#  Accessing System and Understanding Essential Tools

## Objectives :-

- Accessing the command line and run simple commands using shell.

- Configuring SSH Key-based Authentication

- Understand and use essential tools

---

| Index | Topic Covered
|

| ----- | ---------- |

| **1** | <ul><li> Introduction to the bash shell.</li><li> Shell Basics</li><li>Types of Shells</li><li> Default Shell in Linux</li><li>Switching between Shells</li></ul>
|

| **2** | <ul><li>Logging in to Local Computer</li><li>Virtual Consoles</li><li>Logging in over network</li></ul> |

| **3** | <ul><li>SSH Key-based Authentication</li><li>Generating SSH Keys using ssh keygen command</li><li>Access remote system using SSH</li></ul>
|

| **4** | <ul><li>Use grep and regular expressions to analyze text</li> <li> Log in and switch user in multiuser target</li><li>Create hard and soft links</li> <li>use system documetation including man, info,help</li></ul> |

### **Q. copy /etc/fstab to /var/tmp. user natasha has read and write permission, user sarah has no any permission to /var/tmp**

### Answer :-

```

[root@localhost ~]# cp /etc/fstab /var/tmp


[root@localhost ~]# getfacl /var/tmp (check permissions)
```

Question 1 Correct Mark 1.00 out of 1.00 Flag question

What number will the echo $? command return if the previous command's exit code was 0? Answer: 0 õige

Question 2 Correct Mark 1.00 out of 1.00 Flag question

For what purpose should any IT professional learn how to administer Linux?

(choose 2)

a. Linux is portable and can be used for embedded and low physical resource systems. õige

b. Knowing Linux is mandatory when working for an IT company. vale

c. Most Linux systems are open source and therefore customisable. õige

d. Linux is closed source (more secure) and cannot be used as an OS which is modified according to the developer's desire. vale

e. RHEL is free unconditionally.

Question 3 Partially correct Mark 0.50 out of 1.00 Flag question

How to rename directory "dir1" to "dir2" in the current directory? (choose 4)

a. rename 1 2 dir1 õige

b. mv dir1/ dir2 õige

c. mv dir1/ dir2/ õige

d. mv dir1 dir2 õige

e. move "dir1" "dir2"

f. mkdir dir2; mv dir1 dir2

g. mv ./dir1 ~/dir2 vale

mv - move (rename) files rename - rename files

Question 4 Correct Mark 1.00 out of 1.00 Remove flag

echo 1 >> file1

echo 2 > file1

"file1" does no exist and these 2 commands will be executed, what will happen?

a. The "file1" will be created with the content "1" and "2" will be appended on the next line resulting in " 1 2 ". vale

b. The "file1" will be created with the content "1" and "2" will be appended on the same line resulting in "12". vale

c. The "file1" will be created with the content "1" and "2" will be the final content of "file1" after the commands. õige

> will create and/or overwrite.

>> will create and/or append.

Question 5 Correct Mark 1.00 out of 1.00 Flag question

Which CLI utility can not be used to view networking configuration in RHEL.

a. nmcli ...

b. ifconfig ...

c. iwconfig ...

d. ip ...

e. ipconfig ... õige

Question 6 Correct Mark 1.00 out of 1.00 Flag question

For what purpose should any IT professional learn how to administer Linux? (choose 2)

a. Knowing Linux is mandatory when working for an IT company. vale

b. Other OS users also need to interoperate with Linux (servers). õige

c. Linux is the cheapest OS. vale

d. All are correct. vale

e. Cloud environments are often based on Linux distributions. õige

Question 7 Correct Mark 1.00 out of 1.00 Flag question

Various system logs cannot be viewed with.

a. tail -f /var/log/secure õige

b. journalctl õige

c. less /var/log/syslog õige

d. cat /var/log/messages vale

e. dmesg vale

Question 8 Correct Mark 1.00 out of 1.00 Flag question

What does the command sudo id -g return?

0 - õige

Question 9 Partially correct Mark 0.50 out of 1.00 Flag question

When any user runs a command with sudo it is executed as root.

User passwords are typically stored in /etc/shadow vale

Hashed user passwords are stored in /etc/shadow õige

Question 10 Partially correct Mark 0.33 out of 1.00 Flag question

Which of these does not configure the httpd service to start automatically when the OS is started?

a. systemctl enable httpd vale

b. systemctl enable http õige

c. service enable httpd õige

d. service httpd enable vale

e. systemd enable httpd õige

Question 11 Incorrect Mark 0.00 out of 1.00 Flag question

ssh connection to user@host can be done with ... (choose 3)

a. [user@host ~]$ ssh -i user host õige

b. ssh -l user host vale õige

c. ssh host -l $USER õige

d. ssh -user @host vale

e. ssh host@user vale

man ssh

Question 12 Partially correct Mark 0.33 out of 1.00 Flag question

Match the correct answers.

"a+rwx,go-x" "rwxrw-rw-"

644 "rw-r--r--"

"rw-------" 600 õige

[root@localhost ~]# setfacl -m u:natasha:rw- /var/tmp

[root@localhost ~]# setfacl -m u:sarah:--- /var/tmp

[root@localhost ~]# getfacl /var/tmp (again check permissions)

```

### **Q. Change the current active tuning profile to powersave**

### Answer :-

```

[root@localhost ~]# tuned-adm list (command gives list of available profile)

[root@localhost ~]# tuned-adm profile powersave

to confirm that powersave profile is active, execute following command

[root@localhost ~]# tuned-adm active
```

# <center> Controlling_access_to_files</cneter>

## Objectives : -

- Managing file system permissions

- Managing default permissions

| Index | Topic Covered |

| ----- | ---------- |

|**1**|<ul><li>Changing file and directory permissions using **chmod** command</li><li>syntax is --> chmod whowhatwhich file or directory</li><li>who is g,u,o,a (group,user,other,all)</li><li>what is +,-,= (add,remove,exactly)</li><li>which is r,w,x (read, write, execte)</li><li>e.g. chmod g+x file1</li></ul>|

|**2**|<ul><li>Cotrolling default permissions</li><li>Effect of special permissions</li><li>Default permissions (drwxrwxrwx)-(0777)</li></ul>

# <center> Controlling System Services</center>

## Objectives :-
```

- Starting and Stoping services

- Restarting services

- Enabling or disabling services

### Starting and Stoping services

Start services using systemctl command.

**systemctl start SERVICENAME** command used to start specific service.

```

[root@localhost ~]# systemctl start httpd.service

[root@localhost ~]# systemctl start httpd (no need to type .service same command as above)

```

Stop services using systemctl command.

**systemctl stop SERVICENAME** command used to stop specific service

```

[root@localhost ~]# systemctl stop httpd.service

```

### Restarting services

Restarting services using systemctl command

**systemctl restart SERVICENAME** command used to restart specific service.

```
```

[root@localhost ~]# systemctl restart httpd.service

```

### Enabling or disabling services

enabling or disabling services to start or stop at boot

**systemctl enable SERVICENAME** command used to enable services

```

[root@localhost ~]# systemctl enable httpd

```

**systemctl disable SERVICENAME** command used to disable services

```

[root@localhost ~]# systemctl disable httpd.service

```

### **Q. Create a 2GB partition using /dev/sdb make it as ext4 file system and mounted automatically under /mnt/data at boot-start**

### Answer : -

```

[root@localhost ~]# fdisk /dev/sdb

command (m for help): press 'n' here

select (default p): Enter

partition number (1-4, default between 1-4): Enter

first sector(default): Enter

last sector, sector or +size (K,M,G,T,P)(default): +2G

command (m for help): press p for partition info

command (m for help): press w to save changes

[root@localhost ~]# udevadm settle (to setup driver)

[root@localhost ~]# mkfs.ext4 /dev/sdb1 (here sdb1 is partition in /dev/sdb)

[root@localhost ~]# mkdir /mnt/data

[root@localhost ~]# mount /dev/sdb1 /mnt/data

To mount a partition automatically under /mnt/data make the entry of partition in /etc/fstab file. use following command:

[root@localhost ~]# vim /etc/fstab (add partition entry here)

### **Q. Add aditional swap partition of 2GB using /dev/sdd? and mount it permanently.**

### Answer:-

```

[root@localhost ~]# lsblk (check description of partitions)

[root@localhost ~]# fdisk /dev/sdd

command (m for help): press 'n' here

select (default p): Enter

partition number (1-4, default between 1-4): Enter

first sector(default): Enter

last sector, sector or +size (K,M,G,T,P)(default): +2G

command (m for help): press p for partition info
```

now we need this partition to be swap partition. hence we need to change the flag.

command (m for help): press 't' here

partition number (default 1): enter

change the partition regular to swap use hex code 82.

Hex code (type L to list all hex code): 82

changed type of partition 'Linux' to 'Linux swap'

command (m for help): press w to save changes

[root@localhost ~]# partprobe

[root@localhost ~]# mkswap /dev/sdd1

[root@localhost ~]# swapon /dev/sdd1

[root@localhost ~]# swapon -s (check swap summary)

To mount swap permanently make entry in fstab

[root@localhost ~]# vim /etc/fstab

```

### <center>**Add user Krish such that it's password not gonna expire.**

### Answer :-

```

[root@localhost ~]# useradd -h (get help to checkout options for password inactive period)

Graphical window appears

select -f to option to decide password period.

[root@localhost ~]# useradd -f -1 Krish

here password expiry value is -1 which means not expiry period.

```

### Q. **Create a group usergroup with 1788 groupid**.

### Answer :-

```

[root@localhost ~]# groupadd -g 1788 usergroup

[root@localhost ~]# cat /etc/group (checkout groups and id's)

```

### ***Q. Configure a cronjob that runs 14:10 minutes and executes, logger"Target EX200" at the user Krish.***

### Answer :-

```

[root@localhost ~]# crontab -eu Krish

10 14 * * *logger"Target EX200"

[root@localhost ~]# crontab -l (check cronjob)

```

### <cneter>**Q. Enable SELinux.**</center>

### Answer: -

```

[root@localhost ~]# getenforce (checks status of SELinux)

if it shows disabled or permissive go to /etc/selinux/config file and change configuration of selinux as,

SELINUX=enforcing

[root@localhost ~]# systemctl reboot

Again check status of SELinux after reboot.

```

# <center>File System</center>

## Objectives : -

- Linux file-system Hirarchy
- Managing files using command line tools
- Making links between files (hard and soft links)
```

| Index | Topic Covered
|

| ----- | ---------- |

| **1** | <ul><li>File-system Hirarchy</li><li>Types of file in linux</li><li>file system nevigation commands</li><li> / is the topmost directory</li><li>root directory with its sub directories</li></ul> |

| **2** | <ul><li>Command line file managment</li><li>file management commands</li><li>Creating Directories</li><li>Copying and moving files</li></ul>
|

| **3** | <ul><li>Creation of link files</li><li>Types of link files --> 1. Hard link 2. Soft link</li><li>Conclusion about hard link file</li><li>conclusion about soft link file</li><li>Limitations of Hard links</li></ul> |

### **Q. Find string empty in /usr/share/dict/words and put into /root/emptyword.**

### Answer :-

```
[root@localhost ~]# grep empty /usr/share/dict/words > /root/emptyword.
```

### **Q.Locate all files owned by user Eric and copy all those files under /root/Eric-files**

### Answer : -

```
[root@localhost ~]# mkdir /root/Eric-files
```

[root@localhost ~]# find / -user Eric (finds all files owned by Eric user)

/tmp/hello.txt

/tmp/hi.txt

[root@localhost ~]# cp /tmp/hello.txt /root/Eric-files (copy files owned by eric user)

[root@localhost ~]# cp /tmp/hi.txt /root/Eric-files

[root@localhost ~]# ls -la /root/Eric-files (check whether all files copied or not)

```

### **Q. Add three users: harry, natasha, tom. The requirements: The Additional group of the two users: harry, Natasha is the admin group. The user: tom's login shell should be non-interactive.**

### Answer :-

```

[root@localhost ~]# groupadd admin

[root@localhost ~]# useradd -G admin harry

[root@localhost ~]# useradd -G admin natasha

[root@localhost ~]# cat /etc/group (check group and group members)

[root@localhost ~]# useradd -s /sbin/nologin tom

[root@localhost ~]# cat /etc/passwd (check login shell)

```

# <center>**Configuring host name**</center>
```

## Objective :-

- Configuring host name

### configuring host name using hostnamectl command

The **hostname** command shows the host name of system

```
[root@localhost ~]# hostname
localhost.localdomain
```

**hostnamectl status** command displays details of host name

```
[root@localhost ~]# hostnamectl status
```

displays info.

```
**hostnamectl set-hostname NAME** command sets the hostname for system.
```

```
[root@localhost ~]# hostnamectl set-hostname host.example.com
```

To sync the hostname on terminal execute following command or open new terminal.

```
[root@localhost ~]# exec bash
```

check changed hostname here

```
[root@localhost ~]# cat /etc/hostname
```

```

# <center> Managing Local users and Groups </center>

## Objectives : -

- Users and Groups concepts

- Super user access

- Local user accounts

- Local group accounts

- Managing user password

| Index | Topic
Covered
|

| ----- | ---------- |

| **1** | <ul><li>Concept of user and group</li><li>Primary groups and supplimentory groups</li></ul>
|

| **2** | <ul><li>Concept of Superuser</li><li>Switching between users using su command e.g. su - user01</li><li>Running Commands with sudo (super user do) command.</li><li>root shell with sudo</li><li>sudo configuration in /etc/sudoers in file</li></ul> |

| **3** | <ul><li>Managing local users</li><li>Creating users from command line e.g. useradd user01</li><li>Modifying existing users from command line using usermod command</li><li>Deleting users using command e.g. userdel user1</li></ul> |

| **4** | <ul><li>Managing local groups</li><li>Creating groups from the command line. groupadd command create groups.</li><li>Modifying existing grouops from command line
```

using groupmod command</li><li>Deleting groups using command e.g. groupdel group1</li></ul> |

| **5** | <ul><li>Managing user password</li><li>Shadow passwords</li><li>Format of an Encrypted Password</li><li>Password Aging concept</li><li>Restricting Access</li><li>The no login Shell</li></ul> |

# **Managing Storage**

## Objective :-

- Partitions,file system, permanent mounting

- mounting, unmounting file systems

- managing swap partitions

### Partitions, file system, Presistent mounting

```

[root@localhost ~]# fdisk -l (check disk list)

Select disk. Use following command

[root@localhost ~]# fdisk /dev/sdd

command (m for help): press 'n' here

select (default p): Enter

partition number (1-4, default between 1-4): Enter

first sector(default): Enter

last sector, sector or +size (K,M,G,T,P)(default): +2G

command (m for help): press p for partition info

command (m for help): press w to save changes
```

here 2GB partition created.

[root@localhost ~]# udevadm settle (to setup driver)

[root@localhost ~]# mkfs.ext4 /dev/sdd1 (here sdd1 is partition in /dev/sdd)

[root@localhost ~]# mkdir /mnt/data

[root@localhost ~]# mount /dev/sdd1 /mnt/data

To mount a partition automatically under /mnt/data make the entry of partition in /etc/fstab file. use following command:

[root@localhost ~]# vim /etc/fstab (add partition entry here)

```

### Mounting, Unmounting file system

**mount** command used to manually mount a file system. Example of mounting by block device name

```

[root@localhost ~]# mount /dev/sdd1 /mnt/data

```

**umount** command used to manually unmount a file system. Example of unmounting by block device name

```

[root@localhost ~]# umount /dev/sdd1 /mnt/data

```

### Managing swap partition.

```

[root@localhost ~]# lsblk (check description of partitions)

[root@localhost ~]# fdisk /dev/sdd

command (m for help): press 'n' here

select (default p): Enter

partition number (1-4, default between 1-4): Enter

first sector(default): Enter

last sector, sector or +size (K,M,G,T,P)(default): +2G

command (m for help): press p for partition info

now we need this partition to be swap partition. hence we need to change the flag.

command (m for help): press 't' here

partition number (default 1): enter

change the partition regular to swap use hex code 82.

Hex code (type L to list all hex code): 82

changed type of partition 'Linux' to 'Linux swap'

command (m for help): press w to save changes

[root@localhost ~]# partprobe

[root@localhost ~]# mkswap /dev/sdd1

[root@localhost ~]# swapon /dev/sdd1

[root@localhost ~]# swapon -s (check swap summary)

To mount swap permanently make entry in fstab

[root@localhost ~]# vim /etc/fstab

```

### **Q. Change user krish user id from 1200 to 1284.**

### Answer :-

```

checkout uid of user krish

[root@localhost ~]# id krish

uid=1200(krish) gid=1201(krish) groups=1201(krish)

[root@localhost ~]# usermod -u 1284 krish

uid=1284(krish) gid=1201(krish) groups=1201(krish)

```

# <center>**Monitoring Services**</center>

## Objectives :-

- Describing Service Units
- Listing Service Units
- checking status of a service
- verifying status of a service

### Describing Service Units

The systemctl command used to manage units.

### Listing service units

```

[root@localhost ~]# systemctl list-units --type=service --all

only systemctl command list units that loaded and active.

[root@localhost ~]# systemctl

```

### checking/viewing status of a service

**systemctl status SERVICENAME** command shows status of service

```

[root@localhost ~]# systemctl status httpd.service

```

### Verifying status of a service

**systemctl is-active SERVICENAME** command shows states of a service

```

[root@localhost ~]# systemctl is-active httpd.service

active/inactive

```

**systemctl is-enable SERVICENAME** command shows status of service will be persistent after reboot

```

[root@localhost ~]# systemctl is-enable httpd.service

```

# <center> **Networking** </center>

## Objectives :-

- Getiing familiar with nmcli command to configure network settings.
```

- Modifying network configuration using /etc/sysconfig/network-scripts/ifcfg-name.

### Network manager concepts

commands and graphical tools contacts to NetworkManager and saves configuration file in /etc/sysconfig/network-scripts directiry.

### listing networking information.

**nmcli dev status** command shows the status of network devices.

```
[root@localhost ~]# nmcli dev status

DEVICE  TYPE     STATE       CONNECTION

enp0s3  ethernet connected     enp0s3

virbr0  bridge  connected      virbr0
```

nmcli con show shows list of all connections

```
[root@localhost ~]# nmcli con show
```

### Controllng network connections

The **nmcli con up NAME** command activates the connection name on the network interface.

```
[root@localhost ~]# nmcli con up enp0s3
```

The **nmcli con down NAME** command deactivates the connection name on the network interface.

```

[root@localhost ~]# nmcli con down enp0s3

```

### Deleting a network connections

The **nmcli con del NAME** commmand deletes the connection.

```

[root@localhost ~]# nmcli con del enp0s3

```

### Connection configuration files

changes made with nmcli con mod name saved to /etc/sysconfig/network-scripts/ifcfg-name.

### Modifying network configuration

It is easy to configure the network by directly editing the connection configuration files.

these files are named **/etc/sysconfig/network-scripts/ifcfg-name**.

note - after modifying network configuration file run following commands to make NetworkManager read the configuration changes.

```

[root@localhost ~]# nmcli con reload

[root@localhost ~]# nmcli con down enp0s3

[root@localhost ~]# nmcli con up enp0s3

```

### **Configure Host Name, IP address, Gateway and DNS**

### Host Name --> station.domain40.example.com

### IP address --> 172.24.40.40

### Gateway --> 172.24.40.1

### DNS -->172.24.40.1

### Answer :-

```
[root@localhost ~]# hostnamectl set-hostname station.domain40.example.com

[root@localhost ~]# vim /etc/sysconfig/network-scripts/ifcfg-enp0s3

- Things needs to be configured are

BOOTPROTO=static
ONBOOT=yes
IPADDR=172.24.40.40
GATEWAY=172.24.40.1
DNS1=172.24.40.1

[root@localhost ~]# cat /etc/resolv.conf (check DNS configuration)
```

### **Q. Synchronize time of your system with the server classroom.example.com**

### Answer :-

```

[root@localhost ~]# yum install -y chrony

[root@localhost ~]# systemctl start chronyd

[root@localhost ~]# systemctl enable chronyd

[root@localhost ~]# vim /etc/chrony.conf

in this configuration file comment active server and add given server

add --> server classroom.example.com iburst

save changes

[root@localhost ~]# systemctl restart chronyd

[root@localhost ~]# timedatectl (check whether ntp service active or not)
```

# <cneter> **Resetting the root password**

## Objective :-

- Resetting the root password from the boot loader.

### Resetting the root password from the boot loader.

To access that root shell, use following steps.

- step 1 : reboot system

- step 2 : Interrupt the boot loader countdown by pressing up down arrow key.
- step 3 : select the kernel entry to boot
- step 4 : press **e**
- step 5 : move the pointer at the end of kernel command line.
- step 6 : type rd.break enforcing=0
- step 7 : press crtl + x

Now switch_root prompt appears. run following commands to change root password.

```
switch_root:/# mount -o remount,rw /sysroot

switch_root:/# chroot /sysroot

switch_root:/# passwd root

Changing password for user root.
New password: toor
BAD PASSWORD: The password is shorter than 8 characters Retype new password: toor
passwd: all authentication tokens updated successfully

switch_root:/# touch /.autorelabel

switch_root:/# exit
```

```

### **Q. Interrupt the boot process and reset the root password change it to "root" to gain access to system.**

### Answer: -

To access that root shell, use following steps.

- step 1 : reboot system

- step 2 : Interrupt the boot loader countdown by pressing up down arrow key.

- step 3 : select the kernel entry to boot

- step 4 : press **e**

- step 5 : move the pointer at the end of kernel command line.

- step 6 : type rd.break enforcing=0

- step 7 : press crtl + x

Now switch_root prompt appears. run following commands to change root password.

