LINUX

BASIC COMMANDS

ABSTRACT

UNIX is a computer Operating System which is capable of handling activities from multiple users at the same time.

Isha Ankush Malode
LINUX

History of linux

What is an Operating system

Operating system is the collection of programs that coordinates the operation of computer hardware & software

functions of operating system:

1)process management

2)memory management

3)Data management

4)I/O management

Arcitecthure of linux

Kernel:

kernel is a set of functions that makeup heart of an o/s it is used to provide application interface between programs & physical devices

services provided by kernel :

-controls execution of process

-scheduling process fairly for execution on cpu

-Allocating memory for an executing process

shell :

shell is an interface between human readable language & machine language

Multics project

Multics was started by mainframe GE 645 by the joint effort of AT&T bell labs general electrical Masachusetts Institute of Technology Multics was designed in Assembly Language in 1969 Multics project was dropped in 1969, AT & T redesigned multics and introduced new os thai is unics(Uniplex information & Computing system) it is written in 80% of "c" Language and 20% with assembly language by kenthomson & dennis ritchie .Later on totally rewritten in "c" language and renamed as unix(1973)

flavours of Unix

vendor	o/s
-----------	------------
AT&T,Bell labs	SYS III -sys V
SUN	Sun os - Solaris
SCO	Sco unix
IBM	Aix
SG	IRIX
HP	HP-Aux
BSD	free BSD linux

In 1988, AT&T shocked the unix community by purchasing a percentage of Sun microsystems which became a threat for other vendors

Quickly the other vendors form a group and named it as OSF(Open Software Foundation)

and former formed teir group and named it as UI (Unix International)

In 1990,Linuz torvalds a graduate student from Helskiny university designed unix like kernel 386 intel machine and gave it to OSF

Linux is bundled with many softwares from variuos distributors and it gave rise to many flavours of Linux

No of companies are providing tech support for linux o/s

RedHat

SUSE

MANDREKE

NOVEL

PUPPY LINUX

TURBOLINUX

SLACKWARE

UBUNTU

KNOPPIX

name of kernel in RhEl5 is v.m.Linuz 2.6.18-8.el5

Kernel image is initrd

installers name is anaconda-ks.cfg

default shell is /bin/bash

MBR - Master Boot Record

MBR's job is to locate compressed kernel & arrange the architecture logically

default MBR in linux is **GRUB** : Grand Unified Boot Loader in windows is **ntldr**

features of linux

1)Open Source(along with source code)

2)Multiuser& Multitasking

3)Enhanced Security (Inbuilt firewalls)

4)Reliability

5)GUI

Biggest servers on this earth are running on linux without restarting from last twelve years.

Accessing the command line

Date -Mon Oct 10 12:22:13 IST 2016

Date +%R - 12:23

Date +%x - Monday 10 October 2016

Head - This command is used to read first few lines of any given text.

Tail – This command is used to read last few lines of any given text.

By default it will show 10 lines.

If we want to see particular number line :- tail –n 3 /etc/passwd

Tail –f :- It will show current online information.

Wc :- This command is used to count lines, words, and characters.

Wc - l :- for lines

Wc -w :- for words

Wc –c :- for character

For Ex:- wc –l /etc/passwd.

History :- This command will show a list of previously executed command with command number.

MANAGING FILES FROM COMMAND LINE

/	The root directory, the top-level directory in the FHS. All other directories are subdirectories of root, which is always mounted on some partition. All directories that are not mounted on a separate partition are included in the root directory?s partition.
/bin	Essential command line utilities. Should not be mounted separately; otherwise, it could be difficult to get to these utilities when using a rescue disk.
/boot	Includes Linux startup files, including the Linux kernel. Can be small; 16MB is usually adequate for a typical modular kernel. If you use multiple kernels, such as for testing a kernel upgrade, increase the size of this partition accordingly.
/etc	Most basic configuration files.
/dev	Hardware and software device drivers for everything from floppy drives to terminals. Do not mount this directory on a separate partition.
/home	Home directories for almost every user.
/lib	Program libraries for the kernel and various command line utilities. Do not mount this directory on a separate partition.
/mnt	The mount point for removable media, including floppy drives, CD-ROMs, and Zip disks.
/opt	Applications such as WordPerfect or StarOffice.
/proc	Currently running kernel-related processes, including device assignments such as IRQ ports, I/O addresses, and DMA channels.
/root	The home directory of the root user.
/sbin	System administration commands. Don't mount this directory separately.
/tmp	Temporary files. By default, Red Hat Linux deletes all files in this directory periodically.
/usr	Small programs accessible to all users. Includes many system administration commands and utilities.
/var	Variable data, including log files and printer spools.

Pwd : - It display full path name of current location.

Ls :– lists directory contents for specified directory.

Ls –lR :– information about all subdirectorys.

