,filename) users.tar
>tar -zxvf(extract) users.tar.gz
```

```
>star -c -v -f=/folder/file.star *.sh
>star -t -v -f=users.star(view)
>star -x -f=file.star (extract)
```

```
>zip -r /folder/file.zip foldername
```

```
>unzip -l /folder/file.zip
>zip -d(Delete) bidgets.zip *(delete these files inside zip file)
```

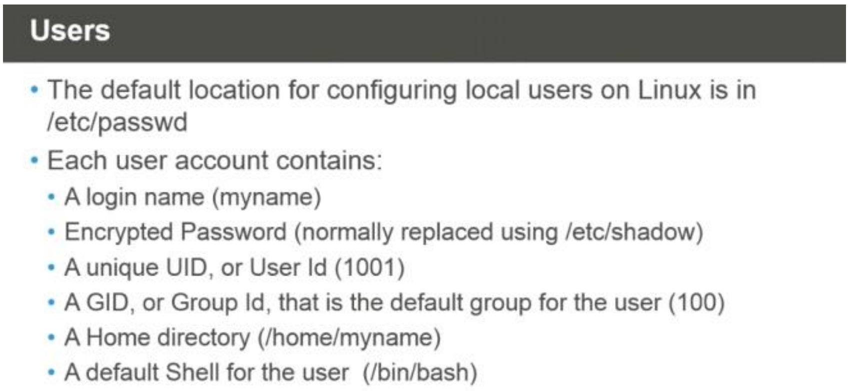
PACKAGE MANAGEMENT:

yum(yellow updater modifier)

```
>yum list | less (list all available softwares on all repos)
>yum search wireshark (show matching packages for this)
>yum install wireshark* (if is there more than one packages to install)
>yum list installed |grep wireshark (list installed packages)

>yum repolist (list of software repositories)
>yum repolist all (all including disabled)

>yum-config-manager --enable "repoid"
>yum-config-manager --disable "repoid"
>yum info wireshark (info on specific package)
>yum update wireshark
>yum update(to update all installed packages)

>yum reinstall wireshark
>yum remove wireshark

>yum swap *vnc* *rdp* (remove all vnc packages and in that place install rdp
packages)

>yum list installed | grep createrepo
>createrepo /directorycontainingpackagefiles/
>vi /etc/yum.repos.d/rhel-local.repo
[custom_repo]
name=Custom Local Repository
baseurl="file:///directorycontainingpackagefiles/"
gpgcheck=0
>yum repolist(enabled repos included ours)
```

USERS AND GROUPS:

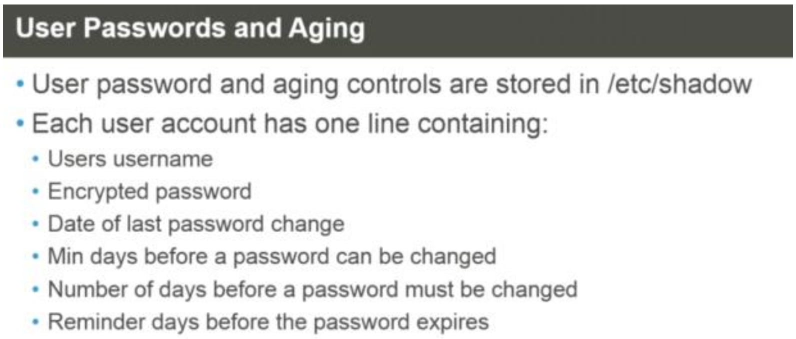
Reserved user account namesL
root,lp,daemon,sys,nobody,gdm
Common groups for users
lp,cdrom,sudo,audio,video,users



Users

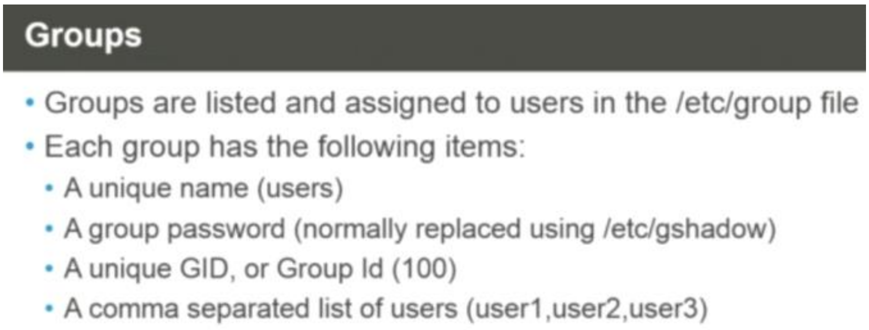
- The default location for configuring local users on Linux is in /etc/passwd
- Each user account contains:
  - A login name (myname)
  - Encrypted Password (normally replaced using /etc/shadow)
  - A unique UID, or User Id (1001)
  - A GID, or Group Id, that is the default group for the user (100)
  - A Home directory (/home/myname)
  - A default Shell for the user (/bin/bash)

Common securty setting s for user are found in /etc/login.defs



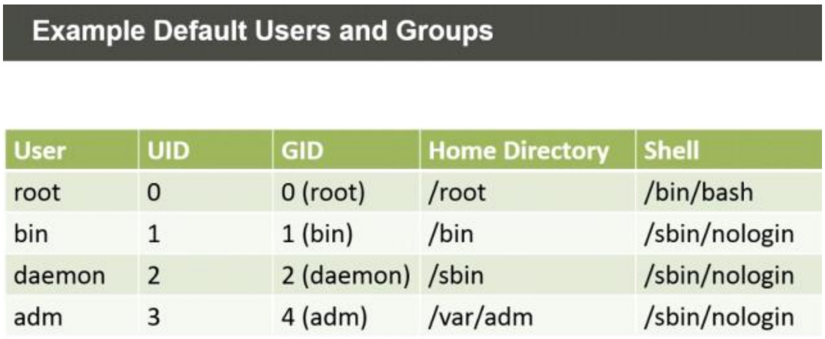
User Passwords and Aging

- User password and aging controls are stored in /etc/shadow
- Each user account has one line containing:
  - Users username
  - Encrypted password
  - Date of last password change
  - Min days before a password can be changed
  - Number of days before a password must be changed
  - Reminder days before the password expires



Groups

- Groups are listed and assigned to users in the /etc/group file
- Each group has the following items:
  - A unique name (users)
  - A group password (normally replaced using /etc/gshadow)
  - A unique GID, or Group Id (100)
  - A comma separated list of users (user1,user2,user3)



Example Default Users and Groups

User	UID	GID	Home Directory	Shell
root	0	0 (root)	/root	/bin/bash
bin	1	1 (bin)	/bin	/sbin/nologin
daemon	2	2 (daemon)	/sbin	/sbin/nologin
adm	3	4 (adm)	/var/adm	/sbin/nologin

passwd (change the password for the user logged in currently)
passwd username (to change the password for a user without knowing the current password when root)

/etc/passwd

Username:x:userid:groupid:UserDescription:userhomedir:loginshell (x means password stored elsewhere usually in /etc/shadow but with /etc/group it measn it has not password)

/etc/shadow

Username:encryptedpassword:lastpasswordchangenoofdays:mindaystochange:maxdaysvalid:nooofdaysbeforeexpiry:absolutedateaccountexpire

If wanted to reset a user password for in /etc/shadow and remove the encryptedpassword by opening the file and save it using :wq! And then it will not password next time

Ctrl_Alt_F2 -> to open up tty 2

```
>useradd username -p PassWorD -c(comment) "User Name"
>tail(last 10 lines) /etc/group (creates a group name same as username)
```

```
>usermod username -s /bin/csh (to make the user use csh)
>userdel username
>usermod username -e 8/1/2015 (expiration date)
```

```
>passwd -e username (immediate expiration)
>passwd username -n 5 (5 days given to change the password)
>passwd username -x 30(max password age 30)
>passwd username -w 2(within 2 days of expiry user will be warned)
```

```
>groupadd hr(create group)
>groupmod hr -n humanresources(rename group)
>groupdel hr
```

```
>usermod username -G groupname(adding to a group can give multiple seprating with comma)
>usermod username -g sales (change primary/default group)
```

>chown to change user group and chgrp to change group

```
Chmod g+w Dir/
```

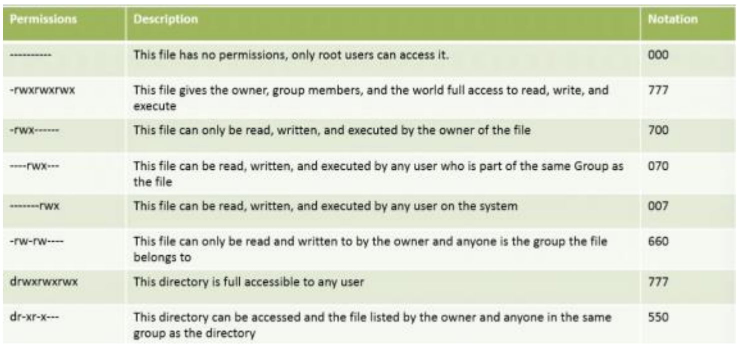
COURSE 1 -Part2

Monday, November 13, 2023 9:38 AM

File permissions:

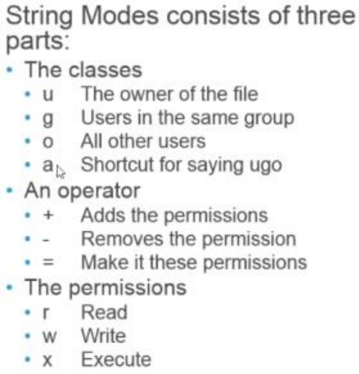
In the file permission in long listing d means directory l means symbolic link - means normal file at last + means acl is applied

Read 4
Write 2
Execute 1



Permissions	Description	Notation
----------	This file has no permissions, only root users can access it.	000
-rwxrwxrwx	This file gives the owner, group members, and the world full access to read, write, and execute	777
-rwx------	This file can only be read, written, and executed by the owner of the file	700
----rwx---	This file can be read, written, and executed by any user who is part of the same Group as the file	070
-------rwx	This file can be read, written, and executed by any user on the system	007
-rw-rw----	This file can only be read and written to by the owner and anyone is the group the file belongs to	660
drwxrwxrwx	This directory is full accessible to any user	777
dr-xr-x---	This directory can be accessed and the file listed by the owner and anyone in the same group as the directory	550

chmod u=rw file.txt (user has read write)



String Modes consists of three parts:

- The classes
  - u The owner of the file
  - g Users in the same group
  - o All other users
  - a Shortcut for saying ugo
- An operator
  - + Adds the permissions
  - - Removes the permission
  - = Make it these permissions
- The permissions
  - r Read
  - w Write
  - x Execute

-R option in chmod recursively applies permission to all files in dir