```

switch_root:/# mount -o remount,rw /sysroot

switch_root:/# chroot /sysroot

switch_root:/# passwd root

Changing password for user root.

New password: toor

BAD PASSWORD: The password is shorter than 8 characters Retype new password: toor

passwd: all authentication tokens updated successfully

switch_root:/# touch /.autorelabel

switch_root:/# exit

```

# Resume

Most of the stuff here comes from the .txt with notes. There are some cases where I copied the information from the man pages or Internet.

Some sections like **`vim`** or **`kickstart`** aren't present, others were reduced (in comparison to the original note files), due printing reasons (paper and ink are expensive).

## **`ls` & `redirect symbols`**

### Why two different things share a table? Because I'm trying to save space

ls option|Description|Redirect Symbol|Description

-|-|-|-

l | Extended output|< \<filename>|uses file as stdin

ld | Directory output|>|stodout overwrites file

a | Shows all files, even hidden ones|>>|sdtout appends to file

Z | SELinux context|2> \<filename>|stderr to file

R|Recursive|2>1|stderr to stdout
```

 | |&> \<filename>|stdout and stderr to file name

## **`touch` command**

Create files if they don't exist, otherwise, modify the timestamp.

**`touch`** `foo` creates the file `foo`

## **`uname` command**

**`uname`**  `-rms` show the current kernel.

## **`ln` command**

The purpose of **`ln`** is create another name for a file. Reference the same contents of the file but with another name.

If you delete a file with a hard link, the content will be available on the hard link.

If you delete a file with a symbolic link, the symbolic link won't work.

**`ln`** `[source] [name of the link]`

**`ln`** `fileA fileB` creates a link where fileA is the original

**`ln`**  `-s fileA symfileB` creates a symbolic link

## **`grep` command**

Option|Function

-|-

-i|case insensitivity

-v|lines without matches

-r|recursive search

-A `[n]`|display X of lines after the match

-B `[n]`|display X of lines before the match

-e|multiple RegEx can be supplied as OR

-n|display line number

### **RegEx**

Symbol|Usage|Example|Applies for

-|-|-|-

\^|beginning of the line|^cat|category

\$|end of the line|$dog|chilidog

\.|wildcard single character|c.t|cat/cbt/cct

*|any amount of characters|c*t|cat/cbt/caaaaat

\.*|zero to infinitely characters|c.*t|ct/cat/coat/culvert

.\\{\\}|explicit multiplier|c.\\{2\\}|coat

\\< \\>|word boundary|\\<ipsum\\>|Lorem ipsum et

[ ]|options for a single character|c\[abc]t|cat/cbt/cct

## **`locate` & `find`**

**`locate`** `[search term]` search every file with the search term on it's name.

**`locate`**  `-i [search term]` case insensitive.

**`locate`**  `-n [n] [search term]` search and stops after `n` results.

**`updatedb`** update the locate database.

**`find`** `[directory to start] [search term]`

Option|Function

-|-

-user|search files that belong to that username

-uid|same as -user but with the UID

-group|search files that belong to that group

-gid|same as -group but with GID

-perm `[permissions]`|search for permissions based on the operator

 |764 only `-rwxrw-r--`

 |\-324 at least `--wx-w-r--`

 |/442 `u r--` OR `g r---` OR `o -w-`

-size `[n][k,M,G]`|search by size (round up to single units 995 KiB = 1MiB)

 |\+10M more than 10 MiB

 |\-1G less than 1 GiB

-mmin `[n]`|modified files since at least `[n]` minutes

-type| `r` regular file, `d` directory, `l` symlinks, `b` block device

-links|regular files with more names

**`find`**  `/home -user foo` find all the files that belong to `foo`

**`find`**  `/ -type l -links +3` find all the symbolic links with 3 or more names.

## **Users**

`/etc/passwd` contains the local user information.

`/etc/shadow` contains the user's passwords.

`/etc/group` contains the local group information.

`/etc/login.defs` contains the default parameters of accounts, such as password age.

**`authconfig`** `--passalgo [algorithm]`

**`useradd`** `[username]`

**`userdel`** `[username]` (add -r to remove the home directory)

**`usermod`** `[username]`

**`usermod`** `-s /sbin/nologin [username]` the user won't be able to log in.

Most of these options works for **`useradd`** and **`usermod`**

Option|Description

-|-

-a, --append|add the user to the supplementary group(s). use only with -G

-c, --comment `[COMMENT]` | full name of the user for the GECOS field.

-d, --home `[HOME_DIR]` |specify user's home directory

-e, --expiredate `[EXPIRE DATE]` |date on which the user account will be disabled (YYYY-MM-DD)

-f, --inactive `[INACTIVE]` |number of day after password expires until the account is disabled

-g, --gid `[GID]` |specify primary group

-G, --groups `[GROUPS]` |supplementary groups

-m, --move-home|moves the user's home directory to a new location, use with -d

-s, --shell `[SHELL]` |specify a new login shell for the user

-L, --lock|lock the user's account

-U, --unlock|unlock the user's account

## **Groups**

**`groupadd` **  `-g [GID] [group]` adds a group with the specified ID and name.

**`groupmod` **  `-g [GID] [group]` changes the ID of the specified group ( `-n` to change name).

**`groupdel` ** `[group]` deletes the specified group.

**`gpasswd`**  `-d [user][group]` remove the user from the group.

## **Password**

```none
  ||--------------max days (-M)------------||
  |            |    |             |
  |            |    |             |
  ||-min days (-m)-|     |-warn days (-w)-|||-inactive days (-I)-|
time|-----------------------------------------|---------------------|
  |                            |               |
last change date (-d)          password expiration date  inactive date
```

**`chage`**  `-l [username]` list user's current settings.

**`chage`**  `-E YYYY-MM-DD [username]` makes the account expire n the specified date.

**`chage`**  `-d 0 [username]` forces a password change on the next login.

**`chage`**  `-m 0 -M 90 -W 7 -I 14 [username]` change the settings to 0 days required to change password,  90 days for the password to expire, warning of password expiring 7 days before it happens, 14 days before the account inactivation.

Option|Description

-|-

-d `[n]`|change the last time the password was changed

-E `[YYYY-MM-DD]`|set date of account's expiration

-I `[n]`|days before the password becomes inactive

-m `[n]`|minimum age/time before changing the password

-M `[n]`|maximum age of the password

-W `[n]`|warning before the password expiration date

## **Permissions**

### **Standard permissions**

### Word model

**`chmod`** `WhoWhatWhich <filename>`

**r** read, **w** write, **x** execute.

**`chmod`** `g=rw- foo` sets read and write for the group of the file _foo_.

**`chmod`** `u+x script` adds the execute permission for the owner of the file _script_

### Shared table for the Word model

Word|Operator|Permission|Special bit

-|-|-|-

u (owner)|+ (add permission)|r (read)|s (suid, using u)

g (group)|- (remove permission)|w (write)|s (sgid, using g)

o (other)|= (set permission)|x (execute)|t (sticky, only directories)

### Numeric model

Number|Permission|Special bit

-|-|-

4|read|suid

2|write|sgid

1|execute|sticky

**` chmod `** ` 0700 foo ` equivalent to ` -rwx------ `

**` chmod `** ` 4554 script ` equivalent to ` -r-sr-xr-- `

**` chmod `** supports ` -R ` for recursive operations.

**` chown `** ` [user]:[group] ` change file ownership.

### **` umask `**

Change the default permissions applied to a new created file/directory using **` umask `**.

Write the value for the permissions excluded.

**`umask`** `0022` new files will be created as `-rwxr--r--` and `drwxr--r--`.

## **ACLs**

Check if a file has ACLs using **`ls`** `-l [file]`. If a `+` symbol is present next to the permissions column, then it contains ACLs.

You can set explicit permissions for users and groups that aren't the owner or primary group of the file.

Each ACL has a mask that gets recalculated every time you modify the ACL settings of a file.

The mask limits what permissions are effective (if the mask is `r--`, ACLs with `rw-` won't make use of the write permission).

**`getfacl`** `<filename>` get the ACL settings of the specified file. The command still works even if the file doesn't have any ACL settings.

**`setfacl`** `[option] [permissions]`

Option|Description

-|-

\-m|modify the ACL of a file or directory

\-x|remove the ACL entry of a file or directory

\--set-file=|apply the ACL from another file (use the `getfacl` output)

**`setfacl`**  `-m u:foo:r notes.txt` add (modify if it's already present) an entry specifying that the user `foo` has read permision on the file.

**`setfacl`**  `-m o:: notes.txt` changes the `others` permissions to `---`

**`setfacl`**  `-x u:foo: notes.txt` removes the entry for the user `foo`. Note that you don't need to specifiy any permissions, just leave the last field empty.

**`getfacl`** `fileWithACL |` **`setfacl`** `--set-file=- newFile` uses the output from the **`getfacl`** command and uses it to set the ACLs on `newFile`.

**`setfacl`**  `-m m::r <filename>` modify the mask to only allow the read permission.

**`setfacl`**  `-m d:u:rx <directory>` modify the default ACLs of the directory.

**`setfacl`**  `-k <directory>` remove all default settings on a directory.

**`setfacl`**  `-b <directory>` remove all ACLs on a directory.

## **Processes**

**`ps`** `aux` processes with `USER PID %CPU %MEM TTY STATUS`.

**`ps`** `lax` long listing style, avoid username lookup.

**`ps`** `-ef` display all processes.

**`ps`** `j` jobs running.

### **Process status**

Name|Flag|Kernel state|Description

-|-|-|-

Running|R|TASK_RUNNING|executing on a CPU or waiting to run

Sleeping|S|TASK_INTERRUPTIBLE|waiting for some condition (hw request, resources, signal)

 |D|TASK_UNINTERRUPTIBLE|sleeping but won't respond to signals

 |K|TASK_KILLABLE|like D but waiting for a signal to be killed

Stopped|T|TASK_STOPPED|stopped by being signaled (by user or another process)

Zombie|Z|EXIT_ZOMBIE|child process signals it's parent as it exists. Free resources

 |X|EXIT_DEAD|parent reaps the remaining child process structure. Now free

### **Jobs**

Useful when you have access to only ONE terminal.

**`[command]`** `&` the ampersand moves the program to the background automatically.

**`jobs`** display running jobs on the background.

**` fg` ** ` %[job ID]` bring job to the foreground.

**` bg` ** ` %[job ID]` resume stopped process in the background.

**` Ctrl + Z` ** suspends the process and send it to the background (use before **` bg` **).

**` kill` ** ` %[job ID]` kill the job running in the background.

### **` kill` command**

**` man` ** ` 7 signal` for more details.

Number|Name|Definition|Purpose

-|-|-|-

1|SIGHUP|Hangup|report termination of the controlling process of a terminal

2|SIGINT|Keyboard interrupt|interrupt from keyboard (**` Ctrl + C` **)

3|SIGQUIT|Keyboard quit|quit from keyboard (**` Ctrl + \` **)

9|SIGKILL|Kill, unblockable|abrupt program termination. Always fatal

15|SIGTERM|Terminate|termination signal, process should close properly

18|SIGCONT|Continue|resume process if stopped

19|SIGSTOP|Stop, unblockable|suspend the process

20|SIGTSTP|Keyboard stop|can be blocked or handled (**` Ctrl + Z` **)

**` kill` ** ` [PID]` kill the process with the default signal (SIGTERM,15).

**` kill ` **  ` -[signal] [PID]` send the specified signal (name or number).

**` kill ` **  ` -l ` list all the available signals.

**` killall ` ** ` [command pattern]` kill all the processes that matches the command pattern.

**` killall ` **  ` -[signal] [command pattern]` send the specified signal to all the process that matches the command pattern.

**` killall ` **  ` -[signal] -u [username] [command]` same as before but only those that belong to the specified user.

### **` pkill ` command**

It's like killall, and uses an advanced selection criteria.

#### Use **` pgrep ` ** to check which processes will be affected

Option|Name|Description

-|-|-

` [command] `|Command|processes matching that command

-U|User ID|processes owned by that user

-G|Group ID|processes owned by that group

-P|Parent|processes belonging to that parent process

-t|Terminal|processes controlled by that terminal

**` pkill ` ** ` [command pattern]`

**` pkill ` **  ` -U 1000 ` kill all the processes that belong to the user with ID 1000.

**` pgrep ` **  ` -l -u foo ` display all the processes running by the user ` foo `

**` w ` **  ` -f ` display who's logged into the system and their activities.

**` pstree ` **  ` -p [username] ` tree representation of the processes running by the specified user.

### **Process activity**

**` uptime ` ** display the load average of the last 1, 5 and 15 minutes.

**` grep ` ** ` "model name" /proc/cpuinfo |`  **` wc -l ` ** Count the cores of the machine (both physical and hyperthread ones).

Divide each number by the amount of cores. If the result is greater than 1 (>1), the CPU is overloaded.

**` top ` ** real-time process monitoring

#### List of columns

Name|Description

-|-

USER|process owner

VIRT|virtual memory is all the memory that the process is using

RES|physical memory used by the process

S|process state.

 |\[D] uninterruptable sleeping  \[R] Running or Runnable

 |\[S] Sleeping \[T] Stopped or Traced \[Z] Zombie

TIME|total processing time since the process started

COMMAND|process command name

#### Keystrokes

Key|Purpose

-|-

?  \/ h|help for interactive keystrokes

l t m|toggles for load, threads and memory header lines

1|toggle showing individual CPUs or a summary in header

s|refresh rate in decimal seconds (0.5,1,5)

b|reverse highlighting for Running processes; default = bold

B|enables use of bold in display

H|toggle threads

u,U|filter for username

M|sort by memory usage

P|sort by processor utilization

k|kill a process, ask for PID and signal

r|renice a process, ask for PID and nice_value

W|save the current display configuration for the next restart

q|quit

### **`nice` & `renice`**

Nice levels of a process goes from -20 to 19 for users.

**`top`** displays them from RT,-99 to 39.

Nice level of 20 for users translates as 0 for **`top`**.

Use **`nice`** for run programs, **`renice`** for already running programs.

**`nice`**  `-n [nice level] [command]` run the program with the specified nice level.

**`renice`**  `-n [nice level] [PID]` renice the process that is already running.

## **systemd & boot process**

**`systemctl`**  `-l` show what's running on the system without abbreviate the names.

**`systemctl`** `[option] [unit]`

The most common units: `service`, `socket`, `path`. Some processes has different units (like the `cups` process)

Option|Function|Option|Function

-|-|-|-

start|starts the unit|reload|reload the configuration of the unit (keep PID)

stop|stops the unit|restart|restarts the unit (new PID)

enable|allow unit to run at boot time|disable|prevent unit from running at boot time

is-enabled|check if the unit is enabled|is-active|check if the unit is active

status|display the status of the unit|mask|disable and hide unit

**`systemctl`** is also used for the boot targets.

A target is used to declare that we reached certain point in the boot process. Their names ends with `.target`

**`systemctl`** `list-units --type=target --all` display all the available targets and their current status.

**`systemctl`** `list-dependencies [target].target |` **`grep`** `target` display all the dependencies for that target.

**`systemctl`** `isolate [target].target` stops all the services that aren't required for the specified target. Not all targets can be isolated, only those with the `AllowIsolate=yes` flag.

### Important targets

Name|Usage

-|-

graphical|system supports multiple users, graphical and text-based logins

multi-user|system supports multiple users, text-based logins only

rescue|sulogin prompt, basic system initialization completed

emergency|sulogin prompt, initramfs pivot complete and system root mounted on / read-only

**`systemctl`** `set-default [target].target` change the default target.

You can override the default target at boot time by appending `systemd.unit=[target].target` to the kernel line.

### **Changing the root password**

1. Edit the GRUB entry of the system.

2. Search the line that starts with `linux16`

3. Append `rd.break` to the end of the line.

4. Press **`Ctrl + X`** to boot with the changes.

5. System will load and present a root shell. The actual boot system is mounted as read-only on /sysroot.

6. Remount the system with read-write permissions **`mount`**  `-oremount,rw /sysroot`.

7. Use **`chroot`** to treat `/sysroot` as the root of the file system tree **`chroot`** `/sysroot`.

8. Change the password of **root** **`passwd`** `root`.

9. Create the file `.autorelabel` to relabel the whole system with the right SELinux context **`touch`** `/.autorelabel`

10. Execute **` exit ` ** twice and the system will finish the boot process.

### GRUB (GRand Unified Bootloader)

**grub2** is the default boot loader on RHEL 7.

The main configuration is located at ` /boot/grub2/grub.cfg ` but you're not supposed to edit that file directly.

**` grub2-mkconfig ` ** generates a new config file.

**` grub2-mkconfig ` **  ` > /boot/grub2/grub.cfg ` generates a new config file and applies the changes permanently.

It's recommended to send the output to another file and review the changes before apply them.

**` grub2-install ` ** reinstalls the boot loader in case it's corrupt.

## **SELinux**

` /etc/selinux/config `

### **Recommended packages**

Package|Description

-|-

`policycoreutils-python`|adds the **`semanage`** command

`selinux-policy-devel`|more man pages related to SELinux

`setroubleshoot-server`|adds the **`sealert`**

**`sepolicy`**  `manpage -a -p /usr/local/man/man8` creates the SELinux man pages.

Security Enhanced Linux (SELinux) is an additional layer of system security.

Every single file in the system has a tag or context assigned.

SELinux labels have several contexts: user, role, type, and sensitivity.

RHEL uses the targeted policy by default, bases it's rules rules on the third context: type.

Every process goes through the SELinux vector table to look up what is allowed to do and which files are going to be used.

If the process is not allowed to do certain action or use certain file, an alert will be emitted.

By default, everything on Linux is denied. You can allow processes to do their stuff with policy rules.

There are three modes for SELinux:

Mode|Description

-|-

Enforcing|denies access to everything without explicit policies for that behaviour

Permissing|used to troubleshoot. Allow any interaction and logs the ones that should be denied.

Disabled|turns off SELinux. Requires a reboot to remove the labeling of SELinux.

It's better to use permissive mode than disable SELinux. The kernel will automatically maintain SELinux file system labels as needed, avoiding the need of relabeling during the system reboot.

**`getenforce`** shows the current SELinux mode.

**`setenforce`** `[Enforcing|Permissive|1|0]` changes the SELinux mode. Or we can edit the `/etc/selinux/config` file.

SELinux also has Booleans that can be used to tune the policy doing selective adjustments.

**`getsebool`**  `-a` display all the current Booleans and their values.

### **Changing SELinux contexts**

We can change contexts with the command **`chcon`** but it's not persistent.

**`chcon`**  `-t [context] <filename>` changes the context of the specified file.

Using the **`semanage`** command we can do persistent changes.

### **`semanage`** **is part of the package `policycoreutils-python`, maybe you'll have to install it.**

**`semanage`** `fcontext -l` show all the contexts on the database (supports RegEx).

**` semanage ` ** ` fcontext -a -t [context] [folder] ` add a new rule on the SELinux database. From now, every time you restore the context of the files inside the specified folder, the specified context will be applied.

**` semanage ` ** ` fcontext -a -t httpd_sys_content_t '/virtual(.*)?' ` set the context ` httpd_sys_content_t ` to the files inside of ` /virtual ` .

**` restorecon ` **  ` -Rv [directory] ` restores the context of the directory.

### **Remember to use ` restorecon ` after changing the directory's context.**

**` getseboolean ` **  ` -a ` list all the current booleans and their current status.

**` getseboolean ` ** ` [Boolean name] ` shows the status of the specified Boolean.

**` setsebool ` ** ` [Boolean] [on|off] ` toggles the Boolean.

**` setsebool ` **  ` -P httpd_enable_homedirs on ` set the ` httpd_enable_homedirs ` Boolean ` on ` and makes the change persistent ( ` -P ` ).

**` semanage ` ** ` boolean -l ` list all the Booleans with their current status, default value and description (use **` grep ` ** to filter what you're looking for).

**` semanage ` ** ` boolean -l -C ` show all the Booleans which value has been changed.

### **Troubleshooting SELinux**

There are times where SELinux may deny something. Most of the time the issue is an incorrect file context.

Check SELinux messages on `/var/log/audit/audit.log` using the command **`sealert`**  `-l`.

### **The package `setroubleshoot-server` must be installed in order to use `sealert`**

**`sealert`**  `-a /var/log/audit/audit.log` search and display SELinux messages in the `audit.log` file.

**`sealert`**  `-l [UUID]` display more information about the SELinux violation.

**scontext** is the source of the problem

**tcontext** is the target that the service was trying to do something to.

**`grep`** `[service] /var/log/audit/audit.log |` **`audit2allow`**  `-M mypol` generate a local policy module.

**`semodule`**  `-i mypol.pp` enable the policy we created.

## **`tar` command**

**`tar`** `[options]`

Option|Description|Option|Description

-|-|-|-

c|create an archive|x|extract an archive

f|name of the archive to work with|t|list the contents of the archive

p|preserve the permissions of files|P|don't strip leading **` /` ** from absolute paths

v|verbosity|compression|`z` gzip, `j` bzip2, `J` xz

**`tar`** `cf [resulting file name] [files to add]` this will create an archive.

Even if we don't use extensions on UNIX, it's good to add `.tar` at the end of the file.

