

Red Hat System Administration I

RH124

PRESENTATION BY:
MOHAMED EL ERAKY



RedHat

Agenda

- 1. Introduce Linux and the Red Hat Enterprise Linux ecosystem
- 2. Run commands and view shell environments.
- 3. Manage, organize, and secure files.
- 4. Manage users, groups.
- 5. Control and monitor systemd services.
- 6. Configure remote access “SSH”
- 7. Configure network interfaces and settings.
- 8. Manage software using DNF





Introduce Linux and the Red Hat Enterprise Linux ecosystem

We will cover these Topics

- Linux History brief
- Filesystem Hierarchy Standard (FHS)

Linux History brief

In 1984, Richard Stallman, an American software engineer, had a goal to create an operating system a completely free UNIX-compatible open source (non-proprietary)

The initiative was called the GNU Project (GNU's Not Unix), Richard wants to make an open-source Operating system.

by 1991, Richard had been developed significant software's.

The only critical piece missing was a core software component called kernel to drive and control the GNU software and to regulate its communication with the hardware, So Richard had created bunch of software's successfully. However, he failed to create and a kernel that should regulate the communications between the GNU software's and the hardware.



Linux History brief

Around the same time, The computer science student Linus

Torvalds developed a kernel that manage resources, memory, and storage, Also provide Multi tasking “switcher tasks” and proclaimed its availability.

The new kernel was named *Linux*, and it was gradually integrated with the GNU software that Richard had created before, So They’re created a compatible OS called *GNU/Linux*, *Linux operating system*, or simply *Linux*.

Linux was released under the GNU *General Public License* (GPL).

Initially written to run on Intel x86-based computers, the first version (0.01) was released in September 1991



Linux History brief

The GPL license is to ensure that the code is published as an open source.

The Linux kernel, and the operating system in general, has been enhanced with contributions from tens of thousands of software programmers, amateurs, and organizations around the world.

Therefore, Currently there are a variety of Linux distributions out there (e.g. RedHat, Debian, CentOS, Arch, etc..)



Filesystem Hierarchy Standard (FHS)

The Linux directory structure follows the Filesystem Hierarchy Standard (FHS), which provides a consistent organization and naming scheme for directories in a Linux-based operating system.

On Windows machines you access your data with the drive letter then the path for data. on Linux machines it's the quite opposite.

Linux world have a wide Rule called **Everything is a file**. So, The Disk appears as a special file under /dev And you're mount This disk file to any path on the system in order to access it.

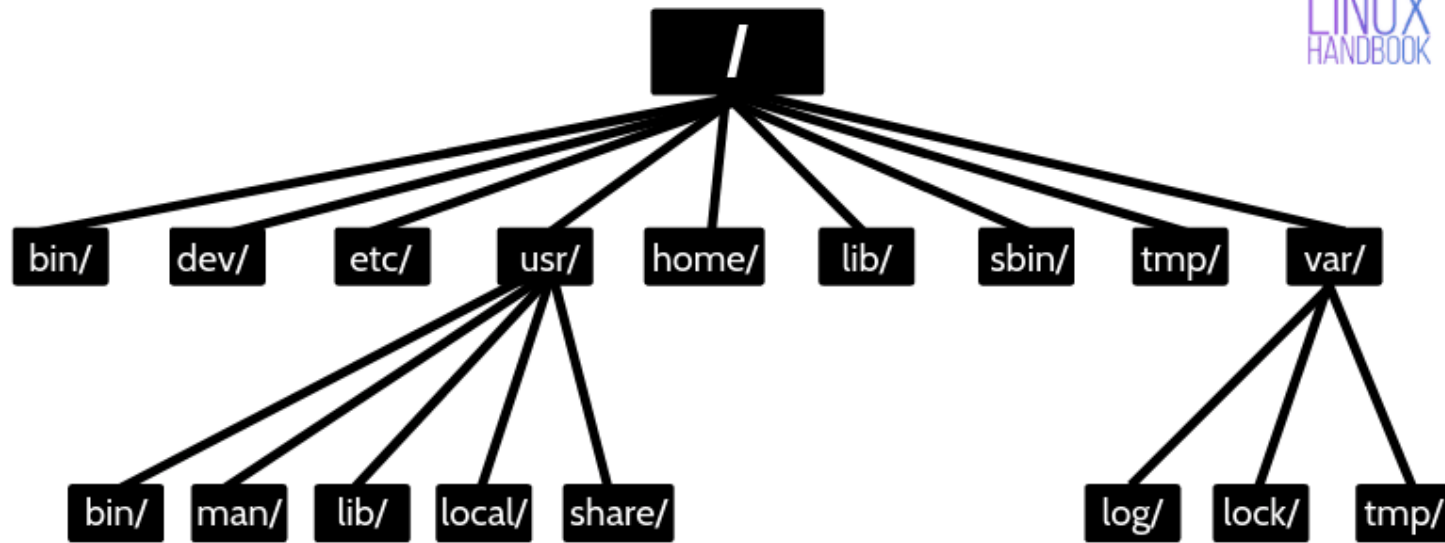
Linux machine have the / path, it's the main path for the system this path include all your system paths.