Ls –l :– long listing format.

Ls –a :- All files includes hidden files.

Ls –R :- Recursive to include contents of all subdirectory.

Ls –al :- List current location with file.

Ls a* :- Begin with a.

Ls *a* :- Containing a.

Ls ???? :- Filename at least 4 character in lenth.

Q :- How many users are there to see ?

Ans :- ls –l /home

Cd – To change directory

Mkdir :- To create directories.

For ex. :- mkdir <dir name>

Cp :- It is use to copy one or more files to become new

For ex.:- cp source/destination/

Mv :- It rename files in same directory, or relocate files to new directory

For ex. :- mv newfile1 newfile2

Rm :- Remove files but not directories

Rm –r :- Delete directory and many subdirectory and files below it

Rmdir :- remove/delete empty directory

USERS AND GROUPS

Id :- This command is used to show information about current logged in user.

Ps :- To view process information.

Ps au :- ‘a’ option view all process with terminal, ‘u’ option first column shows username.

Userdel :- remove user from /etc/passwd.

Userdel –r :- It remove user and users home directory.

Useradd :- Adding user.

Usermod :- Modifies existing user.

To lock account :- usermod -L <username>

To unlock account :- usermod -U <username>

Chage -d 0 :- username will force a password update on next login.

Chage -l :- username will list usernames current settings.

Chage -E :- It will expire an account on specific day.

Q :- Determine date 90 days in future and set each of user account to expire on that day ?

Ans :- date -d “+90 days”

Chage -E 2016-06-16 username.

Q :- Change password policy for <user1> account to require new password every 15 days ?

Ans :- chage -M 15 <username>

Chage -l <username>

Q :- How many files are there in linux/unix ?

Ans :- Regular file – Readable, binary, image or compressed files.

Directory file.

Special file (5 subtypes in it)

- Block files (b).
- Character device file (c).
- Named pipe file or just pipe file (p).
- Symbolic link file (l).
- Socket file (s).

Block File (b) :- These files are hardware files, most of them are present in /dev.

How to create :- use fdisk command.

How can we list :- ls -l | grep ^b.

Character device file :- Provides a serial stream of i/o. (crw)

Pipe files in linux :- The other name of pipe is a “named” pipe, which is sometime called a FIFO refers to property that the order of bytes going in is the same coming out (prw)

Symbolic link files :- These are linked files to another files. They are either directory/regular file. The inode number for this file and its parent files are same.

Two types soft and hard link. (lrw)

Socket files in linux :- It is used for passing information between application for common purpose (srw).

Q :- How to find out desired type of file ?

Ans :- Use find command with –type option

Ex. For socket - Find / -type s

Linked - find / -type l

Explain PS command in output

To view process information

% cpu - How much of cpu the process is using.

%MEM – How much of memory the process is using.

ADDR – memory address of process.

C or cp – cpu usage and scheduling information.

COMMAND – name of process, including arguments, if any.

N1 – nice value.

F – Flags.

PID – process id number.

PPID – ID number of process's parent process.

PRI – Priority of process.

RSS – Real memory usage.

S or STAT – Process status code.

START or STIME – Time when process started.

Sz – Virtual memory usage.

TIME – Total cpu usage.

TT or TTY – Terminal associated with process.

UID or user – username of process owner.

WCHAN – memory address of event process is waiting for.

Explain passwd file in linux

“ /etc/passwd” file contain account information and look like this.

Smithj : x : 561:561:joe smith : /home /smithj :/bin/bash

- Username upto 8 character.
- An ‘x’ in password field. Password are stored in “/etc/shadow” file
- Numeric user id.
- Numeric group id.(usually group id will match user id)
- Full name of user (under 30 character)
- User’s home directory (ex. /home/smithj)
- User’s shell account “/bin/bash”.

The “/etc/shadow” file contain password.

Smithj : Ep6mckr0lchf :10063 : 0 : 999999 : 7 :::

Username (8 character)

Password (13 character) a blank entry (eg ::) indicate is not required to log in (usually a bad idea) and a “x” entry (eg :x:) indicate account has been disabled.

The number of days since password was last change.

The number of days before password may be changed (0 indicate it may be changed at any time)

The number of days after which password must be changed. (999999 indicate user can keep his password unchanged for many year).

The number of days to warn user of an expiring password.

Q :- How many shells are there in linux ?

Ans :- sh, bash, csh and tcsh, ksh

$ - Bourne, korn and Bash shells.