**`tar`** `czf /root/foo.tar.gz /etc` creates a gzip-compressed tar archive, using the contents of the `/etc` folder.

**`tar`** `cjf /root/backup.tar.bz2 /var/log` creates a bzip2 archive.

**`tar`** `cJf /root/bar.tar.xz /etc/selinux` creates a xz archive.

**`tar`** `xzf /root/foo.tar.gz` extracts the content of the archive.

## **Logfiles**

### **rsyslogd**

\/var\/log|Description

-|-

messages|most syslog messages are logged here (except auth and email processing)

secure|security and authentication-related messages and errors (permissions and stuff)

maillog|mail server-related messages

cron|periodically executed tasks

boot.log|system startup-related messages (check first for troubleshooting boot problems)

Every message comes from a facility with a level of priority

Code|Priority|Severity|Code|Priority|Severity

-|-|-|-|-|-

0|emerg|system is unusable|4|warning|warning condition

1|alert|action must be taken immediately|5|notice|normal but significant event

2|crit|critical condition|6|info|informational event

3|err|non-critical error condition|7|debug|debugging-level message

**`man`** `1 logger` for more information.

`/etc/rsyslog.conf` contains predefined rules.

New rules must be created on files inside of `/etc/rsyslog.d` and end with `.conf`

``auth.* /var/log/mostsecure.log`` all messages from the `auth` facility will be logged on `/var/log/mostsecure.log`.

``*.info;mail.none;authpriv.none;cron.none /var/log/messages`` all the messages with priority above `info` (6) will be logged on `/var/log/messages`, except those that comes from the `mail`,`auth` and `cron` facilities.

Syslog entries have a defined format based on `timestamp:host:process:message` (you can add your own format on `/etc/rsyslog.conf`).

**`logger`** `-p [facility].[level] [message]` sends a fake message (useful to test configurations).

### **`journalctl` command**

Provided by **systemd**, writes the log on `/run` so it won't be saved by default.

`mkdir /var/log/journal` this will make **`journalctl`** logs persistent. Remember to assign the right permissions to this folder:

**`chown`** `root:systemd-journal /var/log/journal`

**`chmod`** `2755 /var/log/journal` equivalent to `rwxr-sr-x`.

Still won't be permanent, you need to change the rotation time on `/etc/systemd/journald.conf`, then send the `USR1` signal to `systemd-journald`.

**`journalctl`**  `-n [n]` display `n` amount of lines.

**`journalctl`**  `-p [priority name or number]` display the messages with the specified priority.

**`journalctl`**  `-f` real time output.

**`journalctl`** `--since [date (today| YYYY-MM-DD HH:MM:SS)] --until [date (today) | YYYY-MM-DD HH:MM:SS]` display the messages since the `--since` date to the `--until` date.

**`journalctl`**  `-o verbose` shows more information like:

Verbose|Description| | 

-|-|-|-

_COMM|name of the command|_EXE|path of the executable for the process

_PID|PID of the process|_UID|UID of the user running the process

_SYSTEMD_UNIT|**systemd** unit that started the process

**`journalctl`** `_SYSTEM_DUNIT=[unit].[type of unit] _PID=[PID]` display the logs of the specified process.

**`journalctl`**  `-b` display the last boot messages.

**`journalctl`**  `-b -1` output of the previous boot.

### **Time & date**

Make sure that your system's time is accurate.

**`timedatectl`** display information about how the system time is configured.

**timedatectl** option|Description| | 

-|-|-|-

list-timezones|list available timezones|set-ntp|enable or disable NTP synchronization

set-timezone|set the time to the selected timezone|set-time|set time using `YYYY-MM-DD hh:mm:ss`

**`tzselect`** select timezone interactively.

### **`chrony` & NTP**

`chronyd` is used to synchronize our system with an NTP server.

It uses servers from the NTP Pool Project (it can be changed to local servers).

In order to add an NTP server, we have to add a line on `/etc/chrony.conf`

`server classroom.example.com iburst` the option `iburst` uses four measurements in a short period of time for a more accurate initial clock synchronization.

Restart `chronyd` after making changes.

**`chronyc`** `sources -v` list the NTP servers that we're connected to.

## **Scheduling tasks**

### **`at` command**

The **`at`** is a small and powerful command that let us schedule tasks that won't be repeated

**`at`** `<TIMESPEC> [command]`

The `<TIMESPEC>` is quite flexible. You can use many different combinations.

**`echo`** `touch /root/hello |` **`at`** `now +1min` add a job to create the file `hello` in 1 minute from the moment it's executed.

**`at`** `noon +4 days < myscript` add a job to execute the file `myscript` at noon in four days since today.

**`at`** `<TIMESPEC> -q [queue] [command]` you have 26 queues (from a to z) to schedule tasks.

**`at`**  `-l` shows the current queue.

**`atq`** same as **`at`**  `-l`.

**`atrm`** `[job]` remove the specified job.

### **`crontab` command**

The benefit of **`crontab`** is that you can schedule recurring tasks.

Option|Description

-|-

-e|edit jobs for the current user

-l|list the jobs for the current user

-r|remove all jobs for the current user

-u|manage the jobs of another user (only **root**)

**`crontab`** `<filename>` if you specify a file, all the jobs will be removed and replaced by the jobs of that file. If no filename is specified, `stdin` will be used.

#### **Job Format**

``Minutes Hours Day-of-Month Month Day-of-Week Command``

``* * * * * command``

Symbol|Description

-|-

`*` |Don't care/always

`0-9` |number to specify a number of minutes or hours,a date or a week day (0 and 7 = Sunday, 1 = Monday)

`x-y` |range starting on `x` and ending with `y` both are included

`x,y` |lists, can include ranges (5,10-13,17)

`*/x` |indicate an interval of `x`

Three letter abbreviation|Month (Aug, Oct, Nov, Dec), weekday (Tue, Thu, Mon, Sun)

For the `command` part, we can use `%` to create a new line. It will be considered `stdin` for the `command` we're executing.

`0 9 2 2 * /usr/local/bin/yearly_backup` execute `yearly_backup` every February 2 at 9:00, doesn't matter the week day.

`*/7 9-16 * Jul 5 echo "Chime"` execute **`echo`** `"Chime"` during July but only on Fridays, from 9:00 to 16:59, repeating after 7 minutes.

### **Scheduling system `cron` jobs**

System cron jobs are defined in two locations: `/etc/crontab` and `/etc/cron.d/`.

Some packages install **`cron`** jobs and place them on `/etc/cron.d/`

Predefined folders for hourly, daily, weekly and monthly jobs can be found on `/etc`.

The directories are `cron.hourly cron.daily cron.weekly cron.monthly`.

Any scripts inside those files must have the execute permission activated.

`/etc/anacrontab` keep track of the scripts and the last time they were executed.

### **`systemd-tmpfiles` command**

**`systemd-tmpfiles`** reads configuration files located at `/usr/lib/tmpfiles.d/*.conf`, `/run/tmpfiles.d/*.conf` and `/etc/tmpfiles.d/*.conf`.

**`systemd-tmpfiles`** `[option]`

Option|Description

-|-

`--create`|create files and directories specified on the configuration files

`--clean`|remove all files with an age parameter configured

#### **Configuration files format**

`Type Path Mode UID GID Age Argument`

Column|Description

-|-

Type|action that systemd-tmpfiles should take

Path|path to file

Mode|permissions of the file/directory

UID|owner of the file

GID|group of the file

Age|maximum age of the file

Argument|depends on `Type`, written to the new file or used for a symlink

Action|Description

-|-

d|create directory if it doesn't exist yet

D|create directory if it doesn't exist yet or empty it if already exists

f|create file if it doesn't exist. `Argument` will be the content of the file

F|create or truncate a file. `Argument` will be the content of the file

L|create a symbolic link. `Argument` will be the file to reference

Z|recursively restore SELinux context and file permissions

`d /run/systemd/seats 0755 root root -` create a directory called `seats` on the `/run/systemd` directory with the permissions `rwxr-xr-x` that belongs to the user and group `root`.

This directory won't be automatically purged.

`D /home/student 0700 student student 1d` create a directory for the user and group `student` with `rwx------` permissions, it will be automatically deleted after 1 day.

`L /run/fstablink - root root - /etc/fstab` create a symbolic link to `/etc/fstab`, it won't be automatically purged.

#### **Configuration files priority**

If we have a configuration file that repeats it's name across `/etc/tmpfiles.d`, `/run/tmpfiles.d` and `/usr/lib/tmpfiles.d`, they have certain priority of which file gets to run.

`/etc/tmpfiles.d` \-\> `/run/tmpfiles.d` \-\> `/usr/lib/tmpfiles.d`

`/etc/tmpfiles.d` is top priority, then `/run/tmpfiles.d`, and last `/usr/lib/tmpfiles.d`.

## **Software management**

### **`yum` command**

**`yum`** is a command line tool that knows how to install programs and also knows their dependencies and the relationships between packages.

Option|Description| | 

-|-|-|-

help|display usage information|list|list all the packages available to install

repolist|list all the available repositories| |`package name` search this package (or another with similar name)

 |use the keyword `all` to display all of them, enabled and disabled| |`installed` list all the installed packages

search|search a package that matches the keyword|info|display information about the package specified

provide|search the package that provides the specified file|install|install the specified package (can be used with `.rpm` files)

update|update the specified package|remove|removes the specified package

history|show the list of transactions

 |`undo [n]` reverses the `n` amount of transactions

#### **Group options**

You can install whole groups of packages

Option|Description| | 

-|-|-|-

`list` |show all the package groups availables

`install` | install the specified group

`mark` | marks the group as installed, missing packages will be install on the next update

`info` | display more information about the group

 |`=` package was installed with the group | `+` will be installed with the group

 |`-`isn't installed and won't be installed with the group|`no marker` is installed but not with the group

**`yum`** `update kernel` update the kernel.

**`yum`** `install cowsay` install the package `cowsay`

### **Adding repositories**

**Repository files are located at `/etc/yum.repos.d/`.**

**`yum-config-manager`**  `--add-repo="[repository URL]"` this will create the proper `.repo` file for that repository.

This command belongs to the `yum-utils` package.

```bash
[Repository]
name=Super Repo
baseurl=http://myfirstrepo.com/
## if it's a 0, the repository is defined but not searched by default.
enabled=1
## check the public key when you grab or install a package from that repository.
gpgcheck=1
## where is the public key located
gpgkey=file:///etc/pki/rpm/gpg/RPM-GPGP-KEY
```

### **`rpm` command**

RPM files keep a naming scheme

`name-version-release.architecture`

`httpd-tools-2.4.6-7.el7.x86_64.rpm`

**`rpm`** `-q [option] [package/file]` query information about the specified package/file.

Option|Description
-|-
\-p| display information about the `.rpm` file specified
\-f|what packages provides the specified file
\-l|list of files installed by the specified package

\-c|list of configuration files

\-d|list of documentation files

\--scripts|list of scripts that may run on install or removal of the package

\--changelog|show the changelog of the specified package

**`rpm`**  `-i [package]` install the package.

## **Network**

We use the TCP/IP standard. TCP is used for large data, UDP for queries.

IPv4 addresses are made out of four octets.

Each IP address has a prefix which take part of the four octets available.

`172.17.5.3/16` means `172.17` is the network and `5.3` the host.

The network is the prefix.

Also, each IP has a netmask:

`255.255.0.0` where `255.255` belongs to the network and `0.0` to the host

Network|Host|Prefix

-|-|-

172.17|.5.3|/16

255.255|.0.0

192.168.5|.3|/24

255.255.255|.0

The machine on the subnet connects to the Gateway, which contacts with the rest of the world, for incoming or outcoming connections.

The Gateway connects to the internet using the public IP assigned by the DNS server owned by the ISP.

`0.0.0.0/0` is the default gateway.

Each network device has a MAC address. Also, their naming scheme on the system depends on how the BIOS recognizes the device:

Interface|Short name|Location|Short name

-|-|-|-

Ethernet|en|On-board|o

WLAN|wl|Hotplug|s

WWAN|ww|PCI|p

`enp6s0` translates as Ethernet PCI

**`ip`** `address` display information about the device and IP address

Note: commands like **`ifconfig`** and **`netstat`** are now deprecated.

**`ip`**  `-s link show` show stats of the interface.

**`ip`** `route` display routing information.

**`ping`** `-c[n] [ip/domain]` ping the `[ip/domain] n` amount of times.

**`tracepath`** `[domain]` traces the path to reach the specified domain.

**`ss`**  `-ta` socket statistics, `-t` for TCP sockets, `a` for all; display all the services running and what ports they're running on.

Option|Description|Option|Description

-|-|-|-

-n|numbers instead of names|-t|TCP sockets

-u|UDP sockets|-l|only listening sockets

-a|all sockets|-p|process using the sockets

### **IP Forwarding**

`net.ipv4.ip_forward = 1` add this line to `/etc/sysctl.conf`

After that, you need to apply the changes using `sysctl -p`

### **NetworkManager**

Configuration files on `/etc/sysconfig/network-script`

**`man`** `nm-settings`

Use **`nmcli`** to manage NetworkManager. Any changes to files that you do without using **`nmcli`** will be overwritten. You must turn on NetworkManager and do a `connection reload`, then down and up the connection.

**`nmcli`** `device [option]` manage devices (you can use `d`, `dev` instead of `device`).

Option|Description

-|-

status|list all devices

dis|bring down an interface and temporarily disable autoconnect

**`nmcli`** `net off` disable all manages interfaces.

**`nmcli`** `connection [option] [name of connection]` manage connections (you can use `c`, `conn` instead of `connection`).

Option|Description

-|-

show|view basic network information (more if you specify the connection name)

up|activate a connection

down|deactivate a connect (restart if autoconnect is on)

add|add connection

mod|modify a connection

del|delete a connection

reload|reloads configurations based on your manual changes

**`nmcli`** `con add help` shows all the options that can be used with this command.

### **Basic options for connections**

Common Options|Description

-|-

type|`ethernet wifi wimax ppoe` and more

ifname|device name

con-name|connection name

autoconnect|`yes` (default), `no`

There are many type-specific options, some are better for wired connections, others for wireless.

Note: ipv4 and ipv6 options are accessed using a dot `ipv4.addresses`.

IPv4 Options|Description

-|-

addresses|set the IPv4 address and gateway

dns|set the DNS

method|set `auto` for DHCP, `manual` for static

gateway|use when modifying the connection

**`nmcli`** `c a con-name "Wired Connection X" ifname enp0s3 type ethernet autoconnect yes ipv4.addresses "192.168.1.10/24" ipv4.gateway "192.168.254.254" ipv4.dns "192.168.254.254" ipv4.method manual` create a new static connection.

**`nmcli`** `c m "Wired Connection X" +ipv4.addresses "10.0.0.1/24"` the `+` means we're adding another value instead of replacing the current one.

**`nmcli`** `c a con-name "Dynamic" ifname enp0s3 type ethernet autoconnect yes ipv4.method auto` create a new DHCP conection.

#### **Configuration Options for `ifcfg` File**

Static|Dynamic|Either

-|-|-

BOOTPROTO=none|BOOTPROTO=dhcp|DEVICE=eth0

IPADDR0=`172.25.x.10`||NAME=`"System eth0"`

PREFIX0=`24`||ONBOOT=`yes`

GATEWAY0=`172.25.x.254`||UUID=`some UUID`

DEFROUTE=`yes`||USERCTL=`yes`

DNS1=`172.25.254.254`| | 

`USERCTL` allows non-root users to modify the network.

### **Hostname**

Hostnames aren't configured on the `/etc/hosts` file

The static host name is stored on `/etc/hostname`. If the file doesn't exist, a hostname hasn't been defined.

**`hostnamectl`** `status` display information about the hostname.

**`hostnamectl`** `set-hostname [hostname]` change the hostname of the machine.

**`getent`** `hosts [hostname]` test host name resolution with the `/etc/hosts` file.

**`host`** `[hostname]` test the DNS server connectivity.

## **firewalld**

**Mask iptables.service and ip6tables.service using `systemctl mask`**

**firewalld** replaces `iptables`,`ip6tables` and `ebtables`.

### **Predefined zones (`man 5 firewalld.zones`)**

Zone|Description

-|-

home|reject incoming traffic unless related to outgoing traffic or matching `ssh`, `mdns`, `ipp-client`, `samba-client` or `dhcpv6-client`

internal|same as the home zone

work|reject incoming traffic unless related to outgoing traffic or matching `ssh`, `ipp-client` or `dhcpv6-client`

public|used by default, reject incoming trauffic unless related to outgoing traffic or matching `ssh` or `dhcpv6-client`

external|reject incoming traffic unless related to traffic or matching `ssh`, outgoing IPv4 traffic forwarded through this zone is masqueraded

dmz|reject inconming traffic unless related to outgoing traffic or matching `ssh`

block|reject all incoming traffic unless related to outgoing traffic

drop|drop all incoming traffic unless related to outgoing traffic (without sending a response)

### **Pre-defined services**

Service|Description|Ports

-|-|-

ssh|local ssh server|`22/TCP`

dhcpv6-client|local DHCPv6 client|`546/UDP` or `fe80::/64` on IPv6

ipp-client|local IPP printing|`631/UDP`

samba-client|local Windows file and print sharing client|`137/UDP 138/UDP`

mdns|multicast DNS (mDNS) local-link name resolution|`5353/UDP` to the `224.0.0.251` IPv4 or `ff02::fb` IPv6

### **`firewall-cmd`** command

You can use the graphical tool **`firewall-config`** or **`firewall-cmd`** for command-line.

Changes can be made only runtime or permanent (adding the `--permanent` option).

You can also specify the zone using `--zone` (it's required for some commands).

CIDR = IP

Option|Description

-|-

`--get-default-zone`|query the current default zone

`--set-default-zone=<ZONE>`|change the default zone (runtime and permanent)

`--get-zones`|list all zones

`--get-active-zones`|list all zones currently in use

`--list-all`|list all configured interfaces, sources, services and ports for `--zone=<ZONE>` (otherwise default)

`--list-all-zones`|retrieve information for all zones

`--reload`|drop the runtime configuration and apply the persistent configuration

#### **Zone commands (any of these command uses `--zone=<ZONE>`)**

Option|Description

-|-

`--add-source=<CIDR>`|route all traffic coming from the `<CIDR>`

`--remove-source=<CIDR>`|remove the rule routing all trafic from the `CIDR` specified

`--add-interface=<INTERFACE>`|route all traffic from `<INTERFACE>` to the specified zone

`--change-interface=<INTERFACE>`|associate the interface with `<ZONE>`

`--add-service=<SERVICE>`|allow traffic to `<SERVICE>`

`--remove-service=<SERVICE>`|remove `<SERVICE>` from the allowed list for the zone

`--add-port=<PORT/PROTOCOL>`|allow traffic to the `<PORT/PROTOCOL>` for the zone

`--remove-port=<PORT/PROTOCOL>`|remove the `<PORT/PROTOCOL>` from the allowed list

**`firewall-cmd`**  `--set-default dmz` change the default zone to `dmz`.

**`firewall-cmd`**  `--permanent --zone=internal --add-source=192.186.0.0/24` assign traffic from `192.168.0.0/24` to the `internal` zone.

**`firewall-cmd`**  `--permanent -add-service=mysql` open the network ports for `mysql` on the `internal` zone.

## **`ssh` command**

Configuration file: `/etc/ssh/sshd_config`

**` ssh ` ** ` [remote username]@[remote host] ` connect through SSH to another machine.

**` ssh ` ** ` [remote username]@[remote host] [command] ` connects and automatically executes the specified command.

Wanna connect without passwords? You need a SSH key.

**` ssh-keygen ` ** generate a set of public and private keys.

The private key is stored at the file ` ~/.ssh/id_rsa ` and the public key at the file ` ~/.ssh/id_rsa.pub `.

You can also set a passphrase that you'll have to enter when connecting.

**` ssh-agent ` ** it will enter the passphrase for you during the time you're connected.

**` ssh-copy-id ` ** ` [remote user]@[remote host] ` copy the public key to the remote machine. Once it's done, we can use the password-less system to connect.

### **Disable root access**

1. Edit the file ` /etc/ssh/sshd_config `

2. Search and uncomment the line ` PermitRootLogin `

3. Change the ` yes ` for ` no ` (you can also set it to ` without-password ` for users that already copied their public key).

### **Disable Password Authentication**

1. Edit the file `/etc/ssh/sshd_config`

2. Search the line `PasswordAuthentication`

3. Replace `yes` for `no`.

## **Copying files between systems**

### **`scp` command**

Send files through SSH.

You can use the `-r` flag with **`scp`** to copy files recursively.

**`scp`** `[files to send] [remote user]@[remote host]:/path/to/put/files`

**`scp`** `/etc/hosts root@rmachine1:/root/copied` sends the local file `hosts` to the directory `/root/copied` on the remote machine.

**`scp`** `[remote user]@[remote host]:/file/to/copy /path/to/put/files` send a remote file to our machine.

### **`sftp` command**

SSH FTP interactive interface.

**`sftp`** `[remote user]@[remote host]` start an **`sftp`** session on the remote server.

You can use commands such as `ls`, `cd`, `mkdir`, `rmdir`, `pwd` to navigate.