So, on Linux machines you access the data with the path then the mounted disk un-like windows.



Filesystem Hierarchy Standard (FHS)

Let's discover the Linux directory structure:



LINUX
HANDBOOK



Filesystem Hierarchy Standard (FHS)

- /bin - Binaries

The /bin directory contains the executable files of many basic commands like ls, cp, cd, etc.

- /sbin – System binaries

This similar to /bin however it's contains the executable binary commands that can run only by the root

- /dev- Device files

This directory only contains special files that are related to the devices, these are virtual files, physically on the disk. All system device files

- /etc- Configuration files

The /etc directory contains The core configuration files of the system, use primarily by the administrators and services



Filesystem Hierarchy Standard (FHS)

- /usr - User binaries and program data

The /usr contains all the executable files, libraries, and sources for most of the system programs, because of that most of the files are read-only. In fact, most of the files under **/bin and /lib** and /lib64 are **Linked** “shared under” **/usr**.

- /home – User personal data

The home directory for users, personal directories for the users.

- /lib – Shared library

The directory holds the **libraries needed by the binaries** in **/bin** and **/sbin** directories.

- /tmp – Temporary files

As the name suggests, this directory holds temporary files of the applications



Filesystem Hierarchy Standard (FHS)

- /media – mount point for removable media

when you connect a removable media such as a USB or DVD, a directory is created automatically.