```
>getfacl filename
>setfacl -m(modify) u:username:r(read) filename
                    g:          rw       .(dir)
```

>vi /etc/fstab (list of mount points)
In the defaults add this ,usrquota,grpquota for the mount point you need to change

>mount -o remount /Data (mount point for which we edited)

>quotacheck -cug(create user group) /Data
>ls -a (lists aquota.group , aquota.user files will be there)
>quotacheck -avug (all verbose user group)

>edquota(edit quota) -u(use -g for group) username
(will show the disk (mount point and properties you can change for blocks soft and hard limit and also for inodes soft means there is a grace period)

```
>quota -u username
```

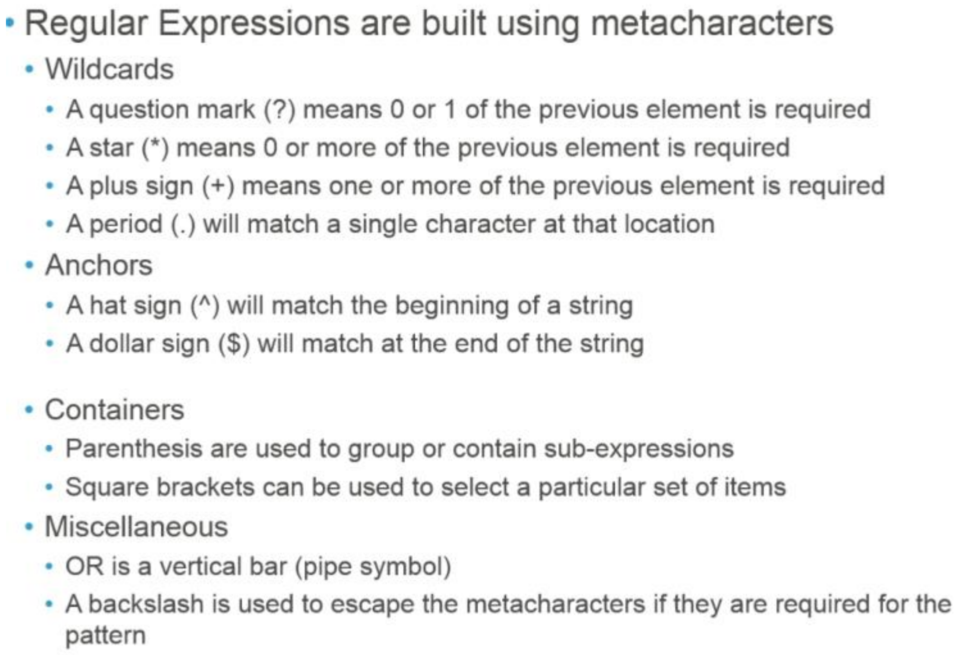
```
>setquota -g groupname 0 5000 0 0 /Data(mount point and 4 is represented as soft and hard link for blocks and inodes)
```

```
>edquota -t
(block , inode grace period editing)
```

```
>quotaon -vug /Data (turned on use quotaoff to turn it off )
>repquota -ug /Data (report)
```

STICKY BIT:

```
>chmod +t Directory (only users owning the file can delete or rename the file)
```



- Regular Expressions are built using metacharacters
  - Wildcards
    - A question mark (?) means 0 or 1 of the previous element is required
    - A star (*) means 0 or more of the previous element is required
    - A plus sign (+) means one or more of the previous element is required
    - A period (.) will match a single character at that location
  - Anchors
    - A hat sign (^) will match the beginning of a string
    - A dollar sign ($) will match at the end of the string
  - Containers
    - Parenthesis are used to group or contain sub-expressions
    - Square brackets can be used to select a particular set of items
  - Miscellaneous
    - OR is a vertical bar (pipe symbol)
    - A backslash is used to escape the metacharacters if they are required for the pattern

```
Grep "word" * -R  (recursively go to each dir and return the result)
```

- ➢ Grep -i(case insensitive)

```
Grep "word" filename | Sort -k 2 (sort column 2)  (-rk means reverse sort)
```

```
Grep "north$" filename (contents where north is end of each line)
```

```
Grep "\$....$" filename (match words with $ as start and 4 digits and end of line)
```

```
>uname -r (list linux kernel version)
```

BASH SCRIPTING:

```
#!/bin/bash
Read varname
Echo "you $varname"
```

```
[root@rhell Scripts]# [ $AGE -gt 65 ] && echo "Senior"
(if conditions in command line itself)
```

```
While [ condition ]  (can use until in place of while)
Do
        Commands
If [ condition ]
Then
        Commands
Elif [ condition ]
Then
        Comands
Else
        Command
Fi
done
```

While executes when condition s true and until executes when it is false

----

```
>man test (test operators)
```

```
>test 10 -gt 5 && echo "True"
>test -e filename (checks its existence) returns 0 succesful 2 not successful
```

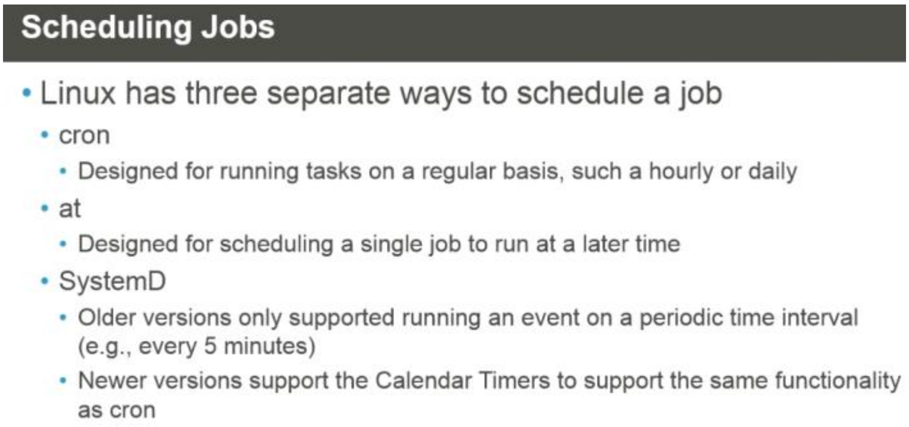
```
>test -w filename (test write access)
```

```
>echo {6..8} (returns 6 7 8)
>echo $((34*20)) (returns the calculation)
```

```
>mkdir 20{09,20,11,12,13,14,15}_suffix (created directories like 2009,2020,2011,etcc.)
```

(if we want to check if previous cmmand was successful or not and use that in condition)

```
If [$? -eq 0]
Then
        Command
Else
        Command
fi
```

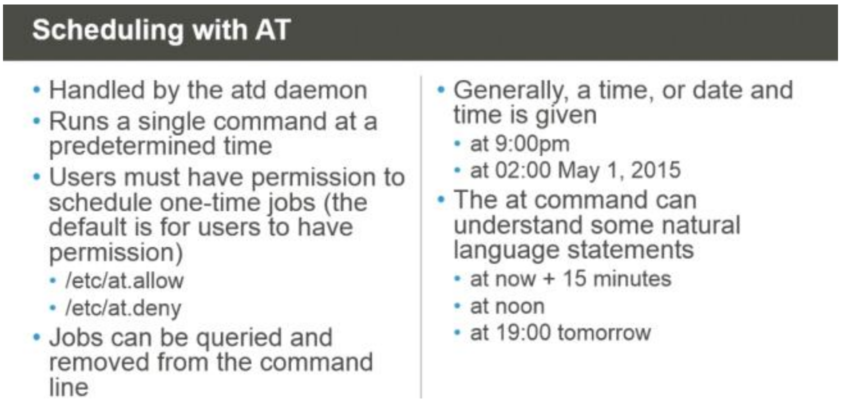


Scheduling Jobs

- Linux has three separate ways to schedule a job
  - cron
    - Designed for running tasks on a regular basis, such a hourly or daily
  - at
    - Designed for scheduling a single job to run at a later time
  - SystemD
    - Older versions only supported running an event on a periodic time interval (e.g., every 5 minutes)
    - Newer versions support the Calendar Timers to support the same functionality as cron

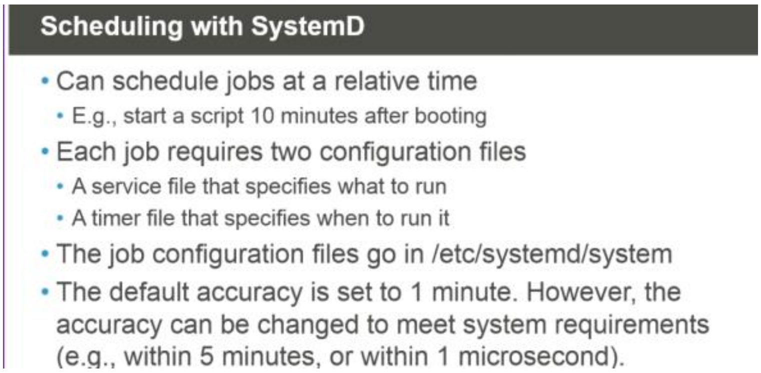
There are two files we can see
/etc/cron.allow will have users who can add cron jobs
/etc/cron.deny file has users with no access to create cron jobs

All jobs are stored in crontab files /var/spool/cron file which only should be edited from crontab command



Scheduling with AT

- Handled by the atd daemon
- Runs a single command at a predetermined time
- Users must have permission to schedule one-time jobs (the default is for users to have permission)
  - /etc/at.allow
  - /etc/at.deny
- Jobs can be queried and removed from the command line
- Generally, a time, or date and time is given
  - at 9:00pm
  - at 02:00 May 1, 2015
- The at command can understand some natural language statements
  - at now + 15 minutes
  - at noon
  - at 19:00 tomorrow



Scheduling with SystemD

- Can schedule jobs at a relative time
  - E.g., start a script 10 minutes after booting
- Each job requires two configuration files
  - A service file that specifies what to run
  - A timer file that specifies when to run it
- The job configuration files go in /etc/systemd/system
- The default accuracy is set to 1 minute. However, the accuracy can be changed to meet system requirements (e.g., within 5 minutes, or within 1 microsecond).

/sbin/service crond status (See status for the crond)

>vi /etc/crontab
(add an entry)
*/2 * * * * username linuxcommands

>crontab -e(edit)
(this we'll run at root so that in the entry no need to specify the username)

history -c (clears the history)

>crontab -l(list user crontabs)
>ls /var/spool/cron (has directories for each user if they have created cronjob)
>crontab -r(Remove)

>/sbin/service crond|atd stop|start
>ps -aux (all user and back ground processes)
>kill -STOP PID (stop/terminate)
>kill -CONT PID (continue)

>ls /etc/at*

```
>at 4:30am (opens a interactive terminal where we can give commands and <EOT> for exiting)
>at noon today
>at 6am +1day (1 day plus today)
>atq (all jobs scheduled)