`put` and `get` can be used to upload and download files.

### **`rsync` command**

Quite useful when you need to **synchronize** files.

**Important** use the `-n` option to simulate the **`rsync`** changes without applying them.

**`rsync`** copy files the first time, then it will only modify those that were affected/copy new files.

Option|Description| | 

-|-|-|-

v|verbosity output|a|archive mode

r|sync recursively the whole directory|l|sync symbolic links

p|preserve permissions|t|preserve timestamps

g|preserver group ownership|o|preserve files owner's

D|sync device files (only for troubleshoot)|H|preserve hard links

A|sync ACLs|X|sync SELinux context

**`rsync`** `[option] [files to synchronize] [/path/to/place/them]`

**`rsync`** `-av /etc/ /etcbackup` synchronize all the files from `/etc` with the ones on `/etcbackup`.

**`rsync`** `-av /home/student/foo.bar student@desktop1:/home/student/` synchronize the local files at the remote machine.

## **LDAP users**

`Lightweight Directory Access Protocol`, used in Active Directory and IPA Server.

**Install these packages:** `authconfig-gtk`, `sssd` and `krb5-workstation`.

There's also a terminal version of `authconfig-gtk` but it's deprecated.

In order to connect to a central LDAP Server, `authconfig` needs:

- The host name of the LDAP server(s).

- The base DN (Distinguished Name) of the part of the LDAP tree where the system should look for users (`dc=example dc=com`).

- If SSL/TLS is used to encrypt communications with the LDAP server, a root CA certificate that can validate the certificates is offered offered by the LDAP server.

Necessary Kerberos parameters:

- The name of the Kerberos realm to use.

- One or more key distribution centers (KDC). This is the host name of your Kerberos server(s).

- The host name of one of more admin servers.

**`getent`** `passwd <username>` test the LDAP + Kerberos configuration.

## **Partitions & File Systems**

### **Useful commands**

Command|Description

-|-

**` df ` **  ` -h ` |display filesystems with space on human readable format

**` du ` **  ` -h ` |display disk usage on human readable format

**` blkid ` **|show all file systems with their UUIDs

**` lsof ` **|show the processes using the specified directory/file

**` free ` **  ` -m ` |display memory usage in MiB

### **` mount ` command**

**` mount ` ** ` [device file or UUID] [mount point] `

**` mount ` **  ` -a ` mount all the file systems specified on ` /etc/fstab ` .

**` mount ` **  ` -o remount,rw /foo ` remounts ` /foo ` with read-write permissions.

### **` umount ` command**

**` umount ` ** ` [mount point] `

**` umount ` ** ` /filesystem-mounted ` unmount the filesystem mounted on ` /filesystem-mounted ` .

If the mount point is being accessed by a process, you can't unmount it (check with **` lsof ` **).

### **Partitions**

**MBR (` Master Boot Record `)**

- 4 partitions (maximum, 15 by using extended and logical partitions).

- Partition size of 2 TiB.

- Located at the first part of the scheme (boot block).

- **` fdisk` **

**GPT (` GUID Partition Table`)**

- Support for 128 partitions.

- Partition size of 8 ZiB.

- First block is the protective MBR, then the partitions table (backup at the end of the disk).

- **` gdisk` **

#### **` fdisk` & MBR partitions**

**` fdisk` ** `[device]`

**` fdisk` ** `/dev/sdb` create MBR partitions on `/dev/sdb`.

Key|Description

-|-

d|delete partition

m|help

n|create partition

p|display partitions available in the disk

t|change partition's type (L to see table of types)

w|write changes

Run **`partprobe`** `[device]` after writing the changes.

#### **`gdisk` & GPT partitions**

**`gdisk`** `[device]`

**`gdisk`** `/dev/sdb` create GPT partitions on `/dev/sdb`

The keys are like the ones used for **`gdisk`** except for others that are new.

Use `?` or `m` to see the help list of commands.

Remember to run **`partprobe`** `[device]` after you write the changes on the disk.

### **Creating file systems**

After a block device has been created, we need to format it.

**`mkfs`**  `-t [type] [device]`

**`mkfs`**  `-t ext4 /dev/sdb1` apply the `ext4` file system to `/dev/sdb1`.

**`mkfs`**  `-t xfs /dev/sdc3` apply the `xfs` file system to `/dev/sdc3`.

### **Swap partitions**

Swap partitions are like extra RAM.

Create a new partition with **`fdisk`** or **`gdisk`**, assigning the type `Linux Swap`.

**`mkswap`** `[device]`

**`swapon`** `[device]`

**`swapon`** `-p [priority] [device]` the priority means which swap partition will be used first (higher value means more priority of use).

**`swapon`**  `-a` activate all the partitions marked as swap space.

**`swapon`**  `-s` summary of swap partitions.

### **/etc/fstab**

**An incorrect /etc/fstab entry may render the machine unbootable.**

**Use `mount`**  `-a`**to check if all the entries are correct.**

Entries on `/etc/stab` will be automatically mounted when the system boots.

`UUID=[UUID] [mount point] [file  system type] [options during mount] [dump flag and fsco order]`

```shell
UUID=some-UUID     /mnt/storage   xfs    defaults    0 0
/dev/sda           /              xfs    defaults    0 0
```

You can use the device name instead of UUID. The problem is that device numbers are assigned when disks are discovered during the boot.

If you change a disk, it may take the same device name.

### **LVM (Logical Volume Management)**

#### **Physical Volume (PV)**

It's the hardware itself, lowest level of LVM.

Your partitions must have the `Linux LVM` type to be used as PV.

Command|Description

-|-

**`pvcreate`** `/dev/sda3 /dev/sdb2`| mark `/dev/sda3` and `/dev/sdb2` as PVs

**`pvmove`** `/dev/sda4`| move PEs from `/dev/sda4`

**`pvremove`** `/dev/sda4`| remove the PV label to `/dev/sda4`

**`pvs`**| display PVs

**`pvdisplay`**| display more information about PVs (specify a PV to get more details)

#### **Volume Group (VG)**

Made with PVs. It can hold Logical volumes.

Command|Description

-|-

**` vgcreate ` ** ` [name] [physical volumes] ` |create a new volume group

 |` -s [n] ` define PE size, ` -s 16M ` define each PE to be 16 MiB

**` vgremove ` ** ` [VG name] ` |delete the VG, leaving the PV available for other volume group

**` vgextend ` ** ` [VG name] [PV] ` |extend the size of the VG

**` vgreduce ` ** ` [VG name] [PV] ` |reduce the size of the VG

**` vgs ` **|display VGs

**` vgdisplay ` **|display more information about VGs (specify a VG to get more details)

#### **Logical Volume (LV)**

Logical volumes are created inside of VG.

Command|Description

-|-

**` lvcreate ` **  ` -n [LV-name] -L [size] [VG-name] ` |create a new logical volume

 |use ` -l ` to assign a size in extents

**` lvremove ` ** ` /dev/[VG]/[LV] ` |remove the LV

**` lvextend ` **  ` -L [size] /dev/[VG]/[LV] ` |extend the size of the LV. ` +300M ` add 300 MiB to the LV

 |` -l ` for increase the size in extents

**` lvreduce ` **  ` -L [size] /dev/[VG]/[LV] ` |reduce the LV, ` [size] ` is the new size for the LV (you can use ` -l ` for PE)

**` lvs ` **|display LVs

**` lvdisplay ` **|display more information about LVs (specify a LV to get more details)

Once a LV has been created, you can format it with **`mkfs`**. The path will be `/dev/[VG]/[LV]`.

**Before reducing or after extending a LV, use the command `resize2fs`** `/dev/[VG]/[LV] [new size]`

The new size is only required for reducing.

## **NFS & SMB**

### **NFS**

We must enable and start the unit `nfs-secure`.

Install **`autofs`** for automount the shares.

NFS can be protected using Kerberos. It will requiere a `/etc/krb5.keytab` and additional authentication configuration (Kerberos realm).

Security methods|Description

-|-

none|anonymous access to the files, writes to the server (if allowed) will be allocated UID and GID of nfsnobody.

sys|standard Linux permissions for UID and GID values. Default if another isn't specified

krb5|client must prove identity using Kerberos and then standard Linux permissions

krb5i|cryptographically strong guarantee that the data in each request hasn't been tampered

krb5p|encryption to all requests between the client and the server. Performance impact

#### **Mount an NFS share**

**`mount`**  `-t nfs -o sync [server]:/share /mountpoint` in this case, the mountpoint should be already created.

We can add the option `sec=` to choose which security method we're using.

`/etc/fstab` entry to automount NFS shares on boot.

```shell
[server]:/share  /mountpoint  nfs sync  0 0
```

### **`autofs`**

**Install `autofs` and activate the unit.**

#### **Creating and automount**

Create a new file at `/etc/auto.master.d` like `home.autofs`

```shell
/shares /etc/auto.demo
```

The base point is `/shares` and the information to create it's content can be found at `/etc/auto.demo`.

Note: Those files at `/etc/` follow a convention of using `auto` and then something else at their names.

`/etc/auto.demo`

```shell
* -rw,sync [server]:/shares/&
```

In this case, the ampersand (\&) will match the asterisk at the beginning.

The mount point is an asterisk and the subdirectory on the source location is an ampresand.

`/etc/fstab` entry to automount a NFS share that uses Kerberos

```shell
[server]:/share /mountpoint nfs sec=krb5p,rw 0 0
```

#### **Mount an SMB share**

**`mount`**  `-t cifs -o guest //[server]/share /mountpoint`

The `-t cifs` option is the file system type for SMB shares and the `-o guest` tells **`mount`** to try and authenticate as a guest account without a password.

#### **Secure SMB share**

We can also specify certain security parameters (like username, password)

`/credentials` file

```shell
username=username
password=password
domain=domain
```

It should be stored somewhere secure with only root access (0600).

`/etc/fstab` entry for secured SMB share

```shell
//[server]/share /mountpoint cifs creds=/[credentials] 0 0
```

### **Q. Set hostname of system as host.example.com**
### Answer :-
```
check hostname of your system before changing it.

[root@localhost ~]# hostname (check hostname)

[root@localhost ~]# hostnamectl set-hostname host.example.com
```

[root@localhost ~]# exec bash (sync hostname on terminal)

[root@localhost ~]# cat /etc/hostname

host.example.com

```

### **Q. Create group named newgroup wiht GID 3099**

### Answer:-

```

[root@localhost ~]# groupadd -g 3099 newgroup

[root@localhost ~]# id newgroup (check groupid)

[root@localhost ~]# cat /etc/group

```

### **Q. Create a stratis pool of size 2GB with name newpool and create a filesystem with name newpart1 it should be mounted on /mnt/partition.**

### Answer: -

Install neccessay packages to do this task.

- yum install stratisd

- yum install stratis-cli

procedure to create stratis pool.

```

[root@localhost ~]# yum install stratisd

[root@localhost ~]# yum enable stratisd

[root@localhost ~]# stratis pool create newpool /dev/sdb (here /sdb is disk of size 2GB)

[root@localhost ~]# stratis pool list (checkout pools)

[root@localhost ~]# stratis filesystem newpool newpart1 (creates filesystem)

[root@localhost ~]# stratis filesystem list (check filesystem)

[root@localhost ~]# mkdir /mnt/partition

[root@localhost ~]# mount /stratis/newpool/newpart1 /mnt/partition

To mount a partition automatically under /mnt/partition make the entry of partition in /etc/fstab file. use following command:

[root@localhost ~]# vim /etc/fstab (add partition entry here)

```

### **Q. set the recommend porfile to vm**

### Answer :-

```

[root@localhost ~]# yum install tuned

[root@localhost ~]# systemctl enable tuned

[root@localhost ~]# tuned-adm recommend

it will show recommended profile as virtual-guest

[root@localhost ~]# tuned-adm profile virtual-guest
```

### **Q. List the available tuning profiles and identify the active profile.**

### Answer :-
```
First check tuned.service is installed and enabled.

[root@localhost ~]# yum install tuned

[root@localhost ~]# systemctl start tuned

[root@localhost ~]# systemctl enable tuned

[root@localhost ~]# tuned-adm list (command gives list of available profile)

[root@localhost ~]# tuned-adm active (command gives current active profile)
```

# <center> Tunning System Performance </center>

## Objectives :-

- Killing Processes
```

- Monitoring Process Activity

- Tunning Profiles

| Index | Topic Covered
|

| ----- | ---------- |

| **1** | <ul><li>Process Control using Signals</li><li>Fundamental Process Management Signals</li><li>working of pkill commnad</li><li>working of SIGKILL command</li></ul>
|

| **2** | <ul><li>Understanding load average calculation</li><li>current load average display using uptime and lscpu like commands</li><li>Real-time Process monitoring using various commands.</li></ul> |

| **3** | <ul><li>Tunning system</li><li></li>Configuring static tunning<li>Configuring Dynamic Tuning using tuned service</li><li>Managing profiles using tuned command</li></ul> |

### **Q. Create user fred with a user id 3945 give password as iamredhatman.**

### Answer :-

```

[root@localhost ~]# useradd -u 3945 fred

[root@localhost ~]# passwd fred

changing password for user fred

New password:iamredhatman

Retype new password:iamredhatman
```

passwd: all authentication token updated successfully.

```

### ***Create a user named Eric, and the user id should be 1234, and the password should be Eric123.***

### Answer :-

```

[root@localhost ~]#  useradd -u 1234 Eric

[root@localhost ~]#  passwd Eric

changing password for user Eric

New password:Eric123

Retype new password:Eric123

passwd: all authentication token updated successfully.

### **Q.Copy /etc/fstab to /var/tmp name admin, the user1 could read, write and modify it, while user2 without any permission.**

### Answer :-

```

[root@localhost ~]# cp /etc/fstab /var/tmp

[root@localhost ~]# groupadd admin

[root@localhost ~]# useradd user1
```

[root@localhost ~]# useradd user2

[root@localhost ~]# chgrp admin /var/tmp/fstab

[root@localhost ~]# getfacl /var/tmp/fstab (check owner, group and permissions)

[root@localhost ~]# setfacl -m u:user1:rwx /var/tmp/fstab

[root@localhost ~]# setfacl -m u:user2:--- /var/tmp/fstab

[root@localhost ~]# getfacl /var/tmp/fstab (again check group, and user permissions)

```

### **Q. add users named newuser with id 1029. set password expiration date as 2023-05-23.**

### Answer :-

```

[root@localhost ~]# useradd -u 1029 newuser

[root@localhost ~]# passwd newuser

changing password for user newuser

New password:newuser123

Retype new password:newuser123

Now set expiration date to password

[root@localhost ~]# chage -E 2023-05-23 newuser

```

### **Q. Create a vdo with size 10GB with name firstvdo and mounted on /mnt/vdo**

### Answer: -

```

install vdo and enable service

[root@localhost ~]# yum install vdo -y

[root@localhost ~]# systemctl enable vdo

[root@localhost ~]# vdo create --name firstvdo --device /dev/sdd --vdoLogicalSize 10G

[root@localhost ~]# vdo list (check all vdo)

[root@localhost ~]# vdo status (status of vdo)

Now format the vdo using mkfs command

[root@localhost ~]# mkfs.xfs /dev/mapper/firstvdo

Now check vdo stats

[root@localhost ~]# vdostats

[root@localhost ~]# mkdir /mnt/vdo

mount the vdo on /mnt/vdo

To mount a vdo automatically under /mnt/vdo make the entry of vdo in /etc/fstab file. use following command:

[root@localhost ~]# vim /etc/fstab (add vdo entry here)

```

# <center> **Download, Install, Update, manage software packages** </center>

## Objectives :-

- Configuring yum locally

- installing software packages with yum

- updating packages with yum

- Enabling and disabling software repositories

### Cofiguring yum

cofiguring yum using **config-manager** command

```

[root@localhost ~]# yum config-manager --add-repo (repo url or repo path)

we can configure repositories with dnf command also.

[root@localhost ~]# dnf config-manager --add-repo (repo path or url)

```

### Installing and removing software packages with yum.

**yum install PackagesName** obtains and install software packages.
```

```

[root@localhost ~]# yum install nmap (installs software called nmap)

```

**yum remove PackagesName** removes software packages.

```

[root@localhost ~]# yum remove nmap (removes software called nmap)

```

### Updating package with yum

**yum update PackageName** updates specific software

```

[root@localhost ~]# yum update httpd (here httpd is package name)

```

### Enabling and disabling software repositories

**yum config-manager** command used to enable or disable repos.

yum config-manager --enable (repo url or name) enables repository

```

First checkout which repo need to enable run following command

[root@localhost ~]# yum repolist all

[root@localhost ~]# yum config-manager --enable epel-testing-source

```

yum config-manager --disable (repo url or name) disable repository

```

[root@localhost ~]# yum config-manager --disable epel-testing-source

```

#kickstart

#grep

#vim

#scheduletasks

#processespriorities

#acls

#selinux

#networkusers

#partitions

#lvmstorage

#nfsnetworkfilesystem

#smbnetworkstorage

#boottroubleshooting

#firewalld

#review

#kickstart

The idea is to automate the installation of RHEL with Kickstart.

By default, when you start an installation, it will ask you all the needed options for the install.

Kickstart allows us to pre-configure the options for installations.

Kickstart = JumpStart (Oracle) / Unattended installation (Windows)

Anaconda is the program that runs the installation.

In a physical system, we can boot off from a a Kickstart server that contains the system to boot and a yum repository where we can find the packages needed for the installation.

We can reference a Kickstart file both from the Kickstart server where the installation media is located or locally at the machine that we're performing the installation.

Kickstart file

Begin with a list of commands that define how the target machine is to be installed

#comments - ignored by the installer

Additional sections begin with a line that starts with a % character and end with a line with the %end directive

%packages specifies the software to be installed on the target system. Package groups can be specified by name or ID, and start with an @ character.

%pre and %post will execute before (%pre) and after (%post) the installation.

%pre will be executed before any disk partitioning is done (identify hardware, change configuration based on hardware, etc)

%post will be executed after the installation is done (anything that you want to script).

Kickstart configuration file commands

---

url --url="ftp://installserver.example.com/pub/RHEL7/dvd"

---

Where's the installation media located?

---

repo --name="Custom Packages" --baseurl="ftp://repo.example.com/custom"

---

Tells Anaconda where to find the packages for installation. The yum repository must be a valid one.

text - forces text mode install

---

vnc --password=redhat

---

Allows the graphical installation to be viewed remotely via VNC

askmethod - don't automatically use the CD-ROM as the source of packages when installation media is detected in the CD-ROM drive

Partitioning commands

---

clearpart --all --drives=sda,sdb --initlabel

---

clearpart - clears the specified partitions before installation

---

part /home --fstype=ext4 --label=homes --size=4096 --maxsize=8192 --grow

---

part - specifies the size, format, and name of a partition (grow means that it will take all the remaining space)

---

ignoredisk --drives=sdc

---

ignoredisk - ignores the specified disks when installing

---

bootloader --location=mbr --boot-drive=sda

---

bootloader - defines where to install the bootloader

We don't work with raw partitions anymore. They're not resizable.

Now we use logical volumes!

---

part pv.01 --size=8192

volgroup myvg pv.01

logvol / --vgname=myvg --fstype=xfs --size=2048 --name=rootvol --grow

logvol /var --vgname=myvg --fstype=xfs --size=4096 --name=varvol

---

Create a big partition, assign a volume group to it, then create logical volumes inside the volume group.

zerombr - disks whose formatting is unrecognized are initializated

Network commands

---

network --device=eth0 --bootproto=dhcp

---

Configures network information for target system and activates network devices in installer environment

---

firewall --enabled --service=ssh,cups

---

Defines how the firewall will be configured on the target system

Configuration commands

---

lang en_US.UTF-8

---

Required. Sets the language to use during installation and the default language of the installed system

---

keyboard --vckeymap=us --xlayouts='us','us'

---

keyboard --vckeymap=es --xlayout='es','us'

---

Sets the system keyboard type

---

timezone --utc --ntpservers=time.example.com Europe/Amsterdam

---

Defines the timezone, NTP servers and whether the hardware clock uses UTC.

---

auth --useshadow --enablemd5 --passalgo=sha512

---

Required. Sets up the authentication options for the system.

---

rootpw --plaintext redhat

---

Defines the initial root password. It's not a good idea to set it as plain text.

---

rootpw --iscrypted $6$KUnFfrTz08jv.PiH$YlBb0t...

---

We can also use the hash generated by another encrypting command

---

selinux --enforcing

---

Enforces the use of SELinux, we can also set other options using that command.

---

services --disabled=network,iptables,ip6tables --enabled=NetworkManager,firewalld

---

Modifies the default set of services that will run under the default runlevel

It's possible to create new groups and users

---

group --name=admins --gid=10001

user --name=jdoe --gecos="John Doe" --groups=admins --password=changeme --plaintext

---

Miscellanous commands

---

logging --host=loghost.example.com --level=info

---

Defines how Anaconda will log during the installation

---

firstboot --disable

---

Determines whether firstboot starts the first time the system is booted.

Firstboot is a program that finishes configuration that is required.

If you configured everything on the Kickstart file, there's not need to have firstboot enabled.

---

[reboot|poweroff|halt]

---

Specify what should happen after the installation finishes

The ksverdiff utility from the pykickstart package is useful for identifying changes of a Kickstart file between two versions of RHEL or Fedora.