- /mnt – Mount directory

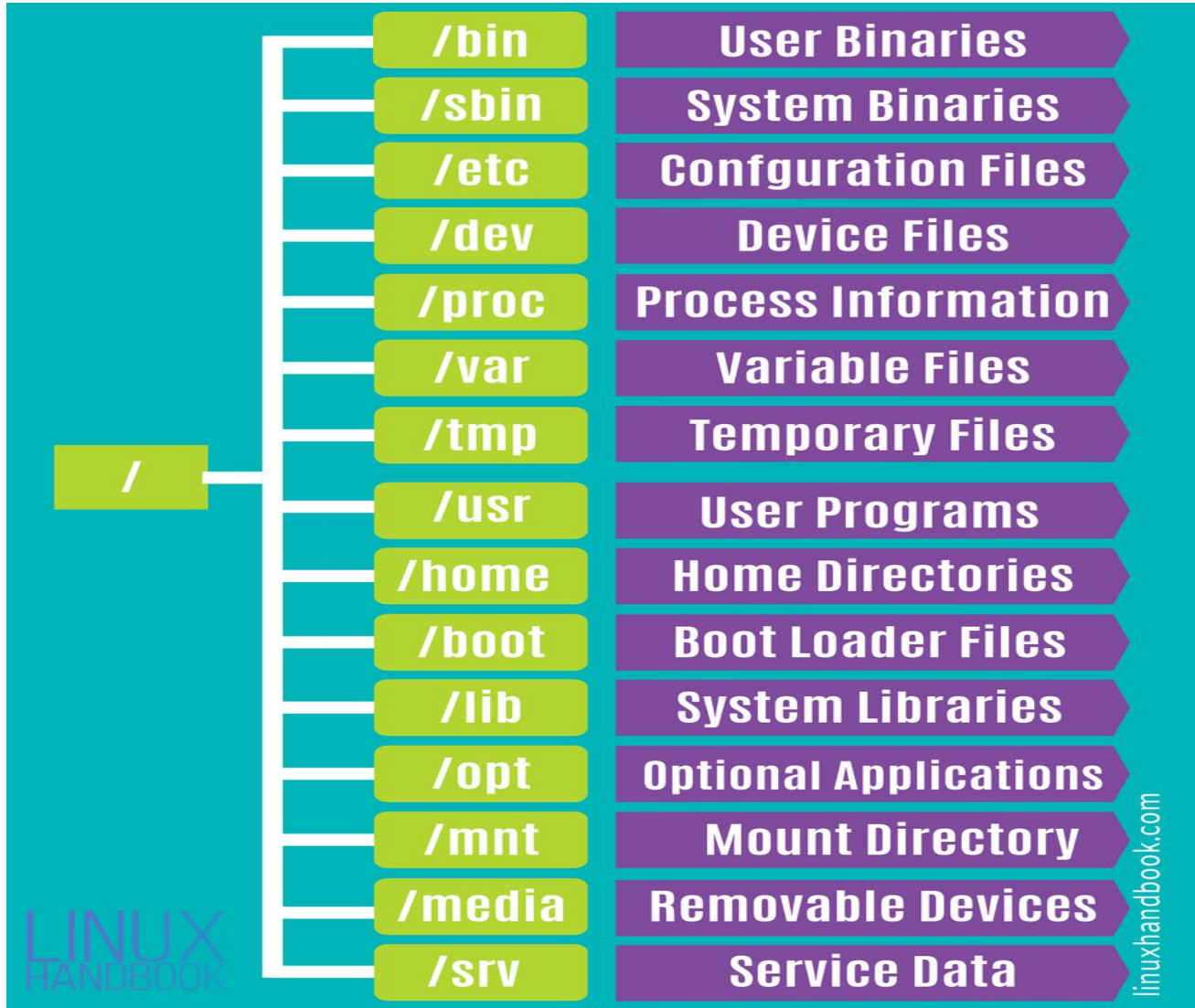
Created for you for locally mount, This is similar to the /media directory but instead of automatically mount in the removable media.

- /srv – Service data

The /srv directory contains data for services provided by the system. For example, if you run a HTTP server, it's a good practice to store the website data in the /srv directory



Filesystem Hierarchy Standard (FHS)





Run commands and view shell environments.

We will cover these Topics

- Linux basic command
 - print system info

Linux basic command

In This slide will try to Print-out the system information:

```
uname # display system info
uname -a # display all system info
uname -r # kernel version
uname -n # hostname name
hostname # get server hostname
hostnamectl set-hostname <new_name> # set hostname
cat /etc/os-release # display all system info
cat /etc/redhat-release
lscpu # check your cpu
lsmem # check your memory
free -h # get memory info
Lshw
```

```
date # print date
date +%x
```

```
# calendar
cal
cal 2019
cal 9 2019
```



Linux basic command

```
ps
```

```
ps -u <username>
```

```
ps aux # get run-time process
```

```
top # get run-time process in interactive mode
```

```
top -u <username>
```

```
# top Options
```

```
# - Press 1 to get how many of your processors
```

```
# - Press s to change the default refresh rate
```

```
# - press t CPU usage
```

```
# - press m memory usage
```

```
# - Press k to kill process
```

```
# - Press w to write "save"
```

```
# - SHIFT + M Sort by memory consumption
```

```
# - SHIFT + P Sort by process consumption
```

```
# - r for renice
```

```
# Renice values are from -20 to 19
```



Linux basic command

```
# help command
```

```
date -help
```

```
Man date
```

```
whoami # print out your account name
```

```
who # name of account's are login now
```

```
w # provide more info
```

```
last # get info for each user login time
```

```
Lastlog # get brief info
```

```
tty # display the terminal number
```





Manage, organize, and secure files.