% - C shell

CREATING, VIEWING, AND EDITING TEXT FILES

in vi editor we have 3 modes :

1)command mode

2)insert mode

3)execute mode

in command mode we can copy,paste,delete,undo,redo,move,save&quit

in insert mode we can only edit data

in execute mode we can set line numbers,delete line numbers,save&quit,quit with out saving,substitute data

when ever we type vi <filename>

if the file is there it will edit the existing file

if the file is not there it will create a new file

by using vi <filename> it will enter into command mode

to go to insert mode from command mode

i: insert text at current cursor position

I: insert text at the beginning of cursor line

a: append text after cursor position

A: append text at the end of cursor line

o: create a new line below cursor line and append text

O: create a new line above cursor line and append text

s: it removes cursor presented character and append data to move from insert mode to command mode press "esc key"

command mode :

----

dd: to delete a line

5dd : to delete five lines

yy: to copy a line

5yy: to copy five lines

p: paste

10p : paste 10 times

u:undo

ctrl+r :redo

shift+g or G : to move last line of a file

gg :to move first line of a file

10g :to move 10th line of a file

shift+zz : save&quit

to move into execute mode from command mode press shift+:

execute mode

----

q: quit with out saving

q!: quit with out saving forcefully

wq: save&quit

x: save&quit

wq!: save&quit forcefully

:/<word> : to serach for a particular word

ex: /sun : searching for a word sun in the file

:15 :to jump into 15th line of a file

set nu: to set line numbers for a file

set nonu: to remove line numbers of a file

substitution

----

syntax: <begingline>,<endingline> s /<oldchar>/<newchar>/g

ex: if i want to change LABEL as SMS in /etc/fstab

1,$ s/LABEL/SMS/g

here 1,$ indicates from first line to lastline

s for substitute

LABEL is old charcter

SMS is new character

g for grouping

note: to set automatic line numbers

----

crate a hidden file as follows in /root

vi .vimrc

(type) set nu

save& quit

to create hidden files in linux use . at the begining of filename

ex: cat > .sun

now .sun is a hidden file

clear : used to clear the screen or use ctrl + l

to get more help for vi editor

type vimtutor

CONTROLLING ACCESS TO FILES WITH LINUX SYSTEM PERMISSION

U, g, o - (user, group, other)

+, -, = - (add, remove, set exactly)

R, w, x – (read, write, execute)

R= 4, w = 2, x= 1

- **Chmod -** for modification

Removing read and write permission for group and other file.

For ex. Chmod go – rw file 1

Add execute permission for everyone of file 2

Ans:- chmod a+x file 2

To set read, write, execute permission

Ans:- chmod all 750

- **Chown** – changing file / directory user or group

For ex. Change ownership of foodir to visitor and group to guest

Ans :- chown visitor : guests foodir

- **Chgrp** - same as chown
- **Umask** – to set permission by default on new file.

MONITORING AND MANAGING LINUX PROCESS

Running jobs in background. Any command can be started in background by appending (&) to command line.

Ps j – It will display job information including initial command

Jcpu – resource consumed by current jobs

Pcpu – current foreground process cpu consumption

To kill process forcefully :– kill -9 PID

To kill process safely :– kill -15 PID

Pkill command like killall can signal multiple processes.

- Top program is a dynamic view of system's process

GREP COMMAND

- **grep**, which stands for "global regular expression print," processes text line by line and prints any lines which match a specified pattern.
- To connect two commands together, so that the output from one program becomes the input of next program
- The grep program searches a file that have a certain pattern
- The grep searches the named input files
- The basic usage of grep command is to search for a specific string in the specified file as shown below.
- Syntax:
- grep "literal_string" filename

Ex. # grep "this" demo_file

o/p. :- this line is the 1st lower case line in this file

Two lines above this line is empty.

And this is the last line.

- Checking for the given string in multiple files.

Syntax:

```
grep "string" FILE_PATTERN
```

This is also a basic usage of grep command. For this example, let us copy the demo_file to demo_file1. The grep output will also include the file name in front of the line that matched the specific pattern as shown below. When the Linux shell sees the meta character, it does the expansion and gives all the files as input to grep.

```
Ex. :-  # cp demo_file demo_file1
        #  grep "this" demo_*
o/p :- demo_file:this line is the 1st lower case line in this file
demo_file:Twoo lines above this line is empty.
demo_file:And this is the last line.
demo_file1:this line is the 1st lower case line in this file
demo_file1:Twoo lines above this line is empty.
demo_file1:And this is the last line.
```

Q.:- How do u display single user from shadow file

Ans.:- # grep <username> /etc/shadow

- Case insensitive search using grep -i

Syntax:

```
grep -i "string" FILE
```

This is also a basic usage of the grep. This searches for the given string/pattern case insensitively. So it matches all the words such as “the”, “THE” and “The” case insensitively

```
Ex.:- #  grep -i "the" demo_file
 o/p :- THIS LINE IS THE 1ST UPPER CASE LINE IN THIS FILE
this line is the 1st lower case line in this file
```

```
This Line Has All Its First Character Of The Word With Upper Case

And this is the last line.
```

- Match regular expression in files

Syntax:

```
grep "REGEX" filename
```

This is a very powerful feature, if you can use use regular expression effectively. In the following example, it searches for all the pattern that starts with “lines” and ends with “empty” with anything in-between. i.e To search “lines[anything in-between]empty” in the demo_file.

```
Ex. #  grep "lines.*empty" demo_file

o/p :- Twoo lines above this line is empty.
```

- Checking for full words, not for sub-strings using grep -w

If you want to search for a word, and to avoid it to match the substrings use -w option. Just doing out a normal search will show out all the lines.