---

ksverdiff -f RHEL6 - RHEL7

---

Identify changes in syntax from RHEL 6 to RHEL 7

Available versions are listed in the top of the file /usr/lib/python2.7/site-packages/pykickstart/version.py

Example Kickstart file

---

```
#version=RHEL7
# System authorization information
auth --useshadow --enablemd5
# Use network installation
url --url="http://classroom.example.com/content/rhel7.0/x86_64/dvd"
# Firewall configuration
firewall --enabled --service=ssh
firstboot --disable
ignoredisk --only-use=vda
# Keyboard layouts
keyboard --vckeymap=us --xlayouts='us','us'
```

```
# System language
lang en_US.UTF-8
# Installation logging level
logging --level=info
# Network information
network --bootproto=dhcp
# Root password
rootpw --iscrypted $6$/h/Mumvarr2dKrv1$Krv7...
# SELinux configuration
selinux --enforcing
# System services
services --disabled="kdump,rhsmcertd" --enabled="network,sshd,rsyslog,chronyd"
# System timezone
timezone --utc America/Los_Angeles
# System bootloader configuration
bootloader --location=mbr --boot-drive=vda
# Clear the Master Boot Record
zerombr
# Partition clearing information
clearpart --all --initlabel
# Disk partitioning information
part / --fstype="xfs" --ondisk=vda --size=10000

%packages
@core                   // install this group
chrony
```

```
cloud-init
dracut-config-generic
dracut-norescue
firewalld
grub2
kernel
rsync
tar
-NetworkManager             // and remove those
-plymouth                   // with the minus in front

%end

%post --erroronfail

# For cloud images, 'eth0' _is_ the predictable device name, since
# we don't want to be tied to specific virtual (!) hardware
rm -f /etc/udev/rules.d/70*
ln -s /dev/null /etc/udev/rules.d/80-net-name-slots.rules

# simple eth0 config, again no hard-coded to the build hardware
cat > /etc/sysconfig/network-scripts/ifcfg-eth0 << EOF
DEVICE="eth0"
BOOTPROTO="dhcp"
ONBOOT="yes"
TYPE="Ethernet"
```

USERCTL="yes"

PEERDNS="yes"

IPV6INIT="no"

EOF

%end

---

In a kickstart file, missing required values cause the installer to interactively prompt for and answer or to abort the installation entirely.

Quiz:

Section of a Kickstart configuration file that specifies what software is installed on the new system. %packages

Required Kickstart command that configures how users access the system.
auth

Location of the software used by Kickstart to install a system. url

Scripting in a Kickstart configuration file that is executed after the software is installed on a system. %post

Kickstart command that specifies which partitions should be cleared before installation.
clearpart

Modifies which services will start by default at system boot.
services

Defines the default authentication credentials for the superuser.
rootpw

Kickstart command that specifies the size, format, and name of a disk partition.
part

Kickstart command used to specify NTP servers. timezone

Determines the network configuration for the installation and the target system.
network

Deploying a new virtual system with Kickstart

We can use system-config-kickstart utility to create Kickstart files. For validate them, the ksvalidator utility is available.

---

system-config-kickstart

---

Opens a GUI to create a Kickstart file

Most of the time, creating a Kickstart configuration file from scratch with a text editor is rare. The Anaconda installer creates a file called /root/anaconda-ks.cfg that contains the Kickstart directives that can be used to generate the freshly installed system.

It's a good idea to check the Kickstart file with a text editor to be sure that you're satistied with the syntax.

Always check the syntax of the file before using it.

Reasons for creating a Kickstart file manually instead of using system-config-kickstart:

- The GUI and/or system-config-kickstart is unavailable.

- Advanced disk partition configuration instructions are needed. system-config-kickstart does not support LVM and software RAID.

- Individual packages need to be included or omitted (not just groups).

- More advanced scripting is needed in the %pre and %post sections.

---

ksvalidator /tmp/anaconda-ks.cfg

---

Validates the file specified. No output means there isn't any errors on the file.

ksvalidator can't tell what's the purpose of your system. Only checks for errors on syntax.

You can publish the Kickstart configuration file locally or using a server.

Once a Kickstart method is chosen, the installer must be told where the Kickstart file is located.

In order to specify the Kickstart file, we need to pass the argument ks=[location] to the installation kernel.

ks=http://server/dir/file

ks=ftp://server/dir/file

ks=nfs:server:/dir/file

ks=hd:device:/dir/file

ks=cdrom:/dir/file

When you're booting from a RHEL ISO, we can press tab while selecting the Install option to set some options for the installation (here's when we add the ks option).

For virtual machines installations using the Virtual Machien Manager or virt-manager, the Kickstart URL can be specified in a box under URL Options.

Quiz:

5 Use system-config-kickstart to create a Kickstart configuration file.

3 Use a text editor to add logical volume management commands to the Kickstart configuration file.

1 Check the configuration file for syntax errors with ksvalidator.

6 Publish the Kickstart configuration file via HTTP, FTP, or NFS.

2 Boot Anaconda from installation media.

4 Specify the ks= option to point the installer to the Kickstart configuration file.

#grep

Regular Expressions Fundamentals

It's fairly complex and can be used on multiple tools that supports RegEx

It's like a programming language somehow

We need to translate what are we looking for into some syntax that looks exactly what we're looking for.

| Symbol | Usage | Example | Applies for |
|---|---|---|---|
| ^ | beginning of the line | ^cat | category |
| $ | end of the line | $dog | chilidog |
| . | wildcard for a single character | c.t | cat\|cbt\|cct\|c1t\|c[whatever]t |
| * | multiplier for any amount of characters | c*t | cat\|cbt\|caaaaat\|cbbbbbt\|cccasdt |

.*  zero to infinitely many characters  c.*t  ct|cat|coat|culvert

.\{\}  explicit multipler  c.\{2\}  coat

\< \>  word boundary  \<ipsum\>  Lorem ipsum et...

[]  options for a single character  c[abc]t  cat|cbt|cct

Quiz:

^Au.*U

Installed

^i

error

Error

s$

Option  Function

-i  do not enforce case sensitivity

-v  display lines that DO NOT contain matches to the RegEx

-r  recursively search through group of files and directories

-A [X]  display X of lines after the RegEx match

-B [X]  display X of lines before the RegEx match

-e  multiple RegEx can be supplied and will be used with a logical or

-n  shows line number

---

grep 'cat$' /usr/share/dict/words

---

Search for all the words that ends with cat

---

ps aux | grep '^student'

---

Only lines that start with student

---

grep -i -v 'cat' dogs-n-cats

---

Display all the lines that doesn't match with 'cat'

---

grep -v '^[#;]' <filename>

---

Show all the lines that aren't commented (because they have a # OR a ;)

---

grep -e 'cat' -e 'dog' dogs-n-cats

---

Show all the lines that contains 'cat' on them and all the lines that contains 'dog' on them

#vim

vi - Visual Interface

vim - VI IMproved

In before, most text editors were line-based (ed, now ex)

When an unprivileged user invokes the command vi on RHEL 7, the command will be executed as vim because there's an alias for it.

If the user UID is less than or equal to 200, vi will be executed.

There are three distinct variations of vim that can be installed on RHEL

vim-minimal only provides vi and related commands (like rvi, the restricted version that cannot spawn commands or a shell). Default on minimal installation

vim-enhanced provides vim (and friends), provinding features such as syntax highlighting, file-type plug-ins and spell checking

vim-X11 provides gvim (vim running in its own graphical window instead of a terminal)

vim has three primmary modes

Command mode This mode is used for file navigation, cut and past, and simple commands. Undo, redo and others are also performed from this mode.

Insert mode This mode is used for normal text editing. Replace mode is a variation on insert mode that replaces text instead of inserting it

Ex mode This mode is used to save, quit and open files, as well as search & replace and other more complex operations. From this mode is possible to insert the output of programs into the current file, configure vim, and much more (equivalent to ex)

Quiz:

Command mode

Insert mode

Ex mode

Basic vim workflow

---

vim [name of the file]

---

Opens the specified file with vim

What if the file doesn't exists? vim will create it for you after the first save.

| Key | Result |
| --- | --- |
| i | Switch to insert mode and start inserting before the current cursor position. |
| a | Switch to insert mode and start inserting after the current cursor position. |
| I | Move the cursor to the start of the current line and switch to insert mode. |
| A | Move the cursor to the end of the current line and switch to insert mode. |
| R | Switch to replace mode, starting at the character under your cursor. |
| o | Open a new line below the current one and switch to insert mode. |
| O | Open a new line above the current one and switch to insert mode. |

I'm sorry, teacher, but my arrow keys doesn't work properly

Oh, well, you can use h|l and j|k to move around on command mode

| | |
| --- | --- |
| ^ | Move to the beginning of the current line. |

$ Move to the end of the current line.

gg Move to the first line of the document.

G Move to the last line of the document.

[X]g Move to the specified line number

:wq Save and quit the current file.

:x Save the current file if there are unsaved changes, then quit.

:w Save the current file and remain in editor.

:w <filename> Save the current file under a different file name.

:q Quit the current file (only if there are no unsaved changes).

:q! Quit the current file, ignoring any unsaved changes.

! Forces an action.

:help [subject] Shows helps about the specified subject. Without a subject specified, the default help will show up.

Editing with vim

Key Result

w Move cursor to the beginning of the next word (W includes punctuation).

b Move cursor to beginning of previous word (B includes punctuation).

( Move cursor to beginning of current or previous sentence.

) Move cursor to beginning of next sentence.

{ Move to beginning of current/previous paragraph.

} Move cursor to beginning of next paragraph.

c Change command. Must be followed by a movement command (cw the current cursor position to the end of the current word).

cc Replace the whole line.

C Equivalent to c$.

r Replace the letter where the cursor is located.

~ Change the case of the character under the cursor.

d Delete command. Must be followed by a movement command (dw deletes the current word).

dd Deletes the whole line.

D Delete from the cursor to the end of the line (d$).

All movement commands can be prefixed by typing a number (5w to move the cursor five words or 12j to move the cursor 12 lines down)

Copy & paste is called yank & put

You can't cut something on vim, just yank, put, go back and delete the line you don't want.

Key Result

y Yank (copy). Must be followed by movement commands (5yaw will copy the current word and the next four).

yy Yank the entire line.

p Put (paste). Put after the current cursor position.

P Put before the current cursor position.

vim has multiple registers to save yanked stuff.

Normal registers are called a to z, and are selected by putting "[registername] between the count for a command and the actual command.

3"tyy copies the current and the next two lines into the t register.

"[a-z]  Saves the yanked text into the specified register.

"[0-9]  Special numbered registers.

"0 will always have a copy of the most recent yanked text, while "1 will have a copy of the most recent deleted text. The contents will be shifted when new text is changed or deleted.

Visual mode

It's easier to select text on visual mode. The selection style depends on the shortcut to enter visual mode.

v  character-based (select each character).

V  line-based (select each line).

Ctrl+V  block-based (you can select blocks across multiple lines).

Searching

It can be done in two ways based on the current cursor position

/  search forward.

?  search backward.

After entering search mode, a RegEx can be typed to search for and pressing Enter will jump to the first match (if any).

To search for the next or previous match, use n and N respectively.

* will search forward for the word under the cursor.

Search & Replace

Search and replace in vim is implemented in ex mode and uses the same syntax as one would use with sed.

ranges/pattern/string/flags

range can be a line number (42), a range of line numbers (1,7 for lines 1-7), a search term (/README\.txt/), % for all the lines in the current document (search and replace normally only works on the current line), or '<,'> for the current visual selection.

Two of the most common flags are g, to enable replacing more than one occurrence of pattern per line, and i, to make the current search case-insensitive.

Let say we have a text like this

---

Roses are roses, violets are blue.

There's a rose on the rose number 5.

---

If we want to search & replace the word "rose", the syntax would be something like this

---

:%s/\<rose/flower/gi

---

ZQ  quit vim without saving

:set all  options that we can set

:sets  activated options

Resulting in

---

flowers are flowers, violets are blue.

There's a flower on the flower number 5.

---

Undo and redo

You can undo changes on a line if you stay on that line and pressing u

If there are many things you need to undo, you can exit without saving or writing again what you deleted.

If you want to redo what you did, just press Ctrl+r

#scheduletasks

Sometimes we can't run a task at certain time because we're busy or we don't have access to our desktop.

Thankfully, the command  at  is here to save us!

The atd daemon provides 26 queues, a to z, with jobs in alphabetically later queue getting less system priority

---

at <TIMESPEC> <command>

---

atq

---

Shows current queue

---

at -l

---

Same as atq

For long or typo-sensitive commands, it's often easier to use input redirection from a script file

---

at now +5min < myscript

---

For the TIMESPEC, we have many different combinations available

02:00pm

15:59

now +5min

teatime tomorow (teatime is 16:00)

noon +4 days

5pm august 3 2016

---

at now +5min -q b < myscript

---

Adds the task to the queue b

---

at -c [jobnumber]

---

You can inspect the actual commands that will run when the specified job is executed.

---

atrm [jobnumber]

---

Remove the specified job.

Scheduling recurring jobs with cron

It's easy to use at but we need to resubmit the job again.

cron allows to set up a schedule

In the early versions of cron, it was designed to abort the job if there was output that can't be shown. It would ended up sending an email notifying that the job was aborted.

Now, you have to redirect the output to wherever you specify or cron will email you the output.

Every user has a cron table

---

crontab -l

---

List the jobs for the current user.

---

crontab -r

---

Remove all jobs for the current users.

---

crontab -e

---

Edit jobs for the current user.

---

crontab <filename>

---

Remove all jobs and replace with the jobs read from the specified filename. If no filename specified, stdin will be used.

---

crontab -u [user]

---

Only root can use the -u option to manage the jobs for another user.

Job format

Minutes Hours Day-of-Month Month Day-of-Week Command

*  *  *  *  *  command

The first five fields uses the same syntax rules

*  "Don't care"/always

0-9  number to specify a number of minutes or hours, a date or a weekday (0 and 7 = Sunday, 1 = Monday, etc)

x-y  range starting on x and ending with y (both are included)

x,y  lists, can include ranges (5,10-13,17)

*/x  indicate an interval of x (*/7 in the minutes column will run a job exactly every seven minutes)

We can use three-letter English abbreviations for both month (Aug, Oct, Nov, Dec) and weekdays (Tue, Thu, Mon, Sun)

The last field is the command to be executed. If the command has an unescaped %, it will be treated as a new line and everything after it will be considered part of the stdin.

A corrupted crontab file could cause the system to become unstable.

0 9 2 2 * /usr/local/bin/yearly_backup

| | | | Any day of the week

| | | February

| | Day 2

09:00

*/7 9-16 * Jul 5 echo "Chime"

| | Any day of July but only Fridays

| Between 9:00 and 16:59

Repeat after 7 minutes

58 23 * * 1-5 /usr/local/bin/daily_report

| | | | From Monday to Friday

| | Every day of every month

At 23:58

0 9 * * 1-5 mutt -s "Checking in" boss@example.com % Hi there boss, just checking in.

| | | | From Monday to Friday

| | Every day of every month

At 09:00

Scheduling system cron jobs

System cron jobs aren't defined using the crontab command, but are instead configured in a set of configuration files.

The main difference in these configuration files is an extra field, located between the Day-of-Week field and the Command field, specifying which user a job should be run.

For more details, check  man 4 crontabs

System cron jobs are defined in two locations: /etc/crontab and /etc/cron.d/*

Packages that install cron jobs should do so by placing a file in /etc/cron.d/

There are also predefined jobs that run every hour, day, week and month. These jobs will execute all scripts places in:

/etc/cron.hourly

/etc/cron.daily

/etc/cron.weekly

/etc/cron.monthly

Any script inside of those folders must have the eXecutable permission activated.

The daily, weekly and monthly jobs are also executed using the run-parts command but from a different configuration file: /etc/anacrontab

Before RHEL 7, /etc/anacrontab was handled by a separate daemon (anacron) but now the file is parsed by the resular chrond daemon.

The syntax of /etc/anacrontab is different from the other cron configuration files. It contains exactly four fields per line:

Period in days  Once per how many days this job should be run

Delay in minutes  The amount of time the cron daemon should wait before starting this job.

Job identifier  This is the name of the file in /var/spool/anacron that will be used to check if this job has run. The timestamp is updated after every run

Command The command to be executed.

/etc/anacrontab also contains environment variable declarations using the syntax NAME=value.

Of special interest is START_HOURS_RANGE: jobs will not be started outside of this range.

Managing temporary files

In RHEL 7, directories for temporary files changed from earlier versions

/run contains runtime files from programs (volatile, only exists in memory)

/tmp, /var/tmp highly user-visible folders

When the system reboots or loses power, the volatile storage will be gone

systemd-tmpfiles replaced tmpwatch

It looks on directories and checks for timestamps, looking what should be deleted from the system.

When systemd starts, one of the first service units launched is systemd-tmpfiles-setup.

This service runs the command system-tmpfiles --create --remove

This command reads configuration files from /usr/lib/tmpfiles.d/*.conf, /run/tmpfiles.d/*conf and /etc/tmpfiles.d/*.conf

Any files and directories marked for deletion in those configuration files will be removed, and any files and directories marked for creation (or permission fixes) will be created with the correct permissions if necessary.

Regular cleaning

There's a systemd timer unit that calls systemd-tmpfiles --clean on a regular interval.

systemd timer units are a special type of systemd service that have a [Timer] block indicating how often the service with the same name should be started.

On RHEL 7, the configuration for the systemd-tmpfiles-clean.timer unit looks like this

---

[Timer]

OnBootSec=15min

OnUnitActiveSec=1d

---

This indicates that the service with the same name (systemd-tmpfiles-clean.service) will be started 15 minutes after systemd has started, and then once every 24 hours afterwards.

systemd-tmpfiles --clean parses the same configuration files as the systemd-tmpfiles --create but instead of creating files and directories, it will purge all the files which have not been accessed, changed or modified more recently than the maximum age defined.

Files on a Linux file system following the POSIX standard have three timestamps:

atime last time the file was accessed

mtime last time the file was modified

ctime last time the file status changed

Files will be deleted if ALL the timestamps are older than the maxium age defined on the systemd-tmpfiles configuration files.

---

stat <filename>

---

Shows information about the specified files (including the three timestamps).

systemd-tmpfiles configuration files

The format of the configuration files for the systemd-tmpfiles is detailed in man 5 tmpfiles.d

Type Path Mode UID GID Age Argument

| | | | | | |

| | | | | | Depending on the type, it will be written to the newly created file or used for a symbolic link

| | | | | Maximum age of the file

| | | | Group of the file

| | | Owner of the file

| | Permissions of the file/directory

| Path/to/the/file

Action that systemd-tmpfiles should take

f create a file if it doesn't exist yet, if the argument is given it will be written to the file

F create or truncate a file, if the argument is given it will be written to the file

d create directory if it doesn't exist yet

D create directory if it doesn't exist yet, empty the directory if it already exists

Z recursively restore SELinux contexts and file permissions and ownership

d /run/systemd/seats 0755 root root -

Create a directory called seats on the /run/systemd directory with the permissions rwx-r-xr-x that belongs to the user and group root. This directory won't be automatically purged.

D /home/student 0700 student student 1d

Create a directory for the user student with owner and group student, permissions rwx------, it will be deleted after 1 day.

L /run/fstablink - root root - /etc/fstab

Create a symbolic link /run/fstablink pointing to /etc/fstab, it won't be automatically purged.

Configuration file precedence

/usr/lib/tmpfiles.d/ anything that was provided by the relevant RPM packages, will be stored here. Sysadmins shouldn't edit them.

/run/tmpfiles.d/ files used by daemons to manage their own runtime temporary files.

/etc/tmpfiles.d/ meant for administrators to configure custom temporary locations and to override vendor-provided defaults.

If a file in /run/tmpfiles.d/ has the same file name as a file in /usr/lib/tmpfiles.d/, then the file in /run/tmpfiles.d/ will be used.

If a file in /etc/tmpfiles.d/ has the same file name as a file in either /run/tmpfiles.d/ or /usr/lib/tmpfiles.d/, then the file in /etc/tmpfiles.d will be used.

An administrator can easily override vendor-provided settings by copying the relevant file to /etc/tmpfiles.d/ and then editing it.

Quiz:

30 6 25 12 * /usr/local/bin/open_presents Early on Christmas morning

30 12 * * 3 reboot Every Wednesday at 12:30 p.m.

0 17 * * 4 rm -rf /home/student Every Thursday at 5:00 p.m.

echo reboot | at 12:30 wednesday Next Wednesday at 12:30 p.m.

3 0 1 * 1 /sbin/dump 0uf /dev/st0 /home Just after midnight on every Monday and every 1st of the month.

echo "userdel -r student" | at 17:00 thursday Next Thursday at 5:00 p.m.

#processespriorities

The purpose of priorities is to decide not so much how much time a process gets on the CPU but when there are more than one process waiting, who gets to go first.

If your system is saturated, the CPU is running at 100% of the time because you don't have enough CPU processing power.

You're getting CPU time because it's not running at 100%.

Priorities decides what process gets to go first over another process based on their relative importance (relative priorities).

The scheduler can be told to use different scheduling policies for different processes.

Processes running at userland (not Kernel ones) use a scheduling policy known as SCHED_OTHER (also called SCHED_NORMAL).