We will cover these Topics

- Linux basic command
 - Copy, move, create, delete, permissions, and organize files while working from the bash shell.

Linux basic command

- in this slide will try to Copy, move, create, delete, set permissions, and organize files while working from the bash shell.

```
cd / # change directory
cd ~ # go to your home dir
cd - # return to the last path
cd .. # return back on step | cd ../../ 2 step back
```

```
mkdir dir1 dir2 # multiple dirs
mkdir -p dir3/dir4/dir5 # dir in dir
ls -lathR
```

```
touch file1 file2 # create file one
vim file1 # edit file one content
cat file1 # print out content
```

```
cp file1 dir1 # copy file
cp file1 file2 dir2/ # copy multiple files
cp -r dir1 dir2 # recursive copy
```

```
ls # List current dir content
ls -l / # list / dir content
```



Linux basic command

```
ls -la # list all | any file starts with . is hidden
```

```
mv file1 dir3 # rename files with move  
mv file /dev/null # delete the file
```

```
rm -r file1 # delete none empty file  
rm -rf dir1 # delete by force
```

```
CTRL + u # delete line  
CTRL + k # delete line  
CTRL + arrow # navigation  
CTRL + a # go to the start line  
CTRL + e # go to the end | Press on end  
CTRL + SHIFT + c # past content
```



Linux basic command

Let's discover some advance stuff

```
tail file1  
tail -n 5 file1  
tail -f file1  
head file1
```

```
cat file1 | grep <value>  
cat file1 | grep -i <value>
```

```
grep <name> <path of file> # use -n print line number  
grep <name> /etc/passwd  
grep cat /usr/share/dict/words # grep on cat word in  
dict files  
grep ^cat /usr/share/dict/words # grep words that  
starts with cat only  
grep cat$ /usr/share/dict/words # grep words that ends  
with cat only  
grep ^cat$ /usr/share/dict/words # grep words that  
starts with cat and ends with cat only  
grep -r <word> /etc/ # search for this word in all  
files under /etc  
grep -rl <word> /etc/
```



Linux basic command

```
ps aux | awk '{print $1}' # specify a column  
cat /etc/passwd | awk -F : '{print $1}'  
cat /etc/passwd | grep -i <username> | awk -F : '{print $1}'
```