>at 10pm Jan 30
>at now +1 min (execute 1 min from now after job is submitted)

>at -f /UserFiles/commands.txt now +1 hour

>at -c jobid  (shows what's init)

>atrm jobid(remove job)
```

SYSTEMD:

There should be two files in order to schedule a systemd timer unit or a job

```
Ls /etc/systemd/system
```

```
Script.service
----------------
[Unit]
Description=Some description
#Wants=another.service(to have dependency)

[Service]
Type=simple
ExecStart=/Userfiles/script.sh
#KillMode=process
#Restart=on-failure
#RestartSec=42s

[Install]
WantedBy=multi-user.target (after enable command executes symbolic link command and creates
the same file under this dir(multi-user.target.wants))
```

```
Script.timer
---------------
[Unit]
Description=Some description

[Timer]
#OnBootSec=10min
#OnCalendar=Mon 2015-*-1 10:00:00
#OnCalendar=*:0/15 (every 15 minutes)
OnUnitActiveSec=10s  (execute every 10s)
Unit=script.service
```

```
>man 7 systemd.time (help for syntax)
```

```
>systemctl list-units (list of units or jobs activated)
```

```
>systemctl enable script.service
>systemctl start script.service
```

```
>getfacl ~/* (for all files under home dir)
```

```
>find ~ -maxdepth 1 -executable -type f (find files in only the home dir having execute permissions files)
```

```
>netstat | grep tcp (network statistics for tcp)
```

PROCESSES|DAEMONS:ETC..:

```
>ps
```

```
>pstree | less (hierarchical processes)
```

```
>sort -nk 4 (sorting numerically on column4)
>tail -3 (last three)
```

```
>top (statistics with processes)
```

Can type d inside this to change the delay seconds
Can type f for fields and move the cursor to the field to remove and click space and q
Type s to sort on that field

```
>top -n 5 -b >file.txt (5 iterations)
```

```
>kill -l (lists all the various ways we can send kill signals)
```

SIGTERM is default

```
>kill -9 PID (number listed in the above command)
```

```
>pkill nameofprocess (only pkill can use process name instead of pid)
```

```
>pgrep sshd (gives the pid of sshd)
```

```
>kill -1(SIGHUP) command & (executes even after logout)
```

PRIORITIES TO PROCESSES:

```
>renice priorityno PID
```

```
>ps axl | grep process (will have the priorityno in 6 column (nice value))
```

```
>nice --5 ./script.sh & (it will give the priority 5 initially(
```

```
Ls /proc (directories created for each PID)
Ls /proc/pid
```

```
>cat /proc/cpuinfo
>df -h (info on disks or filesystem)
>watch -n 5 df -h (watch df -h command and run it every 5 seconds)
```

```
>tail -f /var/log/messages
```

```
>top -p pid (just look stats for pid)
```

```
>systemctl status sshd.service
>service sshd status
```

(both are same)

```
>systemctl list-unit-files --type service(timers,mount,automount,path etc..) | less
(list all services) (--all to list all services)
```

```
>systemctl enable sshd.service (it starts when system boots)
```

```
>ls /usr/lib/systemd/system/*service
```

```
>systemctl mask sshd.service ( masking a service is used to prevent the service from starting.)
           unmask
```

To list dependencies of a service:

```
>systemctl list-dependencies sshd.service
```

```
>ls -ld /usr/lib/systemd/system/runlevel*
```

```
>systemctl get-default (lists which runlevel)
>systemctl set-default graphical.target|multi-user.target
```

```
>journalctl (logged information for services)
>journalctl -n 5(latest 5 log entries)
>journalctl -f( like watch)
>journalctl -b (from current boot session)
```

COURSE 1 -Part3

Monday, November 13, 2023 1:44 PM

```
>lspci (lists pci devices known or there)
>route (routing table)
>route -n (numeric listing)
```

```
>cat /etc/resolv.conf (has name server and host info )
```

```
>hostname newname (change hostname)
```

```
>cat /etc/resolv.conf
```

```
>ifconfig eth0 down|up
```

```
>ls /etc/sysconfig/network-scripts
```

(network related files or directories)
Ifcfg-ifname file will be there we can edit that and restart network service and it will take effet

```
>cat /etc/hosts
```

(like a dns resolution in local name to ip entries)

```
>route add -net 200.1.1.0 netmask 255.255.255.0 gw 192.168.1.2 dev ifacename
```

```
>ntpupdate pool.ntp.org (setting a ntp source)
```

```
>cat /etc/ntp.conf | less
```

```
>service ntpd start
>systemctl ntpd enable
```

```
>ntpq -pn
ntpq>peers
```

```
>route del default
>route add default gw 192.168.1.1 dev eth0
```

LOGGING:

```
>ls /var/log
```

```
>lastlog | grep username (lastlogin information)
```

```
>lastlog -t 5(last login info for only last 5 days)
```

```
>lastlog -u username
```

```
>dmesg (linux kernel logs)
>dmesg --level=err
```

```
>ls /run/log/journal/hash/
```
(will contsin a journal file which csn be read only using journalctl command)

```
>cat /etc/systemd/journald.conf
```
(ForwardToSysLog=yes,MaxLevelSyslog=warn)

```
>service rsyslog status

>service systemd-journald status

>journalctl -r (newest entry in the start)
>journalctl --disk-usage

>journalctl --since "14:00"
>journalctl --since yesterday
>journalctl -b(since last reboot) -p err

>vi /etc/logrotate.conf

>ls /etc/logrotate.d
>cat /etc/logrotate.d/yum


>logrotate -fv(force logrotate) /etc/logrotate.d/sample
```

(in sample we give the log file name and properties of it inside curly braces and execute this command)

```
/var/log/messages
{
	Rotate 5 (log files will be rotated 5 times before old ones are removed)
	Size 50k (before rotating 50k size should be there)
	Create 660 root root
	Compress
	Dateext
}
```

```
>vi /etc/rsyslog.conf
```
(rsyslog daemon read this config file)
Here we can give which logs should go to which file
Local4.* /var/log.local.log
(The facilities local0 to local7 are "custom" unused facilities that syslog provides for the user. If a developer create an application and wants to make it log to syslog, or if you want to redirect the output of anything to syslog (for example, Apache logs), you can choose to send it to any of the local# facilities. Then, you can use /etc/syslog.conf (or /etc/rsyslog.conf) to save the logs being sent to that local# to a file, or to send it to a remote server.)

```
>service rsyslog restart

>logger "Hello World" (this will log into /var/log/messages)

>logger -p(priority) local4.info "Message"
```

NETWORK SERVICES:

Service httpd start|enable

DNS SERVER :
==========

```
>yum list installed | grep bind

>/etc/named.conf (dns conf)
(in last line the zone details like filewhich contsins the records)

>ls /var/named (cintains the zone file listed in named.conf)

>service named start|stop
>systemctl enable named.service
```

DHCP SERVER:
===========

```
>yum list installed | grep dhcp
>vi /etc/dhcp/dhcpd.conf  (DHCP conf)

>service dhcpd start
>systemctl enable dhcpd.service
```

In windows:

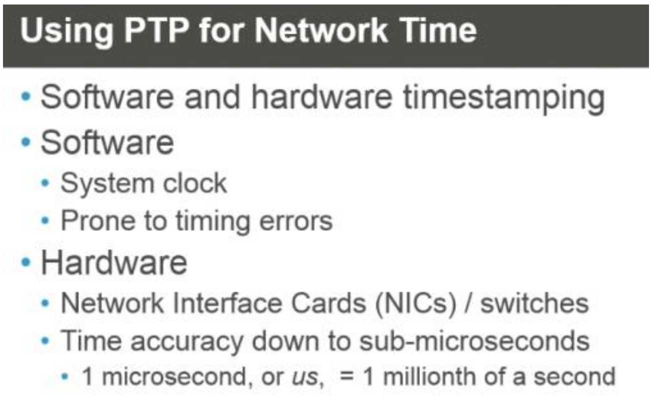
```
>ipconfig /release
>ipconfig /renew
```

NTP:
===

```
>yum list installed | grep ntp
>vi /etc/ntp.conf
( we can give our linux machine's ip in restrict to make it get the nto from ntpd service)
>ntpq -p (list the peers)

>service ntpd status
>systemctl enable ntpd.service
>netstat -anu(all numeric ports for udp)
```

PTP(precision time protocol)
======================

- Synchronize network device clocks on a network
- More accurate than ntp
- Uses udp unicast or multicast transmission
- Master/slave hierarchy



Using PTP for Network Time

- Software and hardware timestamping
- Software
  - System clock
  - Prone to timing errors
- Hardware
  - Network Interface Cards (NICs) / switches
  - Time accuracy down to sub-microseconds
    - 1 microsecond, or *us*, = 1 millionth of a second

```
>yum install linux-ptp
```

```
>ptp4l (configure NIC for software and hardware time stamping)
>phc2sys (synchronize the system clock to the hardware clock on the NIC)
>pmc (PTPT management client,GET AND set ACTIONS)
```

LOGGING SERVER:
==============

```
>vi /etc/rsyslog.conf
(uncomment these
$ModLoad imudp
$UDPServerRun 514
```

Add this under GLOBAL DIRECTIVE

```
$template RemoteHots, "/var/log/%HOSTNAME%/syslog.log" (it is kind of dynamic log entry fcreated from remote hosts)
```

Under RULES put this

```
*.* ?RemoteHost
```

```
>service rsyslog restart
```

In the clients do these

```
/etc/rsyslog.conf
```

Rules:

```
*.* @192.168.1.241:514 (SERVER IP:PORT)
```

SAMBA SERVER:
=============

```
>yum list installed | grep samba
>vi /etc/samba/smb.conf
```

```
(in [global section]
Workgroup = SILVERSIDES (like a domain name)
Server string = RHEL1
Netbios name = RHEL1
)
```

```
>service smb restart
```

```
>testparm (test the conf )
```

```
>smbpasswd -a root (adding a user account called root)
>smbclient -L rhel1 (lists all the shared items)
```

If we want to share a folder then we need to create a section in the conf file of samba like this

```
[FolderName]
        Comment  = sindd
        Path = /FolderName/
        Writable = yes
        Public  = no
```

```
>service smb restart
>getenforce (get the enforcing thing on Selinux)(should be enforcing)
>semanage fcontext -a -t samba_sahre_t "/UserFiles(/.*)?"
>restorecon -R -v /UserFiles
```

SSH:

```
===
>vi /etc/ssh/sshd_config (server)
>vi /etc/ssh/ssh_config (client)
```

HTTPD:

```
=====
>yum list installed | grep http
>vi /etc/httpd/conf/httpd.conf
>service httpd restart
```

TELNET:

```
======
>systemctl start telnet.socket
>netstat -ant

>telnet hostip
```

FTP:

```
===
>yum list installed | grep ftp
(VSFTPD)
>ls /etc/vsftpd/vsftpd.conf|ftpusers|user_list|vsftp_conf_migrate.sh

vsftpd.conf
(local_root=/var/ftp) -> it means the users will be placed under this direcotry when they connect

>service vsftpd restart
>netstat -tan(tcp all numeric) | grep 21(port ftp)

From client
>ftp rhel1(Server name)
Auth
>lcd c:\temp (local direcotry )
>mget *.rpm (it will transfer the files into local cd dir)
```

SQUID:(proxy server)

```
======
>yum list installed | grep squid
>/etc/squid/squid.conf
Cache_dir ufs /var/spool/squid 100 16 256
>service squid restart
>netstat -tan | grep 3128

>cat /var/log/squid/access.log (whatever the clients accesses which came throuhg our squid server)
```

MAIL SERVER:

```
===========
>yum list installed | grep postfix(outbound) , grep dovecot(inbound)
>ls /etc/postfix/
>vi /etc/postfix/main.cf
>service postfix start
>netstat -tan | grep 25

>vi /etc/dovecot/dovecot.conf
>service dovecot start
>netstat -tan | grep 110
```

FIREWALLS:
=========

Iptables are old and firewalld is newest in rhel

```
>firewall-cmd --get-active-zones (list of zones it is enforced like on interfaces)
Zone means block,dmz,drop,external,home,internal,public,trusted,work
>firewall-cmd --zone=trusted --add-port=443/tcp (--permanent entry for permanent entry)
>firewall-cmd --zone=trusted --list-ports
--list-all(list everything)
>firewall-config (GUI)
```

Iptables,ip6tables is the interface to allow users to configure the ipv4,ipv6 firewall

Network packets are compared with rules in chains
Chain contains list of rules

- A rule can DROP ,REJECTACCEPT FORWARD a packet
- Rules can be added or inserted in any chain
- Rules in chain are traversed sequentially until a rule matches a packet

3 predefined chains:

- INPUT
- OUTPUT
- FORWARD

```
>iptables -F (to clear the firewall)
>iptables -L -v (list the firewall chains and rules)
>iptables -A INPUT -p tcp --dport 22 -j ACCEPT(OPEN A PORT TO ALLOW SSH)

>iptables -P(policy) INPUT DROP (drop all packates coming to the current machine)
>iptables -A(add) INPUT -p(protocol) tcp -m(match) tcp --dport 80 -j ACCEPT

>iptables -R(remove) INPUT 1(rules 1 in iptables -L)

>service iptables save(save rules in /etc/sysconfig/iptables file)

>systemctl enable iptables
```

```
>firewall-cmd --get-services (list of services can be configured for allowing) these exist as
/usr/lib/firewalld/services as xml's
```

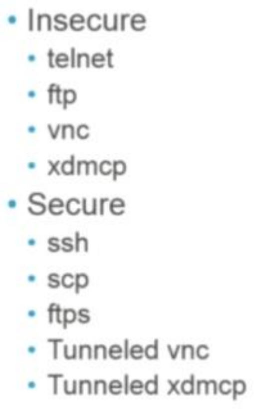
WE CAN ALSO CUSTOMIZE AND CREATE OUR OWN XML

```
>tc -s(stats) qdisc(queing discipline) ls(list) dev(device) eth0
(to see if any rules have been added to limit network traffic)

>tc qdisc add dev eth0 root netem(network emulation) delay 200ms (delay transmission by 200ms)

>tc qdisc del dev eth0 root (to delete the entry)
```

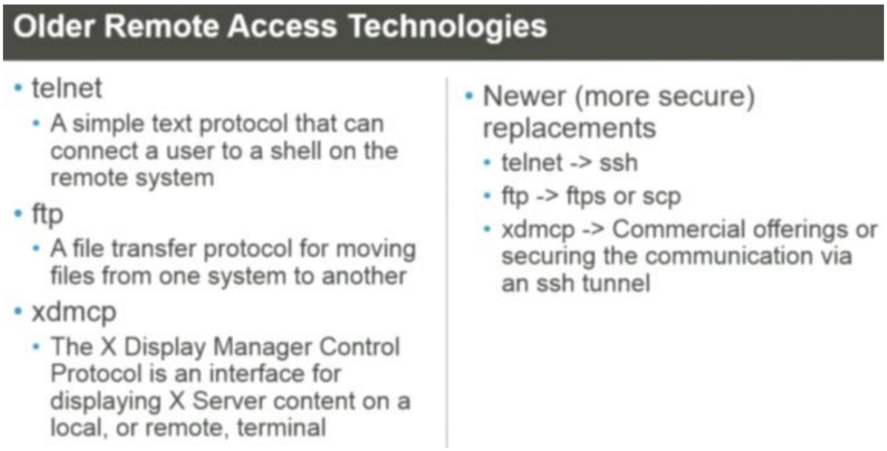
```
>service ntpd stop
>ntpdate pool.ntp.org(setup ntp )
```

COURSE 1 Part 4

Wednesday, November 15, 2023 3:51 PM



- Insecure
  - telnet
  - ftp
  - vnc
  - xdmcp
- Secure
  - ssh
  - scp
  - ftps
  - Tunneled vnc
  - Tunneled xdmcp



Older Remote Access Technologies

- telnet
  - A simple text protocol that can connect a user to a shell on the remote system
- ftp
  - A file transfer protocol for moving files from one system to another
- xdmcp
  - The X Display Manager Control Protocol is an interface for displaying X Server content on a local, or remote, terminal

- Newer (more secure) replacements
  - telnet -> ssh
  - ftp -> ftps or scp
  - xdmcp -> Commercial offerings or securing the communication via an ssh tunnel

VNC server: (graphical)

```
==========
>yum list installed | grep vnc-server
>vi /lib/systemd/system/vncserver@.service

>cp /lib/systemd/system/vncserver@.service /etc/systemd/system/vncserver@:1.service
>vi ncserver@:1.service
(ExecStart=/sbin/runuser -l cblackwell -c "/usr/bin/vncserver %i"
PIDFile=/home/username/.vnc/%H%i.pid)

>firewall-cmd --permanent --zone-pubic --add-port=5901/tcp
>firewall-cmd --reload

>systemctl start vncserver@:1.service
>systemctl daemon-reload


>scp filenames root@hostname:/Usefiles

>scp -Crl(compression recursive limit) 200(kbps) root@hostname:/oldername /Sample(source)
```

SSH tunnel:

```
--------------
>ssh -L(local) 8000:rhell.silverslides.local:80 root@hostname
(this will connect to silverslides.local:L80 when we go to localhost:8000)

>ssh -L 5901:rhell.silverslides.local:5901 root@rhell
```

SSH:

```
===
>ssh-keygen -t rsa

>ssh-copy-id(copy public key to ssh server) root@rhel1
```

>ssh-agent
>ssh-add (asks for the private key path)
(will add it to memory)
>ssh-add -l(lists)

After making the changes to make private key auth default go to /etc/ssh/sshd_config (and change passwordauthentication no) and sshd restart

NFS:
===

- Same version should be used for both clients and servers
- Hosts put shares in the /etc/exports file
- Clients mount host shares with the mount ccommand

NFSV1:
- NEVER RELEASE AND USED FIR INTERNAL TESTING

NFSv2:
- Designed for UDP and it was stateless

NFSv3:
- First support for files > 2GB
- Support for asynchronous writes and additional file attributes

NHSv4:
- Introduces security and become a stateful protocol

Portmap,nfs daemons might be required to be running

In /etc/exports file we can give these permissions
>ro -> read only
>rw -> read write
>sync -> clients must send a commit before hosts write the change to permanent storage
>no_subtree_check -> NFS will not check permissions of directories above the current directory. This can increase the performance not the security
>no_root_squash -> allows root user to connect to the export

>yum list installed | grep nfs
>cat /etc/hosts.allow(will contain the services allowed and on which subnets)
(portmap: 192.168.1.0/24
Lockd: subnet
Rquotad: subnet
Mountd: subnet
Statd: sibnet
)>cat /etc/hosts.deny (can give the list of daemnons with ALL as value)

>service nfs start

>cat /etc/exports
/UserFiles *(rw,root_squash)
*means all hosts else give ip subnet
Go to another host
>showmount -e rhel1 (show mounts)

NFSv4:
- TCP/UDP 2049,111
- Can use kerberos for secure auth

    - Krb5
    - Krb5i
    - Krb5p
- sec=sys uses local linux UID,GID
- sec=krb5p for kerberos on client side
    - Mount sec=krb5p

```
>mount rhel1:/UserFiles /LocalDirectory
>mount | grep User (gives info about mount)

>unmount /LocalDirectory

>vi /etc/fstab (file for adding nfs mount point entries)
Rhel1:/UserFiles /LocalDirecotry nfs defaults defaults 0 0
>mount -a (will look for fstab and update)


>exportfs -ua(make mounts unavailable to client)
>exportfs -r(make again available)
```

PARALLEL NETWORK FILE SYSTEM(Pnfs):

- Allows direct client access to a filesystem
- Clients have direct access to data through multiple servers concurrently
- Alsoused in storage appliances
- Useful in high performance data centers

```
>mount -o v4.1 server:/export /localmountdir

>mount -t cifs(samba server) //rhel1/UserFiles /Localdir
```

In fstab we need to give \\rhel1\serFiles /data cifs username=root,password=password 0 0

```
>smbclient \\\\rhel1\\UserFiles (will connect to via cli itself)
```

Configure ldap auth:
=================

```
>authconfig --enablesssd --enablesssdauth --enablelocauthorize --enableldap --enableldapauth --ldapserver=ldap://ldap.example.com:389 --disableldaptls --ldapbasedn=dc=example,dc=com --enablerfc2307bis --enablemkhomedir --enablecachecreds --update
(this command will update the sssd.conf file)
>service sssd start
>getent passwd mbishop(username)


>cat /etc/auto.master
/localdir /etc/auto.server --timeout=600
>cat /etc/auto.server
userfiles -fstype=cifs,credentials=/etc/auto.auth ://rhel1/Userfiles
>cat /etc/auto.auth
Username=root
Password=password

>service autofs start
```

DIRECTORY SERVICES:

-================

```
>yum list installed | grep authconfig or grep openldap or grep sssd
```

```
>system-config-authentication (starts the GUI for auth)
```

SSSD(RHEL SYSTEM SECURTY SERVICES DAEMON)

```
>/etc/sssd/sssd.conf
```

```
>/etc/nsswitch.conf ( here it is defined somethibg like for password looks for files for sss conf)
```