The following example is the regular grep where it is searching for “is”. When you search for “is”, without any option it will show out “is”, “his”, “this” and everything which has the substring “is”.

```
Ex.:- grep -i "is" demo_file

O/p :- THIS LINE IS THE 1ST UPPER CASE LINE IN THIS FILE

this line is the 1st lower case line in this file

This Line Has All Its First Character Of The Word With Upper Case

Twoo lines above this line is empty.

And this is the last line.
```

- The following example is the WORD grep where it is searching only for the word “is”. Please note that this output does not contain the line “This Line Has All Its First Character Of The Word With Upper Case”, even though “is” is there in the “This”, as the following is looking only for the word “is” and not for “this”.

```
# grep -iw "is" demo_file

O/P :- THIS LINE IS THE 1ST UPPER CASE LINE IN THIS FILE

this line is the 1st lower case line in this file

Twoo lines above this line is empty.

And this is the last line.

#grep ‘^user1’  <filename>

o/p :- user1:x:1017:1017::/home/user1:/bin/bash
```

```
#  grep 'nologin$' /etc/passwd

o/p :- bin:x:1:1:bin:/bin:/sbin/nologin

daemon:x:2:2:daemon:/sbin:/sbin/nologin

adm:x:3:4:adm:/var/adm:/sbin/nologin

lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

mail:x:8:12:mail:/var/spool/mail:/sbin/nologin

operator:x:11:0:operator:/root:/sbin/nologin

games:x:12:100:games:/usr/games:/sbin/nologin

ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
```

AWK COMMAND

Awk Introduction and Printing Operations

Awk is a programming language which allows easy manipulation of structured data and the generation of formatted reports. Awk stands for the names of its authors “Aho, Weinberger, and Kernighan”

The Awk is mostly used for pattern scanning and processing. It searches one or more files to see if they contain lines that matches with the specified patterns and then perform associated actions.

Some of the key features of Awk are:

Awk views a text file as records and fields.

Like common programming language, Awk has variables, conditionals and loops

Awk has arithmetic and string operators.

Awk can generate formatted reports

Awk reads from a file or from its standard input, and outputs to its standard output. Awk does not get along with non-text files.

Ex. :- By default Awk prints every line from the file.

```
#  awk '{print;}' </etc/passwd
```

```
o/p :-  root:x:0:0:root:/root:/bin/bash

bin:x:1:1:bin:/bin:/sbin/nologin

daemon:x:2:2:daemon:/sbin:/sbin/nologin

adm:x:3:4:adm:/var/adm:/sbin/nologin

lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

sync:x:5:0:sync:/sbin:/bin/sync

shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown

halt:x:7:0:halt:/sbin:/sbin/halt

mail:x:8:12:mail:/var/spool/mail:/sbin/nologin

operator:x:11:0:operator:/root:/sbin/nologin

games:x:12:100:games:/usr/games:/sbin/nologin
```

- Print only specific field.

Awk has number of built in variables. For each record i.e line, it splits the record delimited by whitespace character by default and stores it in the $n variables. If the line has 4 words, it will be stored in $1, $2, $3 and $4. $0 represents whole line. NF is a built in variable which represents total number of fields in a record.

Ex.:- `awk -F: '{print $4}' </etc/passwd`

```
o/p :-  0

1
```

```
2
4
7
0
0
0
12
0
100
```

- Ex.:- awk '{print $2,$5;}'</etc/passwd

```
o/p :- ser:/var/ftp:/sbin/nologin

message
for
mDNS/DNS-SD
IPv4LL

SSH:/var/empty/sshd:/sbin/nologin
Malode:/home/isha.malode:/bin/bash
```

- Ex.:- awk '{print $2,$NF;}'</etc/passwd

```
o/p :- root:x:0:0:root:/root:/bin/bash
 bin:x:1:1:bin:/bin:/sbin/nologin
 daemon:x:2:2:daemon:/sbin:/sbin/nologin
 adm:x:3:4:adm:/var/adm:/sbin/nologin
 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
 sync:x:5:0:sync:/sbin:/bin/sync
 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
 halt:x:7:0:halt:/sbin:/sbin/halt
 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
 operator:x:11:0:operator:/root:/sbin/nologin
```

- Find the employees who has employee id greater than 200

```
Ex.:- awk '$1 >200'</etc/passwd
o/p :- isha.malode:x:1000:1000:Isha Malode:/home/isha.malode:/bin/bash
prince:x:1001:1001::/home/prince:/bin/bash
bob:x:1002:1002::/home/bob:/bin/bash
juliet:x:1003:1003::/home/juliet:/bin/bash
```

SED COMMAND

Replacing or substituting string

Sed command is mostly used to replace the text in a file

```
Ex. :- #  sed -n '1p' </etc/passwd
o/p :- root:x:0:0:root:/root:/bin/bash
```

The tr Command

The tr command is used to translate specified characters into other characters or to delete them.