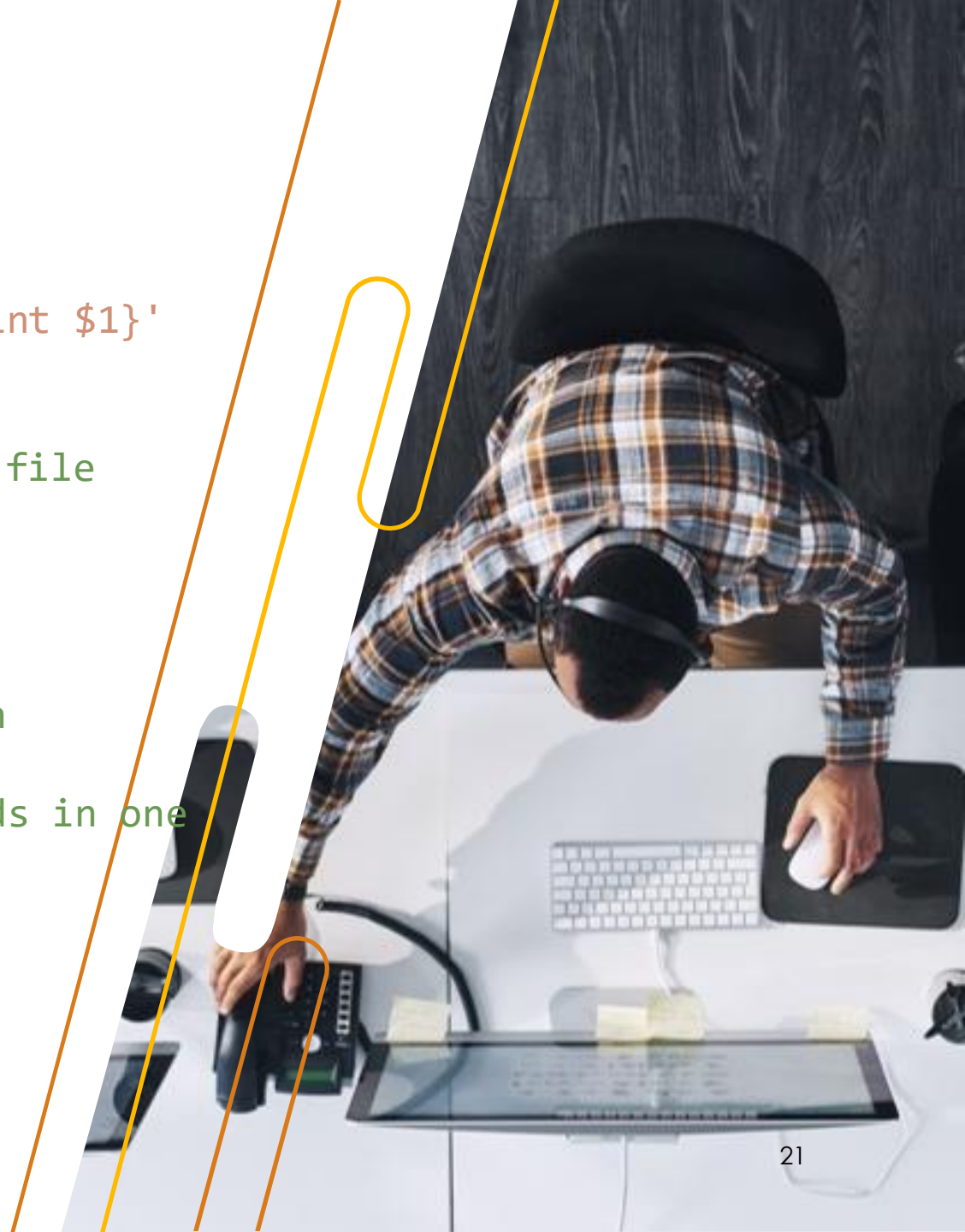
```
cut -f 1 -d : /etc/passwd # specify a column  
cut -f 1 -d : /etc/passwd > file1 # Save in external file
```

```
!ls # get the last value of running cut command  
!cat
```

```
ps -u <username> && echo "$HOSTNAME" # AND condition  
Ps -u <username> || echo "$HOSTNAME" # Or condition  
ps -u <username>; echo "$HOSTNAME" # multiple commands in one  
line
```

```
top & # run in the background  
top -p <procid> # get process
```

```
du -h <file path> # get file/list_files size  
du -hs <file path> # get total size only
```