```
>service sssd status
```

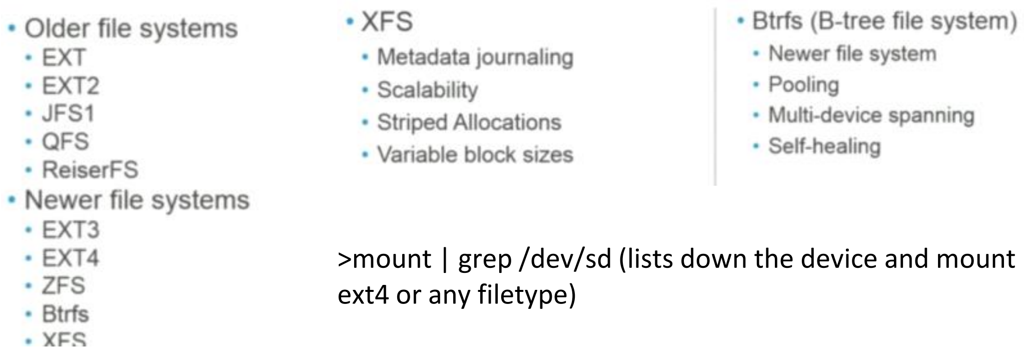
```
>ldapsearch -h 192.168.1.241 -x uid=username -b dc=example,dc=com
(return search result from directory)
>getent passwd username
```

```
>/etc/krb5.conf (THIS IS USED FOR ACTIVE DIRECTORY AUTH OR CONF)
>/etc/samba/smb.conf(for cms shared can iuse this for ad auth)
```

```
>kinit administrator@SILVERSIDES.LOCAL
>klist -k(list the hosts)
>net ads join -U administrator@SILVERSIDES.LOCAL
>authconfig --update --enablesssd --enablesssdauth
>service sssd restart
```

```
>ldapsearch -H ldap://srv2012-1.silversides.local:3260 -Y GSSAPI -N -b "dc=silversides,dc-local"
"{&{objectClass=user}{sAMAccountNae=jdoes}}"
```

==



- Older file systems
  - EXT
  - EXT2
  - JFS1
  - QFS
  - ReiserFS
- Newer file systems
  - EXT3
  - EXT4
  - ZFS
  - Btrfs
  - XFS

- XFS
  - Metadata journaling
  - Scalability
  - Striped Allocations
  - Variable block sizes

- Btrfs (B-tree file system)
  - Newer file system
  - Pooling
  - Multi-device spanning
  - Self-healing

```
>mount | grep /dev/sd (lists down the device and mount points and xfs or
ext4 or any filetype)
```

```
>mount /dev/sdb1 /shared
>unmount /dev/sdb1
```

```
>e2label /dev/sdb1 (can also specify our own label after the /dev/sdb1)
(op: WebData)
>mount -L(label) WebData /tmpmount
```

```
>vi /etc/fstab
LABEL=WebData /tmpmount ext4 defaults 0 0
>mount -a
(automount on boot)
```

```
>cryptsetup -v luksFormat /dev/sdb1 (to encrypt the disk)
```

```
(asks to overwrite  the disk and asks for passphrase)
>cryptsetup luksOpen /dev/sdb1 filevault
>cryptsetup -v status filevault

>mkfs.ext4 /dev/mapper/filevault (like formatting)
>mount /dev/mapper/filevault /Data


>vi /etc/crypttab (for encrypted disks)
(filevault /dev/sdb1 none(can give password if needed) luks)
>vi /etc/fstab
(/dev/mapper/filevault /Data ext4 default 0 0)
>mount -a
```

TO AUTOMATICALLY MOUNT OR UNMOUNT WHENEVER DISK IS PLUGGED IN

```
>cat /etc/auto.master
(/directory /etc/auto.ext-usb --timeout=10)
>cat /etc/auto.ext-usb
(ext_usb1 -fstype=auto :/dev/sdb1)
>ls /directory/ext_usb1
>mount | grep /dev/sdc1 (this will be mounted on /directory/ext_usb1)
```

PARTITION:
=========

```
>fdisk /dev/sdb
(prompts for commands)
P - for available partitions
q- quit
n- new partition
>e2fsck -f /dev/sdb1 (checks if it is ok)
>unmount /mnt(mount of sdb1)
>resize2fs /dev/sdb1 5G

>fdisk /dev/sdb
Command: d(dleete partition)
Commands: w (writed the change)
Command: n(new)

>mkfs -t ext4 /dev/sdb1

>fdisk -l /dev/sdb (list the partitions)
```

(GPT(grid partition table) disks support 128 disk partitions whereas MBR is 4)

```
>vi /etc/iscsi/iscsid.conf ( for iscsi devices)

>iscsiadm --mode discovery --type sendtargets --portal 192.168.1.240
o/p: 192.168.1.240:3260,1 iqn.2015.06.com.example:target1
>iscsiadm --mode node --targetname iqn.2015.06.com.example:target1 --portal 192.168.1.240 --login
>iscsiadm -m session -o show
>lsblk --scsi

>blkid (lists down the devices and their UUID's and use it in fstab as UUID=UUIDSTRING)
```

fdisk is for MBR disks and gdisk is for GPT disks

```
>gdisk /dev/sdc
```

SWAP FILES:
=========

```
>dd if=/dev/zero of=/swapfile bs=1024 count-524280
>mkswap /swapfile
>chmod 0600 /swapfile
>swapon /swapfile
>vi /etc/fstab
(/swapfile swap swap defaults 0 0)
```

Compression for GRUB:(grant unified bootloadeer)

```
>vi /etc/default/grub
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root rhgb quiet
zswap.enabled=1"
>grub2-mkconfig -o /boot/grub2/grub.cfg

>cat /sys/kernel/debug/zswap/stored_pages (how many pages is compressed using zswap)
```

LOGICAL VOLUME MANAGEMENT:
===========================

```
>pvdisplay | more
>vgdisplay| more
>ls /dev/vg1 (has logical volumes as a seprate file or folder)
>lvdisplay vg1 | more

>lvcreate -L 5G -n lvname vg1 -v(verbose)
>lvs(statua of lvs)
>lvextend --size +4G /dev/vg1/lvname

>mkfs -t ext4 /dev/vg1/lvname
>mount /dev/vg1/lvname /mnt
>unmount /mnt
>lvremove /dev/vg1/lvname


>lvcreate -L 9G -T(thin provisioning) vg1/lvname

>lvcreate -V(virtual size) 9G -T vg1/lvname -n lv1
>lvcreate -V(virtual size) 9G -T vg1/lvname -n lv2

>lvdisplay vg1 | more


>pvcreate -v /dev/sdb1 /dev/sdc1
>vgcreate -s 5G vg1 /dev/sdb1
>vgextend vg1 /dev/sdc1 (adding another disk to vg)
```

If wanted to use ssd seperte then use

```
>lvcreate --type cache -L 50G -n lv1 vg1 /dev/sdb(device of ssd)
```

SNAPSHOTS:
==========

```
>lvcreate -L 500M -s(snapshot) -n archive_snap /dev/vg1/archive(origin lv where you need to snapshot)
```

Backups:

```
>tar -cvzf archive.tar.gz *
>dd if='/dev/vg1/archive_snap of=/Backup/archive.dd

>tar -xzvf archive.tar.gz
>dd of=/dev/vg1/archive if=archive.dd
```

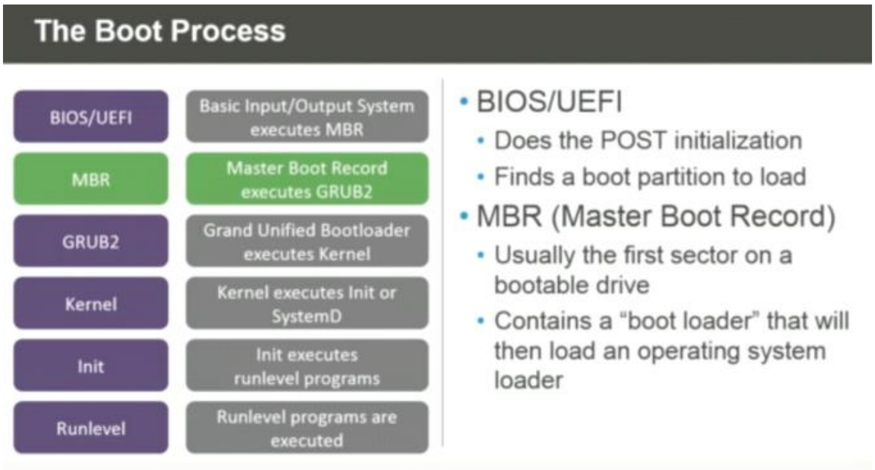
```
>lvmdiskscan (scans the disks and returns the summary)
>lvm
(interactive cmd)
>>pvs
>>vgs
>>exit
```

RAID PARTITIONS:(A method of mirroring or striping data on clusters of low-end disk drives; data is copied onto multiple drives for faster throughput, error correction, fault tolerance and improved mean time between failures.)
==============

```
>lsblk --scsi (block devices of type scsi)
>fdisk /dev/diskname
>>>all accept and in type press fd (for linux raid )

>mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sdb1 /dev/sdc1
>mdadm --detail /dev/md1

>cat /proc/mdstat (to see isks are added in raid array)