In contrast to many command line programs, tr does not accept file names as arguments (i.e., input data). Instead, it only accepts inputs via standard input, (i.e., from the keyboard) or from the output of other programs via redirection.

The general syntax of tr is

```
tr [options] set1 [set2]
```

Convert lower case to upper case

The following tr command is used to convert the lower case to upper case

```
#  tr abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ
 o/p :- thegeekstuff
        THEGEEKSTUFF
```

The following command will also convert lower case to upper case

```
#  tr [:lower:] [:upper:]
 o/p:- thegeekstuff
       THEGEEKSTUFF
```

Translate braces into parenthesis

You can also translate from and to a file. In this example we will translate braces in a file with parenthesis.

```
# tr '{}' '()' < inputfile > outputfile
```

The above command will read each character from “inputfile”, translate if it is a brace, and write the output in “outputfile”.

3. Translate white-space to tabs

The following command will translate all the white-space to tabs

```
# echo "This is for testing" | tr [:space:] '\t'
o/p:- This	is	for	testing
```

diff command

diff analyzes two files and prints the lines that are different. Essentially, it outputs a set of instructions for how to change one file in order to make it identical to the second file.

It does not actually change the files; however, it can optionally generate a script (with the -e option) for the program ed (or ex which can be used to apply the changes.

Ex.:- Let's say we have two files, file1.txt and file2.txt.

If file1.txt contains the following four lines of text:

```
I need to buy apples.

I need to run the laundry.

I need to wash the dog.

I need to get the car detailed.
```

...and file2.txt contains these four lines:

```
I need to buy apples.

I need to do the laundry.

I need to wash the car.

I need to get the dog detailed.
```

...then we can use diff to automatically display for us which lines differ between the two files with this command:

```
diff file1.txt file2.txt
```

...and the output will be:

```
2,4c2,4

< I need to run the laundry.

< I need to wash the dog.

< I need to get the car detailed.

---

> I need to do the laundry.

> I need to wash the car.

> I need to get the dog detailed.
```

- another example. Let's say our two files look like this:

file1.txt:

```
I need to go to the store.

I need to buy some apples.

When I get home, I'll wash the dog.
```

file2.txt:

```
I need to go to the store.

I need to buy some apples.

Oh yeah, I also need to buy grated cheese.

When I get home, I'll wash the dog.
```

```
diff file1.txt file2.txt
```

Output:

```
2a3

> Oh yeah, I also need to buy grated cheese.
```

Cut command

Linux command cut is used for text processing. You can use this command to extract portion of text from a file by selecting columns.

- Ex.:- Select Column of Characters

To extract only a desired column from a file use -c option. The following example displays 2nd character from each line of a file test.txt

```
# cut -c2 test.txt

a

p

s
```

- Select Column of Characters using Range

Range of characters can also be extracted from a file by specifying start and end position delimited with -. The following example extracts first 3 characters of each line from a file called test.txt

```
$ cut -c1-3 test.txt
cat
cp
ls
```

uniq command

- test file is used in some of the example to understand how uniq command works.

```
# cat test
aa
aa
bb
bb
bb
xx
```

1. Basic Usage

Syntax:

```
$ uniq [-options]
```

For example, when uniq command is run without any option, it removes duplicate lines and displays unique lines as shown below.

```
$ uniq test
aa
bb
xx
```

- Count Number of Occurrences using -c option

This option is to count occurrence of lines in file.

```
$ uniq -c test
   2 aa
   3 bb
  1 xx
```

- Print only Duplicate Lines using -d option

This option is to print only duplicate repeated lines in file. As you see below, this didn’t display the line “xx”, as it is not duplicate in the test file.

```
$ uniq -d test
aa
bb
```

sort command

sort is a simple and very useful command which will rearrange the lines in a text file so that they are sorted, numerically and alphabetically.

Ex.:-let's say you have a file, data.txt, which contains the following ASCII text:

```
apples
oranges
```

```
pears

kiwis

bananas
```

To sort the lines in this file alphabetically, use the following command:

```
sort data.txt
```

...which will produce the following output:

```
apples

bananas

kiwis

oranges

pears
```

CONTROLLING SERVICES AND DAEMONS

Daemons are process that wait or run in the background performing various task.