SCHED_NORMAL helps the administrator to set the relative priorities (nice processes).

The nice levels range from -20 to 19.

Using nice with a positive number, will make the process nicer. A negative value will move the process to the front of the line.

Higher nice levels indicate less priority, while lower nice levels indicate a higher priority.

The internal numbers used by the kernel are different.

Nice level -20-19 ... 0 ... 18 19

|-|-------|-------|--|

top [PR] RT -99 ... -3 -2 0 1 ... 20 ... 38 39

|---|---------|--|-|-|-------|-------|--|

top has another way to display the nice levels

You can only influence userland processes.

Most processes by default start with a priority of 20 (0).

If you nice something with 18, it will change it's priority to 38

Root (users with the CAP_SYS_NICE capability) has access to negative numbers. If you're a non-root user, you can only use positive nice numbers.

There are alternate scheduler policies and settings, control groups (cgroups), and more.

---

man 2 sched_setscheduler

---

For more information about schedulers.

Quiz:

High nice level These kinds of processes easily give up their CPU resources for others.

Negative nice level These kinds of processes attempt to keep CPU usage to themselves.

Regular users Cannot assign negative nice levels.

root Can renice processes belonging to other users.

-20 - +19 The complete range of nice levels.

Using nice and renice to influence process priority

Priorities changes automatically depending on how the system looks them.

---

ps axo pid,comm,nice --sort=-nice

---

The o option means options. You can choose which columns to show.

Processes that report a - as their nice level means that they're running with a different scheduling prolicy and will almost certainly be considered a higher priority by the scheduler. Other schedulers do not use nice to reorganize their relative priorieties, only SCHED_NORMAL does.

---

ps axo pid,comm,nice,cls

---

You can see which sheduling policy is being used for that process by adding "cls" to the options. A TS in that column means that the process is running under SCHED_NORMAL and can use nice levels, anything else is another scheduling policy.

---

nice -n [level of nice] [command]

---

nice -n 15 dogecoinminer &

---

Runs the command dogecoinminer, setting it's nice level to 15 and send it on the background immediately.

Alright, my process is already running and I don't want to kill it.

Well, we can use the `renice` command

---

renice -n [level of nice] [PID]

---

Changes the nice level of the already running process by specifying it's PID.

Remember, you can use `pgrep` to get it's PID.

---

renice -n -7 $(pgrep origami@home)

---

The `top` command can also be used to (interactively) change the nice level on a process. From within top, press r, followed by the PID to be changed and the new nice level.

#acls

Standard Linux file permissions are fine when you have only one owner, an owner group and the rest of the world.

What if you want to have specific permissions for another individual user or another group but you don't want to set those permissions on the other column?

ACLs allow fine-grained permissions to be allocated to a file.

Using ACLs we can also set default permissions and attach it to a directory.

In order to set ACLs, we need to be able to store them into the file system.

There's a mount point option that allows the use of ACLs. Your file system must have the ACL option turned on.

Otherwise, the  setfacl  command will fail.

Ok, ok, now I'm curious and wanting to see some ACLs on my system. How do I know if a file has ACLs?

---

ls -l [file]

---

Wait, it's just a normal ls command....Nope!

If a file has ACL, you'll see a + sign next to the permissions column

-rwxrw----+ 1 owner group...

The + indicates that there are ACL settings associated with this file.

Changing group permissions on a file with an ACL by using chmod doesn't change the group-owner-permissions, but does change the ACL mask.

---

setfacl -m g::perms <filename>

---

Update file's group-owner permissions.

---

getfacl <filename>

---

Display ACL settings of the specified file.

This command will still works even if the file doesn't have any ACL settings.

---

$ getfacl <filename>

# file: <filename>

# owner: foo

# group: bar

user::rwx // default owner permissions

user:james:--- // james won't have any permissions because it's been explicitly indicated

user:1005:rwx #effective:rw- // the user with ID 1005 will be able to use read and write on that file

group::rwx #effective:rw- // even if it's the group owner, the mask wins.

group:sodor:r-- // only read permission for the group sodor

group:2210:rwx #effective:rw- // group 2210 will be able to read and write, not execute. Check the mask.

mask::rw- // the max permissions for everybody. Doesn't matter if you have rwx, you'll be able to use the ones specified on mask.

other::--- // just like the default permissions system, if you don't match an specified user/group, this permissions will apply.

---

By default, the mask doesn't applies for the owner of the file.

If you modify the owner's group permissions, the mask will change too.

Well, all those details were for a FILE, what about the ACL settings for a DIRECTORY?

---

$ getfacl <directory>

# file: <directory>

# owner: foo

# group: bar

# flags: -s-  // this means it has an special bit. In this case, the setgid bit.

user::rwx

user:james:---

user:1005:rwx

group::rwx

group:sodor:r-x

group:2210:rwx

mask::rwx

other::---

default:user::rwx  // default file owner ACL permissions. The file owner will get rw on new files and x on new subdirectories

default:user:james:---  // yeah, we don't want James over here

default:group::rwx  // whatever group owns the file, that group will have rwx permissions

default:group:sodor:r-x  // the group sodor will only have r-x permissions

default:mask::rwx

default:other::---

---

The default permissions only apply for newly created files inside that directory. It won't affect already created files.

If you look again at the lines for the user 1005 and the group 2210, they weren't included on the default entries. That means they won't get initial ACL entries added for them

automatically to any new files or new subdirectories. This will effectively limits them to files and subdirectories that they already have ACLs on. They can still create their own files and subdirectories.

The output from getfacl can be used as input to setfacl.

---

getfacl -R /

---

Generates output for the specified directory and it's content.

---

setfacl --set-file=<filename>

---

This will do a massive update using the ACL from the specified file.

ACL permission precedence

- If the process is running as the user that owns the file, then the file's user ACL permissions apply.

- If the process is running as a user that is listed in a named user ACL entry, then the named user ACL permissions apply (as long as it's permitted by the mask).

- If the process is running as a group that matches the group-owner of the file, or as a group with an explicit named group ACL entry, then the matching ACL permissions apply (as long as it's permitted by the mask).

- Otherwise, the file's other ACL permissions apply.

Quiz:

getfacl /directory Display ACLs on a directory.

user:mary:rx file Named user with read, execute permissions for a file.

user::rx file File owner with read, execute permissions for a file.

g::rw /directory Read, write permissions for a directory granted to the directory group-owner.

g::rw file Read, write permissions for a file granted to the file group-owner.

group:hug:rwx /directory Read, write, execute permissions for a directory granted to a named group.

default:m::rx /directory Read, execute permissions set as the default mask.

default:user:mary:rx /directory Named user granted initial read permission for new files and read, execute permissions for new subdirectories.

Changing ACL file permissions

You can use the setfacl command to change or set permissions for existent files or default values for new files and directories.

It uses the same letters than the default permissions system.

---

setfacl [-s|--set|--setfile]

---

Options to completely replace the ACL settings on a file.

If you're going to set ACL on a file, you must specify four permissions: owner, group, other and mask.

---

setfacl -m u:[name]:rX <filename>

---

Add or modify ACL without replacing the existing ones.

If we don't specify the name of the user or the name of the group, it will set those permissions to the owner/owner-group

X = conditional execute :thinking:

Not sure how many weeks later but now I can add that the capital X means "add execute permission for directories only" OR already has execute permission for some user.  man 1 setfacl (line 150).

---

setfacl -m o::- <filename>

---

Sets the permissions for other in ---

Multiple entries can be set set with the same command, using comma to separate each one.

---

setfacl -m u::rwx,g:sodor:rX,o::- <filename>

---

rwx for owner, rX (conditional execute) for the group sodor and nothing for other.

getfacl as input? Yep, just piping.

---

getfacl <filename> | setfacl --set-file=- <filename>

---

Gets the ACL from the specified file and apply them to the second file.

---

setfacl -m m::r <filename>

---

Defines the mask for the file specified.

The ACL mask is recalculated each time one of the impacted ACL settings (named users, group-owner, or named groups) is modified or deleted.

To avoid mask recalculation, use -n or include a mask setting with any setfacl operation that modifies mask-affected ACL settings.

---

setfacl -x u:[name],g:[name] <filename>

---

Delete the specified user and group configuration.

The mask won't be deleted if there are any remaining ACL settings for other users.

What's different with a directory?

Just add 'd' to the beginning of the syntax to set DEFAULT values.

---

setfacl -m d:u:[name]:rx /directory

---

This adds a default named user with read-only permission and execute permission on subdirectories.

---

setfacl -x d:u:[name] /directory

---

Same as deleting an ACL on a file, just add d:

And...if we want to remove all DEFAULT settings on a directory

---

setfacl -k /directory

---

And...if we want to remove ALL ACLs on a directory...

---

setfacl -b /directory

---

#selinux

Security Enhanced Linux (SELinux) is an additional layer of system security.

A primary goal of SELinux is to protect user data from system services that have been compromised.

The standard user/group/other permission security model is known as discretionary access control.

SELinux provides an additional later of security that is object-based and controlled by more sophisticated rules, known as mandatory access control.

Every process goes through the SELinux vector table to look up what has been said about, what is allowed, how files are going to be use.

With SELinux, when a process tries to do something, SELinux will alert about it and allow or deny the process to do that action.

Every single file in the system has a tag or context assigned.

SELinux labels have several contexts: user, role, type, and sensitivity.

The targeted policy, which is the default policy enabled in RHEL, bases it's rules on the third context: the type context.

Type context names usually end with _t . The type context for the web server is httpd_t .

The type context for files and directories normally found in /var/www/html is httpd_sys_content_t .

The type context for files and directories normally found in /tmp and /var/tmp is tmp_t .

The type context for web server ports is http_port_t .

Basically, there's a policy rule that permits Apache (the web server process running as httpd_t ) to access files and directories with a context of httpd_sys_content_t

By default, everything on Linux is denied. These policies allows processes action's.

SELinux has rules for remote files such as NFS and CIFS, although all files on these file systems are labeled with the same context.

Many commands that deal with files have an option (usually -Z) to display or set SELinux contents.

---

ps axZ

---

Show processes with their SELinux label.

---

ls -Z </directory>

---

Shows the SELinux context for the content of that directory.

For troubleshooting purposes, SELinux protection can be temporarily disabled using SELinux modes.

Enforcing mode:

SELinux denies access to anything that doesn't have an explicit policy to allow a behavior.

Permissive mode:

Often used to troubleshoot issues. SELinux allows all interactions, even if there is no explicit rule, and it logs those interactions it would have denied in enforcing mode.

Disabled mode:

Turns off SELinux. A system reboot is required to disable SELinux entirely, or to get from disabled mode to enforcing mode or permissive mode.

It's better to use permissive mode than to turn off SELinux entirely. The kernel will automatically maintain SELinux file system labels as needed, avoiding the need for an expensive relabeling of the file system when the system is rebooted with SELinux enabled.

---

getenforce

---

Shows the current SELinux mode.

---

setenforce [Enforcing|Permissive|1|0]

---

Changes the SELinux mode.

SELinux Booleans

SELinux Booleans are switches that change the behavior of the SELinux policy. SELinux Booleans are rules that can be enabled or disabled.

They can be used by security administrators to tune the policy to make selective adjustments.

---

getsebool -a

---

Shows all the current Booleans and their values.

Quiz:

| | |
|---|---|
| Enforcing mode | Policy rules are obeyed and violations logged |
| Context | Label on processes, files, and ports that determine access |
| Disabled mode | A reboot is required to transition to this mode |

Boolean Switch that enables/disables a set of policy rules

Permissive mode Policy rule violations only produce log messages

Changing SELinux modes

We have two ways for it.

---

getenforce

---

This will tell us what's the current SELinux mode.

---

setenforce [Enforcing|Permissive|1|0]

---

Sets the current SELinux mode.

Alright, what if we want to change the SELinux mode at boot time?

Well, go to the /etc/selinux/config file and change the line

SELINUX=enforcing

to any other mode.

You can also change the SELinux type (targeted, minimum, mls)

Changing SELinux contexts

It's stored in the file system in a matter similar to permissions. There's a database where every file has a label.

When we do a relabel, we're gonna check every file in the system to make them match the label that's stored on the database.

We can change contexts with the command chcon but it's not the recommended way because the files won't survive a relabel.

restorecon will look for the context on the database and restore it to that file.

---

chcon -t [context] <filename>

---

Change the context of the specified file

Alright, I understand. chcon shouldn't be used for changing contexts because of the relabel problem, bla bla bla...but how do I deal with it then?

Use the semanage command.

Note: if you don't have the semanage command, you need to install policycoreutils-python

---

semanage fcontext -l

---

Shows all the contexts on the database. This command also supports RegEx.

---

restorecon -Rv </path/to/directory>

---

Restores the context of all the files inside of the specified directory (R is for recursive, v is for verbose)

---

semanage fcontext -a -t httpd_sys_content_t '/virtual(/.*)?'

---

Adds a new rule on the SELinux database. From now, every time we restore the context, it will set httpd_sys_content_t to the files inside /virtual

Changing SELinux Booleans

While there a lot of tags and Booleans, you only need to work with them on context of each service.

What does it means? If you're working with a web server, you'll only work with the Booleans and tags related to that service.

If we install the package  selinux-policy-devel  , we'll get many man pages.

There's a chance that those pages won't be available by default, so you need to execute the next command

---

sepolicy manpage -a -p /usr/local/man/man8

---

This will create the man pages related to SELinux.

With our new SELinux man pages, we can see the available Booleans for that service and more details even!

---

man httpd_selinux

---

Booleans are switches, they can enable or disable rules.

---

getsebool [-a] [Boolean name]

---

Shows the status of the specified Boolean. If you don't specify one and use the option -a, it will show all the available booleans.

---

setsebool [-P] [Boolean name] [on|off]

---

Toggles the Boolean. Using the -P option makes the change persistent.

---

semanage boolean -l [| grep {name of Boolean}]

---

Shows all the Booleans with their current state, default value and description. I suggest you to use grep to filter the one that you want.

---

semanage boolean -l -C

---

Shows all the Booleans that someone changed it's value.

Troubleshooting SELinux

By default, SELinux is a deny-all system.

There's a policy for each thing, certains permissions for each program.

Before thinking of making any adjustments, SELinux may be doing it's job of deny any unspecified access to files.

The most common SELinux issue is an incorrect file context. This can occur when a file is created in a location with one file context and moved into a place where a different context is expected. In most cases, the  restorecon  command will correct the issue.

Another remedy for a too-restrictive access could be the adjustment of a Boolean. Adjusting Booleans requires more care because they can have a broad impact on system security.

It's possible that the SELinux policy has a bug that prevents a legitimate access. When it's clear that a policy bug has been identified, contact Red Hat support.

Alright, what if we want to check SELinux messages?

We need to install the  setroubleshoot-server  package for it.

SELinux messages from /var/log/audit/audit.log will be sent to /var/log/messages.troubleshoot-server and this last one will send a short summary to /var/log/messages

Each summary includes an UUID for SELinux violations that can be used to gather further information.

---

sealert -l [UUID]

---

See more information about the SELinux violation.

---

sealert -a /var/log/audit/audit.log

---

Search and display SELinux messages in the audit.log file.

scontext is the source of the problem. tcontext is the target that the service was trying to do something to.

We can generate a local policy module by doing

---

grep [service] /var/log/audit/audit.log | audit2allow -M mypol

---

This will generate the policy

---

semodule -i mypol.pp

---

This will enable the policy created.

#networkusers

The purpose of it is to have a centralized server to manage identities.

Having a database with all the users of the network allow us to check this database from all the machines on a realm or domain and use those same shared accounts.

It's also helpful for allowing Single Sign-On (SSO). With SSO, a user authenticates once using a password (or other means), and then obtaions a form of ticket or cookie that can be used to automatically authenticate to other services.

A centralized identity management system will need to provide at least two services:

Account information: information like username, home directory location, UID and GID, group memberships, etc.

Popular solutions include LDAP (Lightweight Directory Access Protocol), used in multiple products such as Active Directory and IPA Server, and Network Information Services (NIS).

Authentication information: how the system validates that the user is the person that claims to be. This can be done by providing a cryptographic password hash to the client system, or by sending the (encrypted) password to the server, and receiving a response. An LDAP server can provide authentication information in addition to account information.

On RHEL 7, local user information is provided by /etc/passwd while authentication information (in the form of a hashed password) is provided by /etc/shadow.

In addition to the LDAP, we also use Kerberos.

Kerberos is an identify system that allow us to set private/public key sets in order to validate users instead of passwords.

It also can be used to register not only users but also services.

For attaching to central LDAP and Kerberos servers, the following files, at a minimum, would need to be updated:

/etc/ldap.conf information about the central LDAP server and its settings

/etc/krb5.conf information about the central Kerberos infrastructure

/etc/sssd/sssd.conf configure the system security services daemon (sssd), responsiblefor retrieving and caching user information and auth info

/etc/nsswitch.conf indicate to the system which user information and authentication services should be used

/etc/pam.d/* configuring how authentication should be handled for various services

/etc/openldap/cacerts store the root certificate authorities (CA) that can validate the SSL certificated used to identify LDAP servers

IPA allows to install LDAP and Kerberos using one script.

To understand all those configuration files, you may have to learn PAM, Kerberos, etc. It's complicated and you may learn it with practice.

It's also easy to make mistakes editing manually those files.

RHEL 7 comes with a suite of tools to automate these configurations: authconfig

authconfig consists of three related tols that can perform the sames actions:

authconfig command-line tool to automate configurations across a number of systems. Commands tends to be very long, with multiple options passed in

authconfig-tui interactive version of authconfig. Uses a menu-driven text interface. Can be used over ssh

authconfig-gtk launches a graphical interface. It can also be launched as system-config-authentication. Installed using the authconfig-gtk package

Necessary LDAP parameters

To connect to a central LDAP server for user information, authconfig needs a number of settings:

- The host name of the LDAP server(s)

- The base DN (Distinguished Name) of the part of the LDAP tree where the system should look for users. dc=example, dc=com, ou=People, o=Ponycorp

- If SSL/TLS is used to encrypt communications with the LDAP server, a root CA certificate that can validate the certificates is offered by the LDAP server

A system will also need some extra packages installed to provide LDAP client functionality. Installing sssd will provide all the necessary dependencies.

Necessary Kerberos parameters

- The name of the Kerberos realm to use. A kerberos realm is a domain of machines that use a common set of Kerberos servers and users for authentication

- One or more key distribution centers (KDC). This is the host name of your Kerberos server(s).

- The host name of one or more admin servers. This is the machine that clients will talk to when they want to change their passwords, or perform other user modifications. Typically, it's the same as the primary KDC but it can be a different machine.

In addition, an administrator can specify if DNS should be used to look up the realm to use for a specific host name, and to atuomatically find the KDCs and admin servers. krb5-workstation can be installed to help debug Kerberos issues, and to work with Kerberos tickets from the command line.

To test the LDAP + Kerberos configuration, an administrator can simply attempt to log into the system (over sssh) using the credentials of one of the network users. In addition, the getent command can be used to retrieve information about a network user with the next command

---

getent passwd <username>

---

Attaching a system to an IPA Server

Red Hat provides an integrated solution for configuring LDAP and Kerberos: IPA (Identity, Policy, and Auditing) Server.

IPA Server can centralize sudo rules, SSH public keys, SSH host keys, TLS certificates, automounter maps and much more.

ipa-client-install will retrieve almost all the necessary information from DNS or ask for missing information.

In addition to installing the client, it also creates new entries on the server for it's host.

Joining a system to Active Directory

RHEL 7 features multiple methods of joining a system to Active Directory.

Administrators can choose to install the samba-winbind package and configure winbind through the authconfig family of tools, or administrators can install both sssd and realmd packages and use sssd and the realm command.

realm can be used to join Kerberos realms, or IPA server domains.

First we install the package realmd

---

sudo realm discover domain.example.com

---

We discover the settings from the specified domain.

Once we discovered the domain, we need to join the Active Directory.

---

sudo realm join domain.example.com

---

This will install all necessary packages and configure sssd, pam, /etc/nsswitch.conf, etc.

Also, this will attempt to join the local system to Active Directory using the Administrator account; enter the password for this account when prompted.

To use a different account, use the --user argument.

---

sudo realm permit --realm domain.example.com --all

---

Active Directory accounts are now usable but logins using AD are still disabled. This command will enable logins.

---

sudo realm permit --realm domain.example.com DOMAIN\\Itchy DOMAIN\\Scratchy

---

Allows only certain users to log in.

#partitions

The purpose of partitions is to divide the hard drive to perform different functions.

- Limit available space to applications or users.
- Allow multibooting of different operating systems from the same disk.
- Separate operating system and program files from user files.

- Create separate area for OS virtual memory swapping.

- Limit disk space usage to improve performance of diagnostic tools and backup imaging.