>mkfs -t ext4 /dev/md1
>mount /dev/md1 /mnt

>mdadm /dev/md1 --add /dev/sdd1(parititoned with fd type)
>mdadm --grow --raid-devices=3 /dev/md1
```

Here are some uses for /dev/zero:
Providing a character stream for initializing data storage
Creating files filled with zeroes
Creating a swap file
Formatting a drive and filling space with zeros to override old data
Wiping out a disk

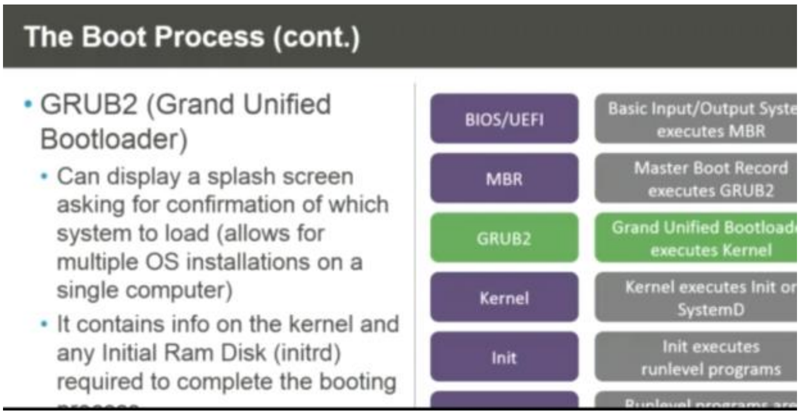
BOOT PROCESS:
=============



The Boot Process

BIOS/UEFI	Basic Input/Output System executes MBR
MBR	Master Boot Record executes GRUB2
GRUB2	Grand Unified Bootloader executes Kernel
Kernel	Kernel executes Init or SystemD
Init	Init executes runlevel programs
Runlevel	Runlevel programs are executed

- BIOS/UEFI
  - Does the POST initialization
  - Finds a boot partition to load
- MBR (Master Boot Record)
  - Usually the first sector on a bootable drive
  - Contains a "boot loader" that will then load an operating system loader



The Boot Process (cont.)

- GRUB2 (Grand Unified Bootloader)
  - Can display a splash screen asking for confirmation of which system to load (allows for multiple OS installations on a single computer)
  - It contains info on the kernel and any Initial Ram Disk (initrd) required to complete the booting

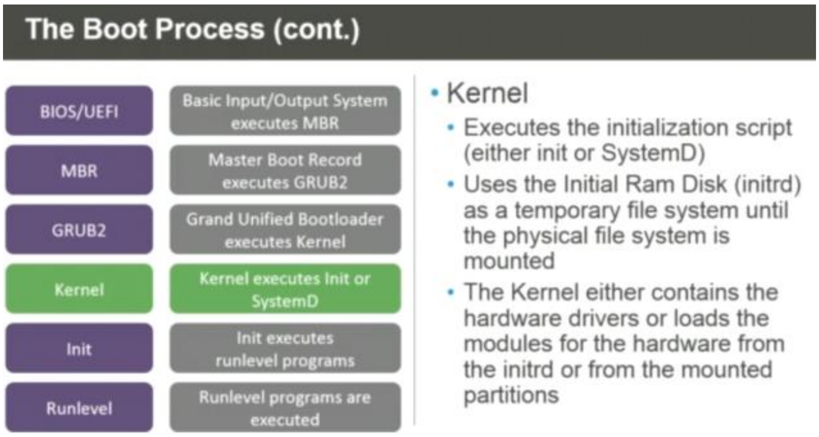
BIOS/UEFI – Basic Input/Output Syste executes MBR

MBR – Master Boot Record executes GRUB2

GRUB2 – Grand Unified Bootload executes Kernel

Kernel – Kernel executes Init or SystemD

Init – Init executes runlevel programs



The Boot Process (cont.)

BIOS/UEFI – Basic Input/Output System executes MBR

MBR – Master Boot Record executes GRUB2

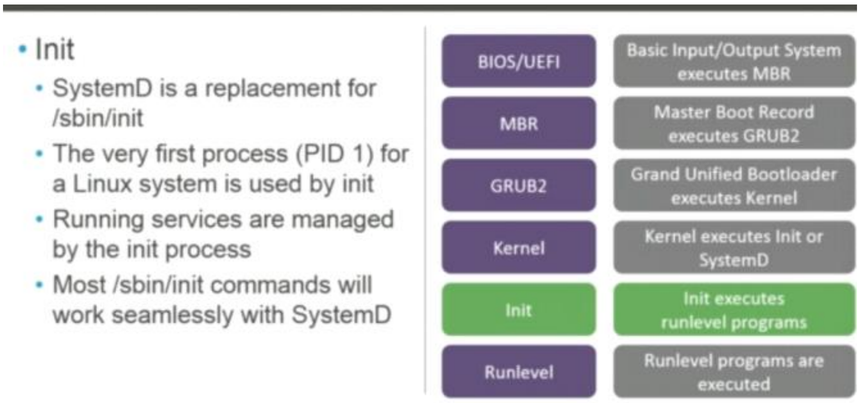
GRUB2 – Grand Unified Bootloader executes Kernel

Kernel – Kernel executes Init or SystemD

Init – Init executes runlevel programs

Runlevel – Runlevel programs are executed

- Kernel
  - Executes the initialization script (either init or SystemD)
  - Uses the Initial Ram Disk (initrd) as a temporary file system until the physical file system is mounted
  - The Kernel either contains the hardware drivers or loads the modules for the hardware from the initrd or from the mounted partitions



- Init
  - SystemD is a replacement for /sbin/init
  - The very first process (PID 1) for a Linux system is used by init
  - Running services are managed by the init process
  - Most /sbin/init commands will work seamlessly with SystemD

BIOS/UEFI – Basic Input/Output System executes MBR

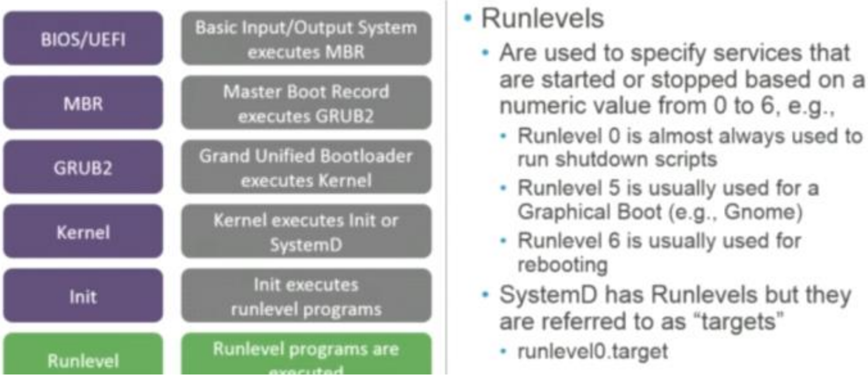
MBR – Master Boot Record executes GRUB2

GRUB2 – Grand Unified Bootloader executes Kernel

Kernel – Kernel executes Init or SystemD

Init – Init executes runlevel programs

Runlevel – Runlevel programs are executed



BIOS/UEFI – Basic Input/Output System executes MBR

MBR – Master Boot Record executes GRUB2

GRUB2 – Grand Unified Bootloader executes Kernel

Kernel – Kernel executes Init or SystemD

Init – Init executes runlevel programs

Runlevel – Runlevel programs are executed

- Runlevels
  - Are used to specify services that are started or stopped based on a numeric value from 0 to 6, e.g.,
    - Runlevel 0 is almost always used to run shutdown scripts
    - Runlevel 5 is usually used for a Graphical Boot (e.g., Gnome)
    - Runlevel 6 is usually used for rebooting
  - SystemD has Runlevels but they are referred to as "targets"
    - runlevel0.target

EG: if we type init 6 (it starts rebooting)

BOOTLOADER:

============

```
>cat /boot/grub2/grub.cfg (don't edit this file)
>vi /etc/default/grub
GRUB_TIMEOUT=20(shows us the boot menu for 20 s)
GRUB_DEFAULT=saved
```

```
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
```

In cmdline param we can change

```
rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root rhgb quiet ipv6.disable=1 zswap.enabled=1
```

```
>vi /etc/grub.d/40_custom (to customize the menu entry)
```

```
Menuentry "wndow 7"{
Insmod ntfs (inserting kernel module )
Set root='{hd0,3}' (fisk1 partition3 where it will be installed)
Search --no-floppy --fs-uuid --set UUIDSTRING
Chainloader +1 (for windows 7 boot loader)
}
```

```
>grub2-mkconfig -o /boot/grub2/grub.cfg (writes into this file with the new config)
>init 6 (reboots) (runlevel)
```

```
>vi /etc/inittab (init daemon was used before systemd)
```

```
>init 3(it makes a reboot and switches to only cli)
```

```
>systemctl list-units -t target (runlevels)
>systemctl get-default
>systemctl set-default multi-user.target
```

To run into single user mode when booting hover over the one which you need to booth on and click e and it goes to edit mode

Edit this line

```
Linux16 /vmlinux*……rhel-root rw init=/sysroot/bin/sh …
```

And ctrl+x

(maintenance mode)

```
# ls /
# chroot /sysroot
# passwd
```

New password and change password

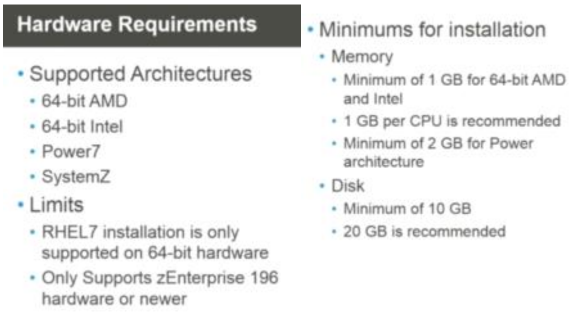
```
>systemctl is-enabled cups.service
>systemctl enable cups.service
```

(See what is socket in net)

Course 1 Part5

Thursday, November 16, 2023 5:07 PM

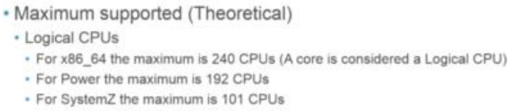
REDHAT LINUX INSTALLATION:



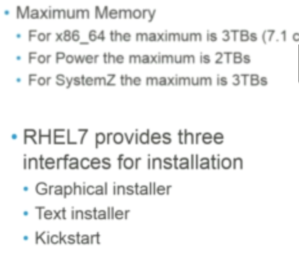
Hardware Requirements

- Supported Architectures
  - 64-bit AMD
  - 64-bit Intel
  - Power7
  - SystemZ
- Limits
  - RHEL7 installation is only supported on 64-bit hardware
  - Only Supports zEnterprise 196 hardware or newer

- Minimums for installation
  - Memory
    - Minimum of 1 GB for 64-bit AMD and Intel
    - 1 GB per CPU is recommended
    - Minimum of 2 GB for Power architecture
  - Disk
    - Minimum of 10 GB
    - 20 GB is recommended

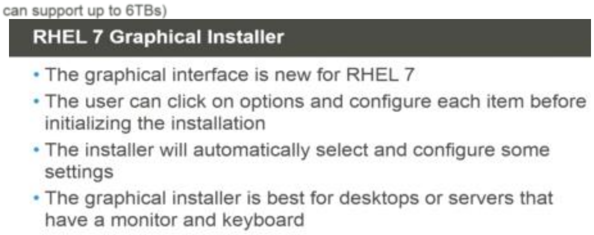


- Maximum supported (Theoretical)
  - Logical CPUs
    - For x86_64 the maximum is 240 CPUs (A core is considered a Logical CPU)
    - For Power the maximum is 192 CPUs
    - For SystemZ the maximum is 101 CPUs



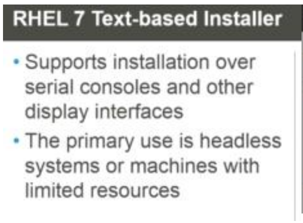
- Maximum Memory
  - For x86_64 the maximum is 3TBs (7.1 can support up to 6TBs)
  - For Power the maximum is 2TBs
  - For SystemZ the maximum is 3TBs

- RHEL7 provides three interfaces for installation
  - Graphical installer
  - Text installer
  - Kickstart



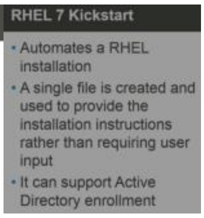
RHEL 7 Graphical Installer

- The graphical interface is new for RHEL 7
- The user can click on options and configure each item before initializing the installation
- The installer will automatically select and configure some settings
- The graphical installer is best for desktops or servers that have a monitor and keyboard



RHEL 7 Text-based Installer

- Supports installation over serial consoles and other display interfaces
- The primary use is headless systems or machines with limited resources



RHEL 7 Kickstart

- Automates a RHEL installation
- A single file is created and used to provide the installation instructions rather than requiring user input
- It can support Active Directory enrollment

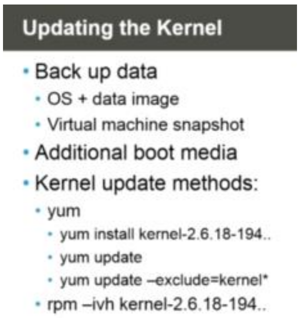
Press TAB in the menu option (test this media dn install*) and add a text word at the end
It will provide text based installation