1. Systemctl - query for all (to verify system startup)
2. Query state of only service units

Systemctl -- type= service

Ex. :- systemctl status sshd.service → to see status of particular service

Systemctl stop sshd.service → To stop the particular service

Systemctl start sshd.service → To start particular service

Systemctl restart sshd.service → To restart the service

Systemctl reload sshd.service → To reload the service

- While reloading service the PID will change.

3. To see active and enabled status

Systemctl is -active sshd

Systemctl is –enabled sshd

Particular service is enabled or not ?

Netstat -napl | grep <port no.>

In /etc services → To see all ports

4. List active state of all loaded units, limit type of unit

Systemctl list-units --type = service

5. View enabled and disabled setting for all units

Systemctl list – unit – files --type = service

6. View only failed service

Systemctl --failed

7. List all socket units, active and inactive on system

Systemctl list-units --type=socket

8. To see listed daemons are running

Ps -p PID

9. To disable and enable services.

Systemctl enable sshd.service

Systemctl disable sshd.service

**** Disabling service does not stop service ****

CONFIGURING AND SECURING OPENSSH SERVICE

W → It will display list of user currently logged in pc.

This is especially useful to show which users are logged in, using ssh from which remote location.

To store password

/etc /sysconfig/network   OR

/etc /hosts

Step 1. Ssh-keygen -t rsa

**** ssh file should be there, if not then create****

Mkdir .ssh

Ls -l

Now try to login into machine with ssh <ip>

IN second machine == Open vim authorized _keys

ANALYZING AND STORING LOGS

/var /log /messages → Most syslog messages are logged here.

/var /log/ secure → The log file for security and authentication related messages and error.

/varr /log /maillog →The log file with mail server related messages.

/var /log /cron → The log file related to periodically executed task.

/var /log / boot.log → Messages related to system startup are logged here.

Journal stored in /run/log

Journalctl command shows the full system. journalctl, starting with oldest log entry when run as root user.

Journalctl -p err → command to only list any log entry of priority error or above.

Vim → /etc /logrotate.d

For modification → /etc /logrotate.conf.

MANAGING RED HAT ENTERPRISE LINUX NETWORKING

- **Displaying IP address**

*/sbin /ip command is used to show device and address information

→ ip addr show eth0

*To see statistics about network performance

The received (Rx) and transmitted (Tx) packets.

→ ip –s link show eth0

*Routing information

→ ip route

*To trace path to remote host

→tracepath <hostname>

Command used to display socket statistics.

ss -ta → show socket status.

ss -n → show number instead of names for interface and ports.

ss -t → Show tcp socket.

ss -u → show udp socket.

ss -l → show only listening socket.

ss -a → show all (listening and established) socket.

ss –p → show process using sockets.

ss –lt → Display listening TCP socket on local system.

* CONFIGURING NETWORK WITH NMCLI *

- To display list of all connection

```
# nmcli  con show
```

- To see device details and status

```
# nmcli dev status
```

```
# nmcli dev show eth0
```

CREATING NETWORK CONNECTIONS WITH NMCLI

- Define a new connection named “default” which will autoconnect as an Ethernet connection on the eth0 device using DHCP

```
# nmcli con add con-name “default” type Ethernet ifname eth0
```

- Create a new connection named “static” and specify IP addr and gateway. Do not autoconnect.

```
# nmcli con add con-name “static”  ifname eth0 autoconnect no type Ethernet ip4.
```

- The system will autoconnect with DHCP connection at boot change to static connection

```
# nmcli  con up “static”
```

- Change back to DHCP connection

```
#   nmcli con up “default”
```

SUMMARY OF nmcli COMMANDS

nmcli dev status → list all devices

nmcli con show → list all connections

nmcli con up <ID>→ Activate a connection

nmcli con down <ID> → Deactivate a connection

nmcli dev di <DEV> → bring down an interface and temp disable autoconnect nmcli net off → Disable all managed interfaces

nmcli con add → Add new connection

nmcli con mod <ID> → modify connection

nmcli con del <ID> → delete connection

# hostname → this command display or temp modifies the system’s fully qualified host name

- Display host name status

# hostnamectl status

- To change permanent hostname → vi /etc/hostname (in 7th)
  /etc /sysconfig /network (in 6th)

- To see the version

# /etc /redhat- release

- To change host name and host name configuration file

# sudo hostnamectl set – hostname

- To add router or gateway

# ip route add <ip> dev eth0

ARCHIVING AND COPYING FILES BETWEEN SYSTEMS

- **Archives files and directories with tar**

c → create an archieve

t → list the content of an archieve

x → extract an archieve

f → file name

- **To create compressed tar archieve**

z → For gzip compression (filename.tar.gz)

j → For bzip2 compression (filename.tar.bz2)

J → For xz compression (filename.tar.xz)