Linux basic command | VIM

```
1 # vim editor
2 Vim file.txt
3
4 # - on exec mode Press esc to go to exec mode
5 # - on exec mode Press i to go to insert mode
6 # - on exec mode Press dd to cut a line, also you can use it for deletion
7 # - on exec mode Press yy to copy a line
8 # - on exec mode Press d10 to cut the next 10 lines, also you can use it for deletion
9 # - on exec mode Press p for past
10 # - on exec mode Press 12G to navigate to line 12
11 # - on exec mode Press /<search-key> for searcheing, and use n, N for navigation
12 # - on exec mode Press :$ to go to the end of the file
13 # - on exec mode Press :0 to navigate to the top of the file
14 # - on exec mode Press :%s/oldword/NEWWORK subtitude oldword with NEWWORK
15 # - on exec mode Press :set number to enable line number
16 # - on exec mode Press q for quit
17 # - on exec mode Press w for write
18 # - on exec mode Press wq for write and quit
19 # - on exec mode Press x for write and quit
20 # - on exec mode Press q! force quit
```



Linux basic command | Permissions

```
377
378 -rwxrwxrwx
379
380 u      g      o
381 owner  group  other
382 rwx    rwx    rwx
383
384 r      w      x
385 4      2      1
386
387 ll file
```



Linux basic command | Permissions

```
ll -d file # print file permissions
ls -l # print all dir files and permissions
chmod u=rw,g=r,o=r file1 # change permissions
chmod 775 file1
```

Access control list ACL

```
ll -d file1 # ensure that there is +
getfacl file1 # get current permissions
setfacl -m u:mohamed:rw file # Set permissions
setfacl -m g:mohamed:rw file
```

1- you should set permission first with chmod then ACL

2- ACL inherit from chmod, meanwhile chmod did not





Manage users, groups

We will cover these Topics

- Create and manage users
- Create and manage groups
- Add a user to group

Manage users, groups

```
=== users categories ===
```

```
# 1- super user
```

```
# 2- user
```

```
# 3- service account
```

```
=====
```

```
# 1- super user
```

```
sudo vim /etc/sudoers # add user to sudoers file
```

```
# 2- Normal user
```

```
sudo useradd <username> # add user
```

```
sudo passwd <username> # set password
```

```
sudo useradd <username> -s /usr/bin/sh # specify a shell
```



Manage users, groups

```
# 3- service account
```

```
useradd --system --no-create-home <username> # create a  
user without home dir
```

```
useradd --system --no-create-home -s /usr/sbin/nologin  
<username> # create a user without home dir and no console  
login
```

```
sudo useradd --system --no-create-home <username> -s  
/usr/sbin/nologin -p <username> # we set a password for  
test purose
```

```
su - <username> # login as
```

```
# print users, passwords, and groups
```

```
tail /etc/passwd
```

```
tail /etc/shadow
```

```
tail /etc/group
```



Manage users, groups

```
groupadd sales # add group
```

```
tail -n 1 /etc/group # get created group
```

```
groupmod -n <newname> <oldname> # rename a group
```

```
groupdel <groupname> # Delete a group
```

```
id <username> # get user info
```

```
group <username> # get user group member
```

```
sudo usermod -aG sales mohamed # Append user to a group
```

```
sudo gpasswd -d <username> <groupname> # delete a user from a group  
| sudo deluser mohamed root
```

```
# Delete user
```

```
sudo deluser --remove-all-files mohamed # this works only on ubuntu
```

```
sudo userdel <username> --remove -f # Delete a user and his home  
directory
```





Control and monitor systemd services.

We will cover these Topics

- Control and monitor network services and system daemons with the systemd service.

Control and monitor systemd services.

The first service start on the system is systemd "system daemon" Then systemd is responsible to start all the services.

systemd will segregate the performance of CPU and memory on the services to start in parallel.

Systemd provide systemctl command to manage services, systemctl is not to manage services only but process and more.

```
systemctl -t service # list active and exited services
```

```
systemctl -t service --all # print all services
```

```
systemctl list-units # print units
```

```
systemctl list-units --all # print all units
```

```
systemctl list-units --type service
```

```
systemctl list-units --type service --all # all active and inactive service
```

```
systemctl status sshd
```

```
systemctl start sshd # start the service
```