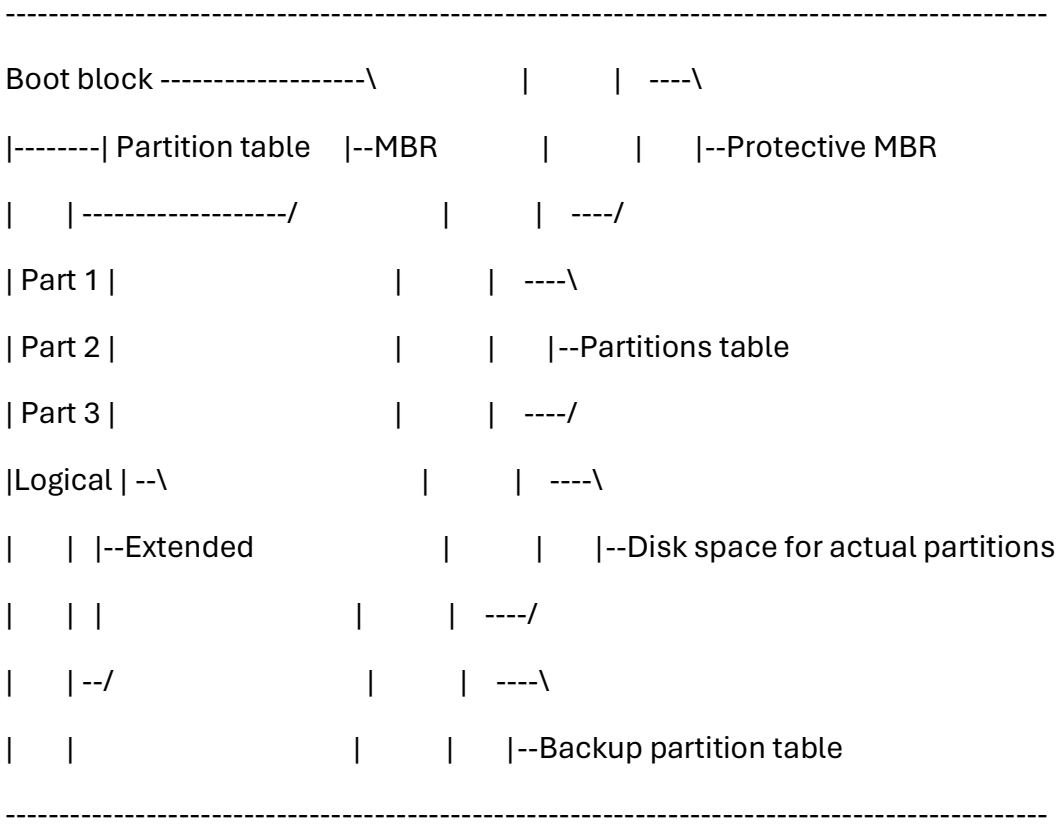
A single disk can store a number of partitions depending on the scheme.

MBR (Master Boot Record)

4 partitions (maximum, 15 by using extended and logical partitions). Partition size of 2 TiB.

GPT (GUID Partition Table)

For systems running UEFI, GPT is the standard for laying out partition tables. Support 128 partitions. Partition size of 8 ZiB.




While using MBR, the first part of the scheme is the boot block and the partition table.

You have to make sure that the partition number 4 is extended. You won't use the extended partition, the purpose is just to create logical partitions inside of it.

In order to keep it compatible with older systems, GPT has a MBR on the first block. Then, the partition table begins.

At the end of the disk, there's another backup of the partition table.

It's very important to use the right tools to clean a disk, otherwise the backup will be used and we'll get notified that the disk is being used.

Creating MBR disk partitions

We use fdisk to create MBR partitions.

---

fdisk [/dev/vdb]

---

Open fdisk, we can specify a disk to work on.

Before you make any changes. Keep in mind that they won't be automatically applied unless you use the w command to write them.

You can exit fdisk at any time and prevent from any changes to be made.

Once fdisk is open, we can press m to get help.

We press n to create a new partition

p for a primary partition

e for an extended partition

It will ask us the number of the partition to create (default value is the lower unused partition number).

Then, we have to set the first sector of the partition. In the old days, the first sector was 63. Nowadays, it's better to use 2048 (leaving 1 MiB gap).

And the last sector of the partition. We can define the last sector as +[size][K,M,G] (+10G) or just +[sectors].

After that, we need to set the partition's type. Press `t` for it.

fdisk will ask us for the hex code for the new partition type. You can see the table of partitions by pressing `L`.

Setting the partition type correctly is crucial, since some tools rely on it to function properly.

Once we're done with this, we can write the changes to the disk.

No changes were made before and won't be made unless you press `w` to write them and exit fdisk.

We need to initiate a kernel re-read of the new partition table, so we use the command `partprobe` with the disk device name as an argument.

---

```
partprobe /dev/[vdb|whatever]
```

---

partprobe won't work on partitions that are already running in RHEL 6.

Removing MBR partitions

Open `fdisk` with the device that we want to work on.

Identify the partition number of the partition to delete by entering the command p (it will print the partition table).

Once we have the number of the partition that we want to delete, use the command d to delete it.

It will prompt for the partition number and then delete it.

Changes won't be saved unless you use the w command to write them.

After that, remember to run partprobe to make the kernel recognize the changes made.

Creating GPT disk partitions

We used fdisk for MBR, now we use gdisk for GPT

While GPT support has been added to fdisk, it's still experimental and gdisk should be used instead.

---

gdisk /dev/[sda1|whatever]

---

Same as fdisk, just choose the disk to work with.

n to create a new partition.

You can choose a number between 1 and 128 for the new partition.

Again, you can choose the starting sector of the partition and then the last sector of it. A value of -512M indicates an ending partition position that is 512 MiB BEFORE the end of the contiguous available sectors (expand the partition across all the available sectors BUT leave 512 MiB at the end of the disk).

We set the type of the partition. Unlike fdisk, we need to specify 4 digits to set the type of the partition. L to see the codes.

Once you press w to write the changes, gdisk will ask you if you want to proceed (how nice).

Aaaaaand...don't forget partprobe after doing changes.

Removing GPT partitions

Open gdisk specifying the disk, press p to see the partitions, press d to initiate a partition removal, select the partition and then w to write the changes.

Creating file systems

After a block device has been created, the next step is applying a file system format to it.

A file system applies a structure to the block device so that data can be stored and retrieved from it.

RHEL suports many different system types but two common ones are xfs and ext4.xfs is used by default in Anaconda.

mkfs can be used to apply a file system to a block device. If no type is specified, an extended type two (ext2) will be used.

To specify the file system type, a -t should be used

Once the file system has been applied, we need to mount the partition.

---

mount /dev/[whatever] [/folder/to/mount]

---

This will mount the partition at the specified folder.

---

mount

---

Using the command without arguments will display any mounted file system, with their mount points and options.

Manually mounting a file system is an excellent way to verify that a formatted device is accessible or working in the way desired.

Mounted file systems with the command mount won't survive a reboot unless it's added to the /etc/fstab file.

Persistently mounting file systems

By adding a listing for a device into the /etc/fstab file, administrators can configure a device to be mounted to a mount point at system boot.

/etc/fstab is a white-space delimited file with six fields per line.

/dev/mapper/centos-root / xfs dfaults 0 0

UUID=700453e-4976-a26c-358f4377175 /boot xfs defaults 0 0

Device to be used | | | Dump flag and fsck order

| | Options applied to the device when mounted. More options at man page of mount

| File system type that has been applied to the block device

Mount point where the device should be attached into the directory hierarchy. Mount point should already exists.

It's better to use UUID instead of /dev/sda, /dev/sdb because the UUID is always the same, unlike the relative name that's based on disk discovery at boot time.

The creation of the UUID happens when you make a file system on the partition or when you make it a swap area of it.

The dump flag is used with the dump command to make a backup of the contents of the device.

The fsck order field determines if the fsck should be run at boot time, in the event that the file system wasn't unmounted cleanly.

The value of the fsck order indicates the order in which file systems should have fsck run on them if multiple file systems are required to be checked.

||||||||||||||||||||||||||||||||||

|||||| I M P O R T A N T ||||||

||||||||||||||||||||||||||||||||||

Having an incorrect entry in /etc/fstab may render the machine unbootable.

To avoid that situation, an administrator should verify that the entry is valid by unmounting the new file system and using mount -a , which reads /etc/fstab to mount the file system back into place. If the mount -a command returns an error, it should be corrected before rebooting the machine.

Managing Swap space

Swap space is an area of disk that is used as an extension of memory (RAM).

It's used when we run out of memory. Objects from the RAM are moved to the swap memory in order to free some space.

Using fdisk or gdisk , create the partition with the size that you want for the swap.

It should start right next to last partition.

The type of the partition should be 82 Linux Swap.

After saving the changes, we won't do a mkfs (that only creates a file system), we'll use the command  mkswap  to mark that partition as swap.

---

mkswap /dev/[sdb2|sda4|whatever]

---

Now we need to activate the swap partition.

---

swapon -a

---

This will activate all the partitions marked as swap space.

---

swapon /dev/[sdb1|sda6|whatever]

---

Activates the specified swap partition.

---

free -m

---

Display information about the memory usage (in -m MiB)

If needed, an administrator can deactivate a swap space using the  swapoff  command.

A  swapoff  will only be successful if any swapped data can be written to other active swap spaces or back into memory.

If data cannot be written to other places, the  swapoff  will fail, with an error, and the swap space will stay active.

It's likely that a swap space will be required to automatically start every time the machine boots. We need to add it to the /etc/fstab file.

```
/-----------------------------------------------------------------\
| UUID=here-comes-the-UUID      swap   swap   defaults   0 0 |
\-----------------------------------------------------------------/
```

Swap spaces require neither backing up nor file system checking.

If we want to set a priority for the swap partition, instead of defaults we have to use the option pri=[number]

#lvmstorage

Logical volume management (LVM) concepts

Just take a lot of disks. Create a partition on each one (or maybe multiple partitions on one).

Use the command pvcreate to mark the partition as a physical volume.

Then, add those physical volumes to the volume group.

Can you use a partition on a disk that contains others partitions and add it to the volume group?

Yes. It's not recommended due to performance.

This volume group will give us a flat amout of space to do whatever we want.

Logical volumes are like partitions on volume groups.

The advantage of using logical volumes is that we can resize them dynamically depending on what our needs are.

We also can use blocks of space that aren't contiguous to our logical volume and mark them as part of it.

Physical devices: storage devices used to persist data stored in a logical volume. Block devices (disk partitions, whole disks, RAID arrays, SAN disks).

Physical volumes (PV): used to register underlying physical devices for use in volume groups. LVM automatically segments PVs into physical extents (PE); these small chunks of data that act as the smallest storage block on a PV.

Volume groups (VG): storage pools made up of one or more physical volumes. A PV can be allocated to a single VG.

Logical volumes (LV): created from free physical extents in a volume group and provide "storage" device used by applications, users, and the operating system. LVs are a collection of logical extents (LE), which map to physical extents, the smallest storage chunk of a PV. Each LE will map to one PE. Setting specific LV options will change this mapping; for example, mirroring causes each LE to map to two PEs.

Quiz:

| | |
|---|---|
| Logical Volume (LV) | Formatted with a file system and mounted for use at runtime |
| Physical volume (PV) | Maps to a physical storage device, such as a disk or partition |
| Logical extent | Storage chunk of a LV, typically maps to a PE |
| Volume group (VG) | Used to identify a pool of PVs for use in creating one or more LVs |

Physical extent Name used for the storage chunk of a PV, also the smallest storage chunk for a LV

Disk, partition, RAID array Potential candidates for use as a single PV

Implementing LVM storage

Creating a logical volume

- Create a new partition for use with LVM. Always set the partition type to Linux LVM on LVM partitions. 0x8e for MBR-style partitions.

It's important to set the partition type, otherwise some tools may not work properly.

- Create a physical volume using pvcreate to label the partition for use with LVM as a physical volume.

A PV is divided into physical extents (PE) of a fixed size (like 4 MiB blocks). Label multiple devices at the same time by using space-delimited names as arguments to pvcreate

---

pvcreate /dev/vda2 /dev/vdb1

---

This will label /dev/vda2 and /dev/vdb1 as PVs, ready for allocation into a volume group.

A PV only needs to be created if there are no PVs free to create or extend a VG.

Check PVs with pvscan/pvdisplay

- Create a volume group using vgcreate, used to create a pool of one or more physical volumes. The size of the VG is determined by the total number of physical extents in the pool. A VG is responsible for hosting one or more logical volumes by allocating free PEs to a LV; therefore, it must have sufficient free PEs available at the time the LV is created.

---

vgcreate vg-alpha /dev/vda2 /dev/vdb1

---

This will create a VG called vg-alpha that is the combined size, in PE units, of the two PVs /dev/vda2 and /dev/vdb1.

A volume group only needs to be created when there is none in existence. Additional VGs may be created for administrative reasons to manage the use of PVs and LVs. Otherwise, existing VGs can be extended to accommodate new LVs when needed.

- Create a logical volume using lvcreate. Sounds very self explanatory.

---

lvcreate -n [name of the Logical Volume] -L [size] [vg-name]

---

lvcreate -n hercules -L 2G vg-alpha

---

This will create a logical volume called hercules and will have 2 GiB from vg-alpha.

There must be sufficient free physical extents to allocate 2 GiB, and if necessary, it will be rounded to a factor of the PE unit size.

The -L option expects sizes in bytes. The -l option expects sizes measured as a number of physical extents.

---

lvcreate -L 128M

---

Size the logical volume to exactly 128 MiB

---

lvcreate -l 128

---

Size the logical volume to exactly 128 extents in size. The total number of bytes depends on the size of the physical extent block on the underlying physical volume.

Once a logical volume is created, there will be a device file on /dev/[vg name]/[lv-name] (example: /dev/vg-alpha/hercules).

The device mapper program (in charge of creating mapping names to constructions for different types of logically built architectures), it also creates another naming convention which starts with /dev/mapper and that name is combined with dash (-) to the logical volume name (/dev/mapper/vg--alpha-hercules).

It's the same device. You can use either one, both will always exists.

- Add the file system to the logical volume using mkfs.

---

mkfs -t xfs /dev/vg-alpha/hercules

---

Make the file system available across reboots by creating a mount point and adding it to /etc/fstab

We can use the /dev/[file] name instead of the UUID because, even if the PVs that are part of the VG change their names, both the VG and the LV keep their names.

Removing a logical volume

- Prepare the file system by moving all the data that must be kept to another file system and unmounting the file system of the LV. Remember to remove any entries related to it on /etc/fstab

Removing a logical volume will destroy any data stored on the logical volume.

- Remove the logical volume using the command lvremove and specifying the device file.

---

lvremove /dev/vg-alpha/hercules

---

The file system must be unmounted before running this command. It will ask for confirmation before removing the LV. The PE will be freed and made available for assignment to existing or new LVs in the volume group.

- Remove the volume group with vgremove, using the name as argument.

---

vgremove vg-alpha

---

The VG's physical volumes will be freed and made available for assignment to existing or new VGs on the system.

- Remove the physical volumes helped by the pvremove command. The PV metadata is wiped from the partition (or disk). The partition is now free for reallocation or reformatting.

---

pvremove /dev/vda2 /dev/vdb1

---

Reviewing LVM status information

Physical volumes

---

pvdisplay

---

Displays information about physical volumes. If no argument is provided, it will list all the PVs on the system.

--- Physical volume ---

PV Name /dev/sda2 Device name

VG Name centos VG where the PV is allocated

PV Size 4,00 GiB / not usable 0 physical size of the PV, including unusable space

Allocatable yes

PE Size 4,00 MiB physical extent size, the smallest size a logical volume can be allocated

Total PE 1024

Free PE 1 how many PE units are available for allocation to new logical volumes

Allocated PE 1023

PV UUID fklWRE-Vl0N-OlVf-UpBU-KzTg-NXoY-00hWqX

Volume groups

---

vgdisplay

---

Displays information about volume groups. If no argument is provided, it will list all the VGs on the system.

--- Volume group ---

VG Name centos name of the volume group

System ID

Format lvm2

Metadata Areas 1

Metadata Sequence No 3

VG Access read/write

VG Status resizable

MAX LV 0

Cur LV 2

Open LV 2

Max PV 0

Cur PV 1

Act PV 1

VG Size 4,00 GiB total size of the storage pool available for logical volume allocation

PE Size 4,00 MiB

Total PE 1024 total size expressed in PE units

Alloc PE / Size 1023 / <4,00 GiB

Free PE / Size 1 / 4,00 MiB how much space is free in the VG for allocating to new LVs or to extend existing LVs

VG UUID XRBow9-I7Tl-BJWo-o5LY-AP44-b0UP-nrVGJ6

Logical volumes

---

lvdisplay

---

Displays information about logical volumes. If no argument is provided, it will list all the LVs on the system.

--- Logical volume ---

LV Path /dev/centos/root device name of this LV (some tools may report the device name as the mapper)

LV Name root VG the LV is allocated from

VG Name centos

LV UUID sk98x5-K1v7-r5RP-J7J4-iWEf-nSTi-iMqcMa

LV Write Access read/write

LV Creation host, time localhost.testing, 2019-08-19 20:53:02 -0300

LV Status available

# open 1

LV Size <3,50 GiB total size of the LV (use file system tools to check free space and used space)

Current LE 895 number of LE used by this LV

Segments 1

Allocation inherit

Read ahead sectors auto

- currently set to 8192

Block device 253:0

Extending Logical Volumes

A volume group can be extended and can be reduced.

We have a PV and we want to add it to the VG.

---

vgextend [volume group] [PV]

---

- Prepare the PV. You know, make a partition with LVM type.

- Create the physical volume (pvcreate).

- Extend the volume group using the vgcreate command.

- Verify the new space is available with the help of vgdisplay to confirm the additional physical extents are available. Check the Free PE / Size in the output. Shouldn't be zero.

Reducing a volume group

When we're reducing, first we reduce the file system, then the logical volume.

- Move the physical extents using pvmove to move the data out of the PV that you want to remove.

---

pvmove /dev/vdb2

---

Moves the PE away from /dev/vdb2. Only works if there's enough free extents in the VG and if all of those come from other PVs.

It's recommended to back up data stored on all logical volumes in the volume group. An unexpected power loss during the operation may leave the volume group in an inconsistent state. This could cause a loss of data on logical volumes in the volume group.

- Reduce the volume group with the vgreduce command, used to remove the PV from the VG.

---

vgreduce vg-alpha /dev/vdb2

---

The /dev/vdb2 PV is now removed from the vg-alpha VG and can be used on another VG. pvremove  will make it stop from being used as a PV.

Extend a logical volume and XFS file system

- Verify the volume group has space available,  vgdisplay  will be useful for it. Then expand the LV.

---

vgdisplay [VG]

---

lvextend -L +300M /dev/vg-alpha/hercules

---

Adds 300 MiB to the current size of the LV.

---

lvextend -l +50%FREE /dev/vg-alpha/hercules

---

Add 50% of the current free space in the VG to the LV.

- Extend the file system. If the file system is XFS, we'll use the command xfs_growfs

---

xfs_growfs [mount point]

---

Extends the XFS file system.

It's normal to forget to run xfs_growfs after doing a lvextend. You can add the -r option with lvextend, it will automatically resize the file system after the LV is extended.

Extend a logical volume and ext4 file system

- Verify the volume group has space available, same as before, with vgdisplay

---

vgdisplay [VG]

---

- Extend the logical volume

---

lvextend -l +[extents] /dev/VG/LV

---

- Extend the file system, for an ext4 file system, we use the resize2fs command

---

resize2fs /dev/VG/LV

---

The different with xfs_growfs is that the first one uses the mount point of the file system. resize2fs uses the logical volume name.

#nfsnetworkfilesystem

The Network File System has been around as a way to share files from one system to another.

You set up a NFS server on one machine and you connect to it with a NFS client.

NFS is an open standard under active extension which supports native Linux permissions and file system features.

RHEL 7 supports NFSv4 (version 4 of the protocol) by default, and falls back automatically to NFSv3 and NFSv2 if that isn't available.

NFSv4 uses the TCP protocol to communicate with the server, while older versions of NFS may use either TCP or UDP.

NFS servers export shares (directories) and NFS clients mount an exported share to local mount point (directory).

The local mount point must exists. NFS shares can be mounted a number of ways:

- manually mounting an NFS share using the mount command.

- automatically mounting an NFS share at boot time using /etc/fstab.

- mounting an NFS share on demand through a process known as automounting

Securing file access on NFS shares

NFS servers secure access to files using different methods: none, sys, krb5, krb5i and krb5p. The NFS server can choose to offer one or more methods for each exported share. NFS clients must connect to the exported share specifying the mount option sec=[method]

Security methods

none anonymous access to the files, writes to the server (if allowed) will be allocated UID and GID of nfsnobody.

sys file access based on standard Linux file permissions for UID and GID values. Default if not specified.

krb5 client must prove identity using Kerberos and then standard file permissions.

krb5i adds a cryptographically strong guarantee that the data in each request hasn't been tampered with.

krb5p adds encryption to all requests between the client and the server, preventing data exposure on the network. Performance impact.

Kerberos options will require, as a minimum, a /etc/krb5.keytab and additional authentication configuration (joining the Kerberos Realm).

The /etc/krb5.keytab will normally be provided by the authentication or security administrator. Request a keytab that includes either a host principal, nfs principal, or (ideally) both.

NFS uses the nfs-secure service to help negotiate and manage communication with the server when connecting to Kerberos-secured shares.

It must be running to use the secured NFS shares

---

sudo systemctl [enable,start] nfs-secure

---

The nfs-secure is part of the nfs-utils package.

Mount an NFS share

Alright, yes, we need to set a secure parameter as a server and stuff, but how do we use this from a client side?