```
>mount|  grep efi
(for installing on UEFI systems)
>ls /boot/efi/EFI/redhat/
(grub.cfg has the configs)
```

```
(List of kernel parameters along with values)
>sysctl -a

>sysctl -w net.ipv4.ip_forward=1 (so that this host can act as router)
>vi /etc/sysctl.conf (conf for sysctl)
>sysctl -p(for saving the changes made in the above file unless after reboot it works)

>ulimit -a(all restrictions including max open files etc.)
>ulimit -n(max open files )
>ulimit -u(max user processes)

>vi /etc/security/limits.conf (to edit the limits)
@username hard/soft nproc 400(max user processes)
@username hard/soft nofile 100(max no of open files)
```

Getting current version information:

```
>uname --r
>cat /etc/redhat-release
```



Updating the Kernel

- Back up data
  - OS + data image
  - Virtual machine snapshot
- Additional boot media
- Kernel update methods:
  - yum
    - yum install kernel-2.6.18-194..
    - yum update
    - yum update –exclude=kernel*
  - rpm –ivh kernel-2.6.18-194..

```
>/boot/grub/grub.cfg (should reference the new kernel version)
>need to configure GRUB to us ethe new kernel menu as the default
>reboot and verify os functions preperly
>verify correct hardware drivers has been loaded
        >lsmod
        >modinfo kernelmodule

>lsmod | grep e1000 (network card drive)
>rmmod e1000
>ifconfig(will not have interfaces)
>modprobe e1000 (will add this one)


>modinfo e1000
```

When there is an failure when booting we can do something like blacklist a module from loading when bot menu comes up and click e and on linux16 line add modprobe.blacklist=modulename after rhel-root

Here we can also make ro to rw and login as root and check the /etc/default/grub file is there is any mistake in command

We can use troubleshooting when boot menu and run rescue mode and it will load the system in a mount point and doing a chroot command on it
We can also reinstall grub by

```
>grub2-install /dev/sda
```

And after that agin going to boot menu and troubleshoot > bot from lcocal drive and execute as command by pressing c and it goes to cmd and give set root=(hd0,1) (set root as the hard drive number 0 and partition 1 and reboot command

```
>fschk /dev/sda
>umount /dev/sda1
>xfs_repair /dev/sda1
```

To blacklist while in kernel

```
>echo "blacklist modulename" >> /etc/modprobe.d/blacklist.conf


Menuentry "BSD LINUX"{
        Set root=(hd1,4)  (load from disk 2 and partition 4)
        Chainloader +1
}
>grub2-mkconfig -o /boot/grub2/grub.cfg




>ls /proc
```

VIRTUALIZATION:
=============

```
>yum install qemu-kvm qemu-img
>yum install "libvirt"

>YUM LIST INSTALLED | GREP "KVM" "VIRT"


>virsh help | more (managemenrt user interface)

>vi /etc/libvirt/gemu.conf
>service libvirtd status

>virsh list(list virtual machines)
>virsh dominfo vmname
>virsh snapshot-create vmname
>virsh console vmname
Ctrl+] for exiting
```

```
>man virt-install
>virt-install --ram 2040 --name rhel7.0.2 --disk path=/vm_disks/webserv2.img,size=0 --network network:default --cdrom /software/rhel-server-7.1-x86_64-dvd.iso
```

```
>virsh start vmname --console --force-boot
>virsh setmem vmname --size 1G
>virsh edit vmname(open xml file for that)
```

```
>localectl status
>localectl list-keymaps | grep french
>localectl set-x11-keymap cm-french (keyboard layout is set to french)
>localectl set-keymap cm-french (cli )
```

```
>virsh list --all
```

```
>virsh edit vmname
(go to devices section and add panic section
<panic>
        <address type='isa' iobase='0x505'/>
</panic>
)
```

A panic notifier is a way to pass an ID to callbacks to determine the type of event that happened.

Protecting guest os:
===============

```
>getenforce(to see if selinux is enforced)
>cat /etc/sysconfig/selinux
```

```
>virsh dumpxml vmname | grep label
```

```
>ls -laZ(gives labeling information)
>cd /vm_disks;ls -laZ
```

```
>virsh net-list --all(virtual networks)
>virsh net-dumpxml default(networkname)
```

```
>brctl show(bridge networks)
```

(bridge to virtual machine it includes different interfaces can be added to this brdge)

```
>virsh dumpxml vmname | grep network
```

```
>virsh nodedev-list --cap vports (nport virtualization isa added or not)
```

Nportid virtualization(multiple virtual machine share a single physical fibre channel host bus adapter(physical card in a host that would allow acccess to storage area netowkr storage)

```
>ls /var/lib/libvirt/images
```

```
>virsh domdisplay vmname(returns the vm host)
>vncviewer localhost:1
```

```
>virsh dumpxml vmname | grep graphic
>ssh vmip
```

```
>virsh start vmname
>virsh shitdown vmname
>virsh destroy vmname(like a force shutpff)
```

```
>virsh autostart vmname(auto boots when host machine starts)
>virsh autostart --disable vmname
```

```
>virsh
Virsh# domstate vmname
Virsh# vcpuinfo vmname
Virsh$ domiflist vmname
Virsh# dominfo vmname
```

```
>virsh screenshot vmname (saves the screenhos tof the current state of the vm)
```

```
>virt-top (like top command in linux)
```

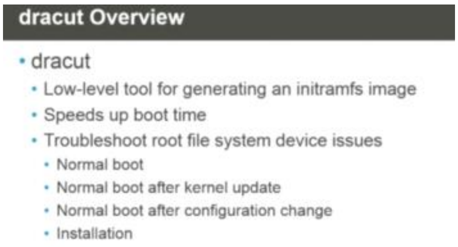
```
>virsh cpu-stats vmname
```

```
>virt-install --ram 2040 --name rhel7.0.2 --disk path=/vm_disks/webserv2.img,size=0 --location /software/rhel-server-7.1-x86_64-dvd.iso --nographics --extra-args="ks=http://192.168.1.201/ks.txt ip=dhcp console=tty$0" --os-variant rhel7
```

(using kickstart script generated from rehat website

In root home directory a file named anaconds-ks.cfg is placed if a manual installation has been made

>ksvalidator ksfiledownloadedfromwebsote



dracut Overview

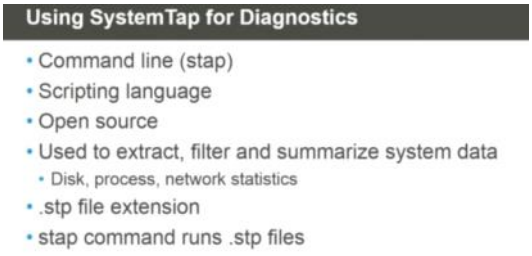
- dracut
  - Low-level tool for generating an initramfs image
  - Speeds up boot time
  - Troubleshoot root file system device issues
    - Normal boot
    - Normal boot after kernel update
    - Normal boot after configuration change
    - Installation

>yum install *dracut*
>modify grub.cfg and add rdshell as kernel parameter

>dmesg | grep dracut

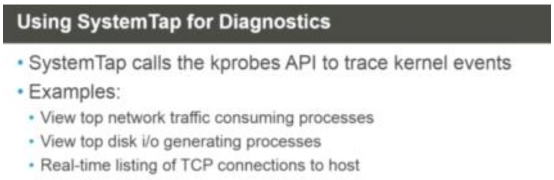
>ls -l /boot/init* (lists all initramfs files/imgs)

>dracut (look current kernel and generate new initramfs image)
An initramfs (initial ram file system) is used to prepare Linux systems during boot before the init process starts. The initramfs usually takes care of mounting important file systems (by loading the proper kernel modules and drivers) such as /usr or /var, preparing the /dev file structure, etc



Using SystemTap for Diagnostics

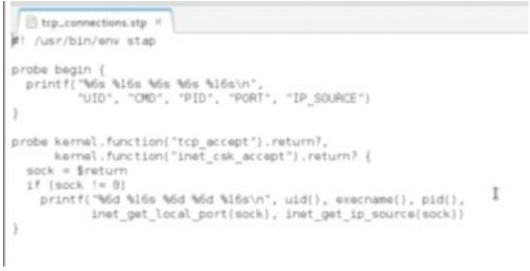
- Command line (stap)
- Scripting language
- Open source
- Used to extract, filter and summarize system data
  - Disk, process, network statistics
- .stp file extension
- stap command runs .stp files



Using SystemTap for Diagnostics

- SystemTap calls the kprobes API to trace kernel events
- Examples:
  - View top network traffic consuming processes
  - View top disk i/o generating processes
  - Real-time listing of TCP connections to host

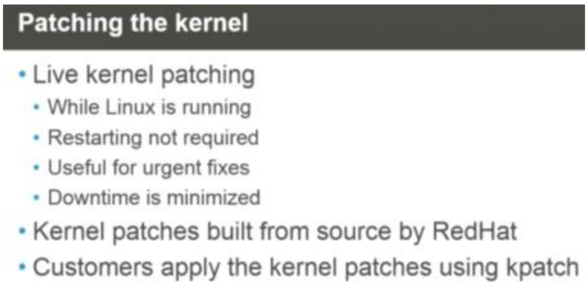
>stap tcp_connections.stp