Control and monitor systemd services.

```
systemctl status sshd
```

```
systemctl start sshd # start the service
```

```
systemctl enable sshd # enable to start auto while booting
```

```
systemctl restart sshd # restart the service
```

```
systemctl reload sshd # reload the config files
```

```
systemctl stop sshd
```

```
systemctl status NetworkManager
```

```
echo $? # print out the last command value if 0 the last command submitted  
successfully if have any other value so it went through error
```

```
#There is some services depend on other services, Therefore if you stopped it The  
other services will start it again
```

```
systemctl list-dependencies sshd # list services The depend on sshd || under sshd
```

```
systemctl list-dependencies sshd --reverse # list service that sshd depend on |  
above sshd
```

```
# Any edit on the conf file must have reload | restart the service after
```





Configure remote access “SSH”

We will cover these Topics

- Configure secure command line service on remote systems

Configure remote access “SSH”

- Using port 22
- using asymmetric encryption (pub and priv keys)
- Note that there's no file .ssh on the client or the remote

1- Run this on the client machine

```
ssh-keygen # Generate the pub and private keys on the client machine
```

- The file that ends with .pub it's the public key that should be copied to the remote
- There's a file under the .ssh path on the remote server called known_hosts, it's automatically created when you login via SSH command on this server

```
ssh-copy-id mohamed@192.168.1.1 # 2- copy the pub key to the remote to automate authentications
```

```
# this will generate a file on the remote called .ssh/authorized_keys that have the pub key
```

```
# you can copy and paste the pub key content to the remote authorized_key file
```



Configure remote access “SSH”

- Note that .ssh dir created on the remote and have a file called authorized_key have your pub key.
- Now you can SSH on the remote server with out asking for password

```
ssh <username>@192.168.1.1 # remote on the server
```

```
sudo vim /etc/ssh/sshd_config # sshd service config file, any change in this file  
must reload the service
```

```
systemctl status sshd
```





Configure network interfaces and settings.

We will cover these Topics

- Configure network interfaces and settings

Configure network interfaces and settings.

- Get info

```
ip address show # print NIC info
```

```
ip addr show
```

```
ip a s
```

```
ip a s <NIC Name>
```

```
# get network statistics, receive and transfer packets
```

```
ip -s link show <NIC Name>
```

- There're two ways I prefer to set an IP

```
1- vim /etc/sysconfig/network-scripts/ifcfg-eth0 # put the Ip in the  
configuration file
```

```
nmcli con down <NIC Name> ; nmcli con up <NIC Name>
```

```
Systemctl restart NetworkManager
```

- 2 user mtui command

```
Nmtui # then down and up the NIC, and restart the service
```





Manage software using DNF

We will cover these Topics

- Download, install, update, and manage software packages from Red Hat and DNF package repositories.

Manage software using DNF

- You can download the software package file with .rpm extension then install it using `rpm -lvh <package name>` However each software package have dependencies and some dependencies have dependencies. The package manager handle all of this for you



Package Manager

- ▶ downloads, installs or updates existing software from a repository
- ▶ ensures the integrity and authenticity of the package
- ▶ manages and resolves all required dependencies
- ▶ knows where to put all the files in the Linux file system
- ▶ easy upgrading of the software



Manage software using DNF

```
dnf list installed # it's the same if you use yum
```

```
dnf list installed | grep -i "python"
```

```
rpm -qa | grep -i "python"
```

```
sudo vim /etc/yum.repos.d/redhat.repo
```

```
yum search <package>
```

```
yum remove -y <package>
```

```
yum install <package>
```

```
yum update <package>
```

```
yum info <package>
```

```
vim /var/log/yum.log
```

```
dnf update
```

```
yum history # or user dnf history info
```

```
yum history undo 3 # undo to action num 3
```



Resources

- [GPL](#)
- [Linux wiki](#)
- RHEL Book
- [RH124](#)
- [RHEL personal Document](#)
- [RH124 course -none official-](#)





Thank you

Thank you, I Really appreciate your attendance

Contact me:

Mohamed-ibrahim2021@outlook.com