Q :- Create the archie with named archive.tar with contents of file1, file2, file3 in home directory

Ans :- tar cf archive.tar file1, file2, file3

Q :- Create tar archive /root/etc.tar with /etc

Ans :- tar cvf /root/etc.tar /etc

Compression of file

Ex. :- 1. Create directory

```
#mkdir test/
# tar  -zcvf test.tar.gz  test1 tom tomjerry
```

Uncompressed

```
# tar –xvzf  test.tar.gz
```

- **To list contents of a tar archive**

```
# tar tf /root/etc.tar
```

COPYING FILES BETWEEN SYSTEMS SECURELY

- **For copying file**

```
# scp  -r <filename>  root @ <ip> : /root
```

Ex. :- scp -r abcd root@ 172.24.2.52: /root

- **For transferring file remotely**

```
# sftp
```

- **Synchronizing files between systems securely**

```
# rsync
```

INSTALLING AND UPDATING SOFTWARE PACKAGES

Yum → It is used to list repositories, packages group

```
# yum  repolist
# yum list yum*
# yum list installed
# yum grouplist
```

FOR INSTALLATION

```
# yum install <packagename>
```

FOR UPDATION

```
# yum update <packagename>
```

FOR REMOVE

```
# yum remove <packagename>
```

Yum group install command will install group which will install its mandatory and default packages

```
# yum group install
```

VIEWING TRANSACTION HISTORY

```
# tail -5 /var/log/yum.log
```

A summary of install and remove transaction can be viewed with yum history

```
# yum history
```

Search for package by keyword

```
# yum search  <keyword>
```

To view all available repositories

```
# yum repolist all
```

Display information about package

```
# rpm  -q  -c  NAME
```

List all files included in package

```
# rpm  -q  -l NAME
```

List config files included in package

```
# rpm  -q  -c  NAME
```

Show a short summary of reason for a new package release

```
# rpm  -q  --changelog  NAME
```

Display the shell scripts included in package

```
# rpm  -q  --scripts  NAME
```

ACCESSING LINUX FILE SYSTEM

Overview about file system mount points and amount of free space available

```
# df

# df  -h

# df  -H
```

For more detailed information about space used by certain directory tree

```
# du /root
```

Use to discover the UUID of newly added partition on server x

```
# blkid
```

To create mount point /mnt/newspace on server

```
# mkdir /mnt /newspace
```

Mount file system by UUID on /mnt/newspace directory of server x machine

```
# mount UUID= ‘  ‘  /mnt/newspace
```

Change to /mnt/newspace directory on server

```
# cd /mnt/newspace
```

Create a new directory, /mnt/newspace/newdir on server

```
# mkdir newdir
```

Create a new empty file, /mnt/newspace/newdir/newfile, on server

```
# touch newdir /newfile
```

Unmount the file system mounted on /mnt/newspace directory on servers

```
# umount  /mnt/newspace
```

INSTALLATION AND CONFIGURING OF APACHE

```
# Yum install httpd*
```

For configuration

```
# vim /etc /httpd /conf /httpd.conf
```

For basic installation modification not required

Create a file in /var/www/html

```
# cd /var/www/html
# vim index.html
= = = = =
# service httpd restart
```

Server configured.

Go to firefox and check. OR

We can use e-link for that

Install e-link

```
# yum  install  elinks
```

If error comes check configuration file

```
# view /etc /httpd /conf /httpd.conf
```

QUESTIONS AND ANSWERS

Q :- How do you display single user from shadow file ?

```
Ans.:- # vi /etc/shadow
         # grep  <username>  /etc/shadow
```

Q :- View the contents of a directory : A directory may contains visible and invisible files with different file permissions.

```
Ans: #ls –al
o/p: drw-r--r--.  2 root root     18 Jun 17 14:46 .test
```

Q :- Viewing system partitions and used space

```
Ans: #fdisk –l
o/p: Disk /dev/sda: 37.6 GB, 37580963840 bytes, 73400320 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000b08b1
```

Q :- Know your machine name, OS and Kernel

```
Ans:  #hostnamectl status
o/p:  Static hostname: localhost.localdomain
        Icon name: computer-vm
          Chassis: vm
        Machine ID: ef67461d095844519f29acc21b90cf35
           Boot ID: 3b427f7acc18458991b51d0a9acf2577
    Virtualization: microsoft
  Operating System: CentOS Linux 7 (Core)
       CPE OS Name: cpe:/o:centos:centos:7
            Kernel: Linux 3.10.0-123.el7.x86_64
      Architecture: x86-64
```

Q :- Viewing history

Ans: #history

```
o/p:  252  nmcli con show
  253  nmcli con show --active
  254  nmcli con show "static-eth0"
  255  nmcli dev status
  256  nmcli dev show eth0
  257  nmcli dev status
```