```
#! /usr/bin/env stap

probe begin {
  printf("%6s %16s %6s %6s %16s\n",
         "UID", "CMD", "PID", "PORT", "IP_SOURCE")
}

probe kernel.function("tcp_accept").return?,
      kernel.function("inet_csk_accept").return? {
  sock = $return
  if (sock != 0)
    printf("%6d %16s %6d %6d %16s\n", uid(), execname(), pid(),
           inet_get_local_port(sock), inet_get_ip_source(sock))
}
```

>systemctl is-active abrtd.service
>ls /var/tmp/abrt(aitomatic bug report tool)

>abrt-cli list



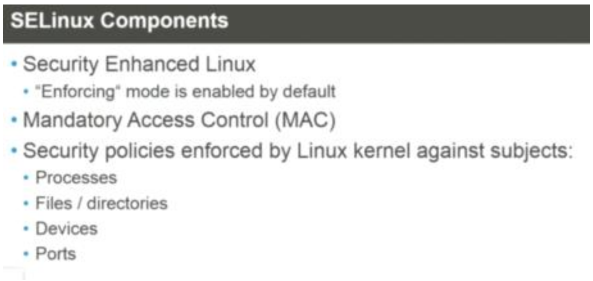
Patching the kernel

- Live kernel patching
  - While Linux is running
  - Restarting not required
  - Useful for urgent fixes
  - Downtime is minimized
- Kernel patches built from source by RedHat
- Customers apply the kernel patches using kpatch

>yum list installed| grep *kpatch*

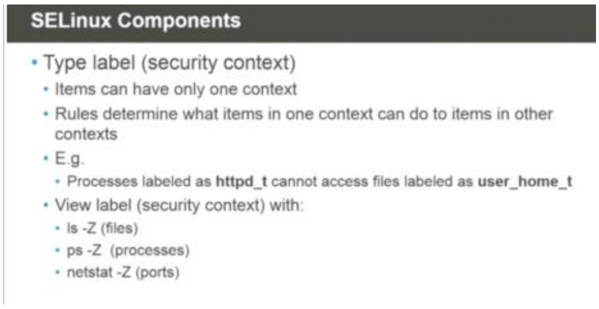
```
>kpatch install /unistall
>kpatch list
>kpatch load/unload
```

SELINUX:
=======



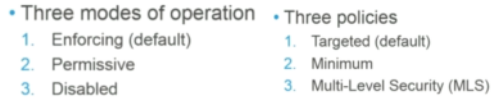
SELinux Components

- Security Enhanced Linux
  - "Enforcing" mode is enabled by default
- Mandatory Access Control (MAC)
- Security policies enforced by Linux kernel against subjects:
  - Processes
  - Files / directories
  - Devices
  - Ports

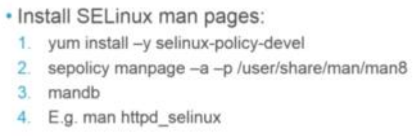


SELinux Components

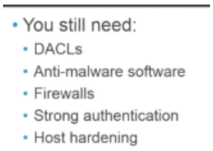
- Type label (security context)
  - Items can have only one context
  - Rules determine what items in one context can do to items in other contexts
  - E.g.
    - Processes labeled as **httpd_t** cannot access files labeled as **user_home_t**
  - View label (security context) with:
    - ls -Z (files)
    - ps -Z (processes)
    - netstat -Z (ports)



- Three modes of operation
  1. Enforcing (default)
  2. Permissive
  3. Disabled
- Three policies
  1. Targeted (default)
  2. Minimum
  3. Multi-Level Security (MLS)

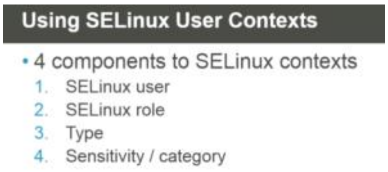


- Install SELinux man pages:
  1. yum install –y selinux-policy-devel
  2. sepolicy manpage –a –p /user/share/man/man8
  3. mandb
  4. E.g. man httpd_selinux



- You still need:
  - DACLs
  - Anti-malware software
  - Firewalls
  - Strong authentication
  - Host hardening

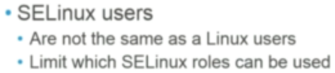
```
>setenforce 0(permissivemode)
>getenforce (gived the current mode)
>vi /etc/sysconfig/selinux (to be persistent across reboots)

>sestatus -v(view info on selinux)

>ps -axZ | grep http
System_u:system_r:httpd_t:s0 .. .. ..
>semanage permissive -a httpd_t

>semodule -l | grep permissive
>semanage permissive -d httpd_t

>getsebool -a| more
>setsebool -P(persistent) ftpd_anon_write on

>tail /etc/selinux/targeted/modules/active/booleans.local
```



Using SELinux User Contexts

- 4 components to SELinux contexts
  1. SELinux user
  2. SELinux role
  3. Type
  4. Sensitivity / category

Unconfined:object_r:default_t:s0
Selinuxuser:selinuxrole:type:sensitivity

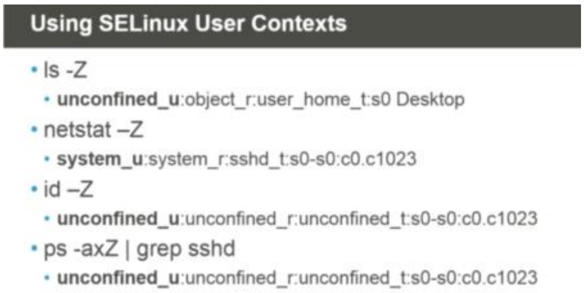


- SELinux users
  - Are not the same as a Linux users
  - Limit which SELinux roles can be used

```
>semanage login -l (login name corresponding to the selinux user mapped)

>seinfo -adomain -u (lists the selinux users available -r means roles)
>semanage user -l (selinux users corrsponding with their labels roles)
```

```
>semanage login -a(add) -s staff_u username
>semanage login -l(username and corresponding selinux user)
```

One user can be linked to one selinux iuser and selinux user can be linked to one or moreroles



Using SELinux User Contexts

- ls -Z
  - **unconfined_u**:object_r:user_home_t:s0 Desktop
- netstat –Z
  - **system_u**:system_r:sshd_t:s0-s0:c0.c1023
- id –Z
  - **unconfined_u**:unconfined_r:unconfined_t:s0-s0:c0.c1023
- ps -axZ | grep sshd
  - **unconfined_u**:unconfined_r:unconfined_t:s0-s0:c0.c1023

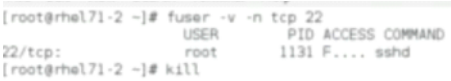
```
>chcon -t user_home_t(context name) filename
>man httpd_selinux
>restorecon filename(to restore back to previous one) (IF GIVEN -vR then it recursive to that folder)
>semanage port -l | grep httpd
>tail /var/log/audit/audit.log (here the permissive ennforcing logs)

>seinfo -rstaff_r -x| more ( types of/related to role staff_r)
>sestatus -v
```

Selinux type will be targeted(targeted processes are protected and minimum where we can include specifric processes and mls for even more granular

```
>semanage module -l

>getsebool -a

>semanage port -a -t ftp_data_port_t -p tcp 30 (add port 30 to tcp in ftp_data_port_t type)
>semanage fcontext -a -t httpd_sys_content_t "/website(/.*)?"

>semanage port -m -t unreserved_port_t -p tcp 80
```

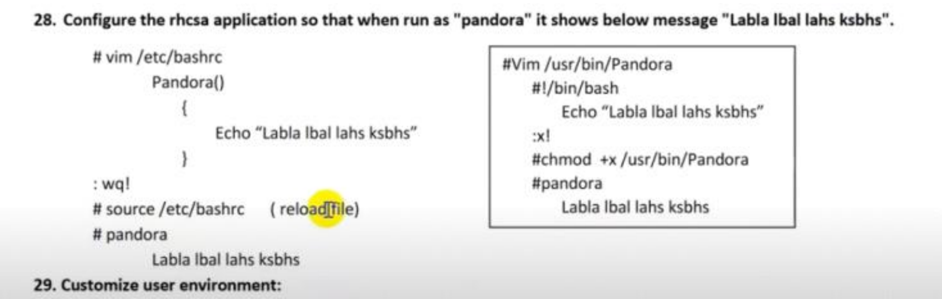
TROUBLESHOOTING:
=================

```
>cat /proc/meminfo
>free (free memory)
>free -m(in megabytes)
>top(stats for all processes mem,cpu)

>lsmod | grep -I ^e


>ifup eth0
>cat /etc/sysconfig/network-scripts/ifcfg-eth0

>lsof | more (list of open files)
>lsof -u username (files assigned to username)

>fuser -v -n tcp 22
```



```
[root@rhel71-2 ~]# fuser -v -n tcp 22
                     USER        PID ACCESS COMMAND
22/tcp:              root       1131 F.... sshd
[root@rhel71-2 ~]# kill
```

```
>xfs_repair -n /dev/sdva1
>xfs_repair -v(verbose) /dev/sdva1
>xfs_metadump /dev/sda1 /dumpfile
>xfs_admin -L(label) "labelname" /dev/sda1

>fsck -n /dev/sda1
>fsck -y(for giving yes for all questions)


>xfs_db /dev/sda1 -r(read only)
(interactive shell)
xfs_db>help (uuid|check|frag)
```



28. Configure the rhcsa application so that when run as "pandora" it shows below message "Labla lbal lahs ksbhs".

```
# vim /etc/bashrc
        Pandora()
            {
                Echo "Labla lbal lahs ksbhs"
            }
: wq!
# source /etc/bashrc      ( reload file)
# pandora
        Labla lbal lahs ksbhs
```

```
#Vim /usr/bin/Pandora
    #!/bin/bash
        Echo "Labla lbal lahs ksbhs"
    :x!
    #chmod  +x /usr/bin/Pandora
    #pandora
        Labla lbal lahs ksbhs
```

29. Customize user environment:

Rpm -qc autofs (lists all files associated with it)

Monday, November 27, 2023 12:48 PM

```
#!/bin/bash

while IFS=":" read user uid group ; do
echo "Creating user $user..."
useradd -b /mnt/autofs_home -G $group -u $uid -M $user
done < userlist.txt
```

IFS(internal field seprator in the file it will be groupname:gid\ngroupname:gid

https://infotechys.com/tag/rhcsa-exam/
https://github.com/aggressiveHiker/rhcsa9/blob/main/study_notes/general_notes.md

Windows recovery if boot order does not show the entry for it:

Install a bootable windows iso and when installing it instead of installing it just click rapair and startup repair and give in the command prompt if startup repair does not work these

```
>sfc /scannow
>bootrec /scanos
>bootrec /rebuildbcd
>bootrec /fixmbr
>bootrec /fixboot
```

And then restart and try out startup repair

And inside windows to remove the redhat or other os entry go to cmd and type
>mountvol X: /s (mounts the efi partition which may contain the redhat entry directory)
>X: > cd EFI> rmdir /s /q RedHat
>bcdedit /enum firmware (lists all entries)
>bcdedit /delete id (will be listed in previois command for redhat)

If the boot entry is not deleted after this also then we need to go to installation media of linux and troubleshotting,rescue redhat > 1 > efibootmgr > efibootmgr -b bootentryno -B > efibootmgr -w(write changes) > reboot