There are three basic steps to mounting an NFS share:

- Identify: the administrator for the NFS server can provide export details, including security requirements.

NFSv4 can be identified by mounting the root folder of the NFS server and exploring the exported directories. Do this as root.

Access to shares that are using Kerberos will be denied but the share (directory) will be visible.

---

sudo mkdir /mountpoint

sudo mount [server]:/ /mountpoint

---

We create the folder "mountpoint" and then we mount the server root's folder on that directory we just created.

---

showmount -e serverX

---

NFSv2 and NFSv3 shares can be discovered using the command  showmount.

- Use mkdir to create a mount point in a situable location

---

mkdir -p /mountpoint

---

- Mount: we can mount the NFS manually (using the command mount) or incorporate it to the /etc/fstab file.

---

# mount -t nfs -o sync [server]:/share /mountpoint

---

Mount manually the share. The -t nfs option is the file system type for NFS shares (not required, shown for completeness).

The -o sync option tells mount to immediately synchronize write operations with the NFS server (asynchronous by default).

The default security method (sec=sys) will be used to try mounting the NFS share, using the standard Linux file permissions.

---

[server]:/share  /mountpoint  nfs  sync  0 0

---

This is the line that we must add to /etc/fstab if we want to mount the NFS share automatically at boot time.

If we want to unmount the share manually, we use the umount command

---

# umount /mountpoint

---

Automounting network storage with NFS

Mounting NFS shares with the automounter

The automounter is a service (autofs) that can automatically mount NFS shares "on demand", and will automatically unmount NFS shares when they're no longer being used.

Automounter benefits:

- Users don't need to have root privileges to run mount/umount.

- NFS shares configured in the automounter are available to all the users (if they have permissions).

- NFS shares are not permanently connected like entries in /etc/fstab.

- Configured entirely client side.

- Uses the same mount options, including security options.

- Support for both direct and indirect mount point mapping.

- Indirect mount points are created and removed by autofs.

- Automount a range of different file systems (more than just NFS).

- autofs is a service that is managed like other system services.

Direct mount : where you have a known and previously created mount point.

Indirect mount : the mount points don't have to exist already, they'll be created dynamically.

Creating an automount

- Install autofs (use yum if you don't have it).

- Add a master-map file to /etc/auto.master.d

This file identifies the base directory used for mount points and identifies the mapping file used for creating the automounts.

The name of the file doesn't matter but it's normally something meaningful.

Every file must have an .autofs extension.

The master-map file can hold multiple mapping entries, or use multiple files to separate configuration data.

---

/shares /etc/auto.demo

---

This one is an indirect map.

The base point is /shares and the information about anything created inside of it is available at the /etc/auto.demo file.

---

/- /etc/auto.direct

---

All the direct map entries use "/-" as the base directory. The mapping file that contains the mount details is /etc/auto.direct

The mapping file identifies the mount point, mount options and source location to mount.

```
---------------------------------------
work   -rw,sync   serverX:/shares/work
|--------|----------|-------------------
|      |       Source location
|      Mount options
Mount point
```

The file name isn't important but by convention is located in /etc and called auto.[name]

The mount point (known as key in the man pages) will be created and removed automatically by the autofs service. In the example, the fully mount point will be /shares/work. The /shares directory and the work directory will be created and removed as needed by the autofs service.

The local mount point mirrors the server's directory structure. The local mount point can be named anything. There's no need to align the names of the local mount point and the server directory structure.

Mount options start with a dash (-) and are comma-separated with no whitespace. The mount options available are the same available on a manual mount command.

There are some automounter specific options like -fstype= and -strict. Use fstype to specify the file system (if it's not NFS) and strict to treat errors, when mounting file systems, as fatal.

The source location for NFS shares follows the host:/pathname pattern.

If the file system to be mounted begins with a slash (/), such as local device entries or SMB shares, then a colon (:) needs to be prefixed.

For a SMB share would be ://serverX/share

Inside of a direct map file we'll have the mount point, mount options and source location.

---

/mnt/docs -rw,sync serverX:/shares/docs

---

The mount point always (or key) is always an absolue path, starting with slash (/). The rest of the mapping file uses the same structure.

The mapping file-indirect wildcard maps

---

* -rw,sync serverX:/shares/&

---

The ampersand (&) at the end will match the asterisk at the beginning.

The mount point is an asterisk and the subdirectory on the source location is an ampersand.

You don't need to specify sec= unless you'd been told to.

#smbnetworkstorage

The Server Message Block (SMB) protocol and Common Internet File System (CIFS) are the same. CIFS is a variation of SMB.

You may find utilities that use the letters CIFS instead of SMB.

In order to grab a share from an SMB server, you'll have to use tools that'll allow you to identify the remote shares that you want to access.

Mount SMB Share

- Identify: the administrator for the SMB server host can provide share details, such as username and password, share names, etc. An alternative is to use a client that can browse the shares, such as  smbclient

---

smbclient -L //serverX

---

The -L option asks the  smbclient  to list the shares available on serverX.

- Mount point: Use mkdir to create a mount point in a suitable location.

- Mount: manual mount or /etc/fstab file.

---

# mount -t cifs -o guest //serverX/share /mountpoint

---

The -t cifs option is the file system type for SMB shares and the -o guest option tells mount to try and authenticate as a guest account without a password.

---

//serverX/share /mountpoint cifs guest 0 0

---

/etc/fstab entry

Authentication to SMB shares

SMB shares can be flagged as non-browsable, meaning clients such as smbclient will not display them.

The SMB shares can still be accessed if you explicitly specify the SMB share name.

---

# mount -t cifs -o username=watson //serverX/cases /bakerst/cases

---

We specify to connect as the user watson.

---

# mount -t cifs -o credentials=/secure/sherlock //serverX/sherlock /home/sherlock/work

---

Here we're loading the credentials from the /secure/sherlock file.

The format for the credentials file is:

username=username

password=password

domain=domain

It should be placed somewhere secure with only root access (chmod 0600).

During file operations, the SMB will check file access against the credentials used to mount the share.

The client will check file access against the UID/GID of the files sent from the server.

The client will need to have the same UID/GID (and maybe also supplementary group membership) as the files on the SMB server.

Mounting SMB file systems with the automounter

Add an auto.master.d configuration file that identifies the base directory for shares and the associated mapping.

Create or edit the mapping file to include the mount details for the SMB share.

Enable and start the autofs service.

The mapping file

The file system type needs to be specified with the -fstype=cifs option. The URL needs to be prefixed with a colon (:).

#boottroubleshooting

High-level overview of the tasks involved for a physical x86_64 system booting.

- Machine is powered on. The system firmware (either UEFI or BIOS) runs a Power On Self Test (POST) and initialize some hardware.

- The system firmware searches for a bootable device (either configured in the UEFI or searching the MBR on all disks).

- The system firmware reads a boot loader from disk, then passes control of the system to the boot loader (on RHEL 7, grub2; configured using grub2-install).

- The boot loader loads it's configuration from disk and presents the user with a menu of possible configurations to boot.

/etc/grub.d/ /etc/default/grub /boot/grub2/grub.cfg (it gets overwritten by the system)

- After selecting an entry, the boot loader loads the configured kernel and initramfs from disk and places them in memory. An initramfs is a gzip-ed cpio archive containing kernel modules for all hardware necessary at boot, init scripts and more. On RHEL 7, the initramfs contains an entire usable system by itself.

/etc/dracut.conf

- The boot loader hands control of the system over to the kernel, passing in any options specified on the kernel command line in the boot loader, and the location of the initramfs in memory.

- The kernel initializes all hardware for which it can find a driver in the initramfs, then executes /sbin/init from the initramfs as PID 1. On RHEL 7, initramfs contains a working copy of systemd as /sbin/init, as well as a udev daemon. (configured using init= parameter).

- The systemd instance from the initramfs executes all units for initrd.target (includes mounting the actual root file system on /sysroot).

/etc/fstab

- The kernel root file system is switched (pivoted) from the initramfs root file system to the system root file system that was previously mountedon /sysroot, then systemd re-executes itself using the copy of systemd installed on the system.

- systemd looks for a default target, either passed in from the kernel command line or the configured on the system, then starts (and stops) units to comply with the configuration for that target.

/etc/systemd/system/default.target  /etc/systemd/system

Boot, reboot and shut down

Nowadays, the commands poweroff and reboot are aliases of systemctl [poweroff|reboot].

systemctl halt and halt won't power off the system but they will bring a system down to a point where it's safe to manually power it off.

Selecting a systemd target

A systemd target is a set of units that should be activated to reach a desired system state.

The purpose of a target is to be an organization point.

The most important targets are:

graphical.target System supports multiple users, graphical and text-based logins.

multi-user.target System supports multiple users, text-based logins only.

rescue.target sulogin prompt, basic system initialization completed.

emergency.target sulogin prompt, initramfs pivot complete and system root mounted on / read-only.

emergency and rescue are targets used for fixing things.

It's posible for a target to be a part of another target (graphical.target includes multi-user.target, which depends on basic.target).

---

systemctl list-dependencies graphical.target | grep target

---

List all the dependencies for the graphical.target

---

systemctl list-units --type=target --all

---

List all the available targets and their current status.

---

systemctl list-unit-files --type=target --all

---

List all the installed targets on disk.

A target is a declaration that we have reached a certain point in the boot process.

Selecting a target at runtime

---

systemctl isolate multi-user.target

---

Isolates a target by stopping all the services that aren't required by that target (and it's dependencies).

Not all targets can be isolated, only those with the AllowIsolate=yes flag. For example, graphical.target can be isolated but cryptsetup.target cannot.

Setting a default target

When the system starts and control is passed over to systemd from the initramfs, it will try to activate the default.target (normally it will be a symlink to graphical.target or multi-user.target, located at /etc/systemd/system)

---

systemctl set-default graphical.target

---

This will change the default target and create a symbolic link to that target.

You can override the default target at boot time by appending to the kernel
systemd.unit=[target]

---

systemd.unit=rescue.target

---

The system will load the rescue.target instead of the default one.

Steps

- (Re)boot the system.
- Interrupt the boot loader menu countdown by pressing any key (except Enter).
- Move the cursor to the entry (not the mouse).
- Press  e  to edit the current entry
- Move the cursor to the line that starts with linux16. This is the kernel command line.
- Append  systemd.unit=[desired].target  .
- Press  Ctrl + X  to boot with these changes.

Repairing Common Boot Issues

Recovering the root password

One task that every system administrator should be able to accomplish is recovering a lost root password.

There are different ways to change the root password in case you forgot it.

Boot the system using a Live CD, mount the root file system from there, and edit the /etc/shadow

On RHEL 6 and earlier, an administrator could boot the system into runlevel 1 and be presented with a root prompt.

The closest analogs to runlevel 1 on RHEL 7 are the rescue.target and emergency.target targets, both require the root password.

- (Re)boot the system.

- Interrupt the boot loader countdown by pressing any key.

- Move the cursor to the entry that needs to be booted.

- Press e to edit the selected entry.

- Move the cursor to the kernel command line (the line that starts with linux16).

- Append rd.break (this wil break just before control is handed from the initramfs to the actual system).

- We can also remove the quiet keyword.

- Press Ctrl + x to boot with the changes.

At this point, a root shell will be presented, with the root file system for the actual system mounted as read-only on /sysroot

Note: SELinux isn't enabled at this point, so any new files won't have a SELinux context assigned to them. Some tools (such as passwd) first create a new file, then move it in place of the file they are intended to edit, effectively creating a new file without an SELinux context.

---

mount -oremount,rw /sysroot

---

We need to remount the file system as read-write

---

chroot /sysroot

---

Switch the chroot jail, where /sysroot is treated as the root of the file system tree.

---

passwd root

---

Now we change the password of root

---

touch /.autorelabel

---

This file will make SELinux to relabel the whole system, applying the right SELinux context to each file.

Type  exit  twice, one for exit the chroot jail and the second to exit the initramfs debug shell.

After this procedure, the system will continue booting, perform a full SELinux relabel, then reboot again.

Using journalctl

It can be useful to look at the logs of previous (failed) boots. If the journald log has been made persistent, this can be done with the journalctl tool.

First make sure that you have persistent journald logging enabled:

---

mkdir -p -m2775 /var/log/journal

chown :systemd-journal /var/log/journal

killall -USR1 systemd-journald

---

---

journalctl -b-1 -p err

---

This will filter previous logs, looking for errors.

Diagnose and repair systemd boot issues

Early debug shell

---

systemctl enable debug-shell.service

---

Spawns a root shell on TTY9 (Ctrl + Alt + F9) early during the boot sequence.

This shell is automatically logged in as root, so that an administrator can use some of the other debugging tools while the system is still booting.

Remember to disable debug-shell.service when you're done, otherwise an unauthenticated root shell will be open for anyone to access it.

Emergency and rescue targets

Appending systemd.unit=rescue.target or systemd.unit=emergency.target t o the kernel command line from the boot loader, the system will spawn into a special rescue or emergency shell.

Both of those shells require the root password.

emergency.target keeps the root file system mounted read-only, while rescue.target wait for the sysinit.target to complete first.

These shells can be used to fix any issues that prevent the system from booting normally (dependency loop between services, incorrect entry in /etc/fstab).

Exiting those shells will continue with the regular boot process.

Repairing File System Issues at Boot

Errors in /etc/fstab and corrupt file systems can stop a system from booting.

In most cases, systemd will actually continue to boot after a timeout, or drop an emergency repair shell that requires the root password.

Corrupt file system:

systemd will attempt a fsck. If the problem is too serious, the user will be prompted to run fsck manually from an emergency shell.

Non-existent device/UUID referenced in /etc/fstab

systemd will wait for a set amount of time, waiting for the device to become available. If it doesn't happens, the user is dropped to an emergency shell after the timeout.

Non-existen mount point in /etc/fstab

systemd will create the mount point if possible, otherwise it drops to an emergency shell.

Incorrect mount option specified in /etc/fstab

The user is dropped to an emergency shell.

In all cases, an administrator can use the emergency.target to diagnose and fix the issue.

Repairing Boot Loader Issues

grub2 is the default boot loader on RHEL 7.

grub2 is the second major version of the GRand Unified Bootloader.

The main configuration file for grub2 is /boot/grub2/grub.cfg but administrators aren't supposed to edit that file directly.

There's a tool called grub2-mkconfig that generates that configuration using a set of different configuration files, and the list of installed kernels.

grub2-mkconfig will look at /etc/default/grub for options such as the default menu timeout and kernel command line to use.

Then use a set of scripts in /etc/grub.d/ to generate a configuration file.

---

grub2-mkconfig > /boot/grub2/grub.cfg

---

Redirect the output of grub2-mkconfig to make the changes permanent.

When you have to make major changes, better not redirect the output, so you can inspect the changes first.

Important directives

Actual bootable entries are encoded inside menuentry blocks.

Remember that linux16 and initrd16 point to the kernel to be loaded from disk and the initramfs to be loaded. These are your kernel lines.

Tab completion is available to find those files.

The set root lines inside those blocks do not point to the root file system for the RHEL 7 system. They point to the file system from which grub2 should load the kernel and initramfs files. The syntax is harddrive,partition , where hd0 is the first hard drive in the system. msdos1 indicates the first MBR partition. gpt1 indicates the first GPT partition.

Reinstalling the boot loader

Sometimes the boot loader itself become corrupt, so you can reinstall it using the command grub2-install

On BIOS systems, the disk where grub2 should be installed in the MBR should be provided as an argument.

On UEFI systems, no argument is required when the EFI system partition is mounted on /boot/efi

Quiz

5 The system firmware loads the boot loader.

9 The boot loader loads its configuration from disk.

3 The boot loader presents the user with a menu.

1 A kernel and initramfs are loaded from disk.

4 The kernel initializes and launches /sbin/init from the initramfs.

7 Basic hardware initialization takes place.

6 The system root file system is mounted read-only on /sysroot.

8 The root file system is switched, and control is passed over to a new systemd instance

2 All units for the default target are started.

#firewalld

It's a new service that is in RHEL 7, extending the capabilities of network filtering.

Netfilter and firewalld concepts

The Linux kernel includes a powerful network filtering subsystem called netfilter

It allows kernel modules to inspect every package traversing the system.

This means any incoming, outcoming, or forwarded network packet can be inspected, modified, dropped or rejected in a programmatic way, before reaching components in user space.

Interacting with netfilter

You can write your own kernel modules to interact with netfilter but it's not typically done.

The command `iptables` is a low-level tool used to manage firewalls. Only adjusts IPv4 firewall rules.

ip6tables is used for IPv6 and ebtables for software bridges.

## Introducing firewalld

firewalld replaces iptables (also replaces ip6tables and ebtables).

It's a system daemon that can configure and monitor the system firewall rules.

Applications request ports using the DBus messaging service (feature that can be disabled or locked down).

Can be installed with the `firewalld` package. It's also part of the base install but not part of the minimal install.

firewalld simplifies firewall management by classifying all network traffic into zones. Based on criteria such as the source IP address packet or the incoming network interface, traffic is diverted into the firewall rules for the appropiate zone.

Each zone can have it's own list of ports and services to be opened or closed.

Every packet has a source address. If that source address is tied to a specific zone, the rules for that zone will be parsed.

If not, the incoming network interface will be used.

If the network interface is not associated with a zone for some reason, the default zone will be used.

The default zone is not a separate zone itself; it's one of the other zones.

Most zones will allow traffic through the firewall which matches a list of particular ports and protocols ("631/udp") or pre-defined services ("ssh").

The trusted zone permits all traffic by default.

home Reject incoming traffic unless related to outgoing traffic or matching ssh, mdns, ipp-client, samba-client or dhcpv6-client.

internal Same as the home zone.

work Reject incoming traffic unless related to outgoing traffic or matching ssh, ipp-client or dhcpv6-client.

public Used by default. Reject incoming traffic unless related to outgoing traffic or matching ssh or dhcpv6-client.

external Reject incoming traffic unless related to outgoing traffic or matching ssh. Outgoing IPv4 traffic forwarded through this zone is masqueraded.

dmz Reject incoming traffic unless related to outgoing traffic or matching ssh.

block Reject all incoming traffic unless related to outgoing traffic.

drop Drop all incoming traffic unless related to outgoing traffic (do not even respond with ICMP errors).

Check all the available pre-defined zones and their intended uses on

---

man 5 firewalld.zones

---

Pre-defined services

firewalld comes with some pre-defined services.

ssh local ssh server. Traffic to 22/tcp

dhcpv6-client local DHCPv6 client. Traffic to 546/udp on the fe80::/64 IPv6 network

ipp-client local IPP printing. Traffic to 631/udp

samba-client local Windows file and print sharing client. Traffic to 137/udp and 138/udp

mdns multicas DNS (mDNS) local-link name resolution. Traffic to 5353/udp to the 224.0.0251 IPv4 or ff02::fb (IPv6) multicast addresses

---

firewall-cmd --get-services

---

List other pre-defined services.

The configuration files that define the ones included in the firewalld package can be found in the /usr/lib/firewalld/services directory, in the format defined by firewalld.zone (5).

The easiest options for a system administrator new to firewalld is to either use pre-defined services or to explicitly specify the port/protocol they wish to permit. The firewall-config graphical tool can also be used to configure the firewall.

Any permanent changes that you made to firewalld won't be active until firewalld is restarted. Any changes during Runtime won't survive a restart.

Configure firewall settings with firewall-cmd

firewall-cmd is installed as part of the main firewalld package. It can perform the same actions that firewall-config can.

---

firewall-cmd [--option]

---

All the changes will apply to the runtime configuration, unless the --permanent option is specified.

Many of the command take the --zone=<ZONE> option to determine which zone they affect.

--get-default-zone query the current default zone

--set-default-zone=<ZONE> set the default zone. Changes both runtime and permanent configuration

--get-zones list all available zones

--get-active-zones list all zones currently in use

--add-source=<CIDR> route all traffic coming from the IP address or network/netmask <CIDR> to the specified zone (--zone=<ZONE>, or default)

--remove-source=<CIDR> remove the rule routing all traffic coming from the IP address or network/netmask <CIDR> from the specified zone (or default)

--add-interface=<INTERFACE> route all traffic from <INTERFACE> to the specified zone (--zone=, otherwise default)

--change-interface=<INTERFACE> associate the interface with <ZONE> (--zone=, otherwise default)

--list-all [--zone=<ZONE>] list all configured interfaces, sources, services and ports for <ZONE> (otherwise default)

--list-all-zones retrieve all information for all zones

--add-service=<SERVICE> allow traffic to <SERVICE> (--zone=, otherwise default)

--add-port=<PORT/PROTOCOL> allow traffic to the <PORT/PROTOCOL> port(s) (--zone=, otherwise default)

--remove-service=<SERVICE> remove <SERVICE> from the allowed list for the zone (--zone=, otherwise default)

--remove-port=<PORT/PROTOCOL>  remove the <PORT/PROTOCOL> port(s) from the allowed list (--zone=, otherwise default)

--reload  drop the runtime configuration and apply the persistent configuration

---

firewall-cmd --set-default-zone=dmz

firewall-cmd --permanent --zone=internal --add-source=192.168.0.0/24

firewall-cmd --permanent --zone=internal --add-service=mysql

firweall-cmd --reload

---

Changes the default zone to dmz, assigns all traffic coming from the 192.168.0.0/24 network to the internal zone, also opens the network ports for mysql on the internal zone.