Q :- Being root from your user

Ans: #su

Q :- create Directory

Ans: #mkdir

```
o/p: isha2              snap2.jpg
```

Q :- create Files

Ans: # touch, or #cat

```
o/p:  -rw-r--r--. 1 root root      0 Sep 12 17:45 nzfile1
```

Q :- Changing the file permission

Ans: #chmod

```
o/p: -rw-r--r--. 1 root root      0 Jul 14 15:46 test1
chmod go-r test1
-rw-------. 1 root root      0 Jul 14 15:46 test1
```

Q :- Install, Update and maintain Packages

Ans: #using yum

o/p: yum install wget

```
Loaded plugins: fastestmirror
base                                          | 3.6 kB     00:00
extras                                        | 3.4 kB     00:00
updates                                        | 3.4 kB     00:00
Loading mirror speeds from cached hostfile
* base: mirror.fibergrid.in
* extras: mirror.fibergrid.in
* updates: mirror.fibergrid.in
Resolving Dependencies
--> Running transaction check
---> Package wget.x86_64 0:1.14-10.el7_0.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

===================================================================================
Package       Arch          Version                Repository      Size
===================================================================================
Installing:
wget         x86_64         1.14-10.el7_0.1         base          545 k

Transaction Summary
===================================================================================
Install  1 Package
```

Q :- Uncompressing a file

Ans:#tar –zxvf file. tar.gz

o/p:

Q :- See current date, time and calendar

```
Ans: #date +%x
o/p:  Monday 12 September 2016
```

Q :- Print contents of a file on command line

```
Ans:# cat
o/p:  cat> nzfile1
hii
```

Q :- Copy and Move

```
Ans:# cp and mv
o/p:   mv sum.sh moon.sh
   ls moon.sh
moon.sh
```

Q :- See the working directory for easy navigation

```
Ans: #pwd
o/p:  ]# pwd
/root
```

Q :- Change the working directory, etc…

```
Ans: #cd
```

Q :- List out all files and directories in a given directory for ex: on /home

```
Ans: # ls –l /home
o/p: /home:
abc    ateam       bob     elvis   isha.malode  prince  sspade
alice  ateam-text  dolly   hamlet  jerry        reba    user1
andy   bboop       dtracy  isha    juliet       romeo
```

Q :- Finding a file in a given directory

```
Ans: #ls <filename>
o/p:  ls nzfile1
nzfile1
```

Q :- Searching a file with the given keywords

```
Ans: #grep <word> <filename>

o/p:  grep "number" pali.sh

echo -n "Enter number : "

# store number in reverse order

# store original number

    # store previous number and current digit in reverse
```

Q :- See the current running processes

```
Ans: #ps

o/p:    PID TTY          TIME CMD

6426 pts/0    00:00:00 bash

6823 pts/0    00:00:00 su

6843 pts/0    00:00:00 su

6847 pts/0    00:00:00 bash

6935 pts/0    00:00:00 su
```

Q :- Kill a running process

```
Ans: # kill -9 pid
```

Q :- Starting, Ending, Restarting a service

```
Ans: # systemctl status<service name>.service

            Systemctl start <service name>.service

            Systemctl stop <service name>.service
```

Q :- Making and removing of aliases

```
Ans: #unalias <name>  for removing

          # alias –p <name>

o/p: # alias p="pwd"

[root@localhost ~]# p

/root
```

Q :- View the disk and space usages

```
Ans:# df –h and du –s
```

```
O/p:  Filesystem             Size  Used Avail Use% Mounted on
/dev/mapper/centos-root   33G  1.4G   32G   5% /
devtmpfs                 488M     0  488M   0% /dev
tmpfs                    494M     0  494M   0% /dev/shm
tmpfs                    494M  264K  494M   1% /run
tmpfs                    494M     0  494M   0% /sys/fs/cgroup
/dev/sda1                497M   96M  402M  20% /boot
```

Q :- Removing a file and/or directory

Ans: #rm –r

```
o/p: drwxr-xr-x. 2 root root      6 Jun 17 10:39 family
drwxr-xr-x. 2 root root      6 Jun 17 10:39 friend
o/p:  drwxr-xr-x. 2 root root      6 Jun 17 10:39 friend
drwxr-xr-x. 5 root root     45 Jun  7 12:53 glob
```

Q :- Changing password of on-self and other’s, if you are root.

Ans:# passwd

```
o/p: passwd
Changing password for user root.
New password:
```

Q :- Compare two files

Ans: #diff file1 file2

```
o/p: [root@localhost ~]# diff vyankat tom
1,66d0
<
< 1.Insert Mode
< 2.Escape Mode
< 3.Colon Mode
```

Q :- Download a file, the Linux way (wget)

Ans:#wget <file name>

```
o/p:  ]# wget http://website.com/files/file.zip
```

```
--2016-09-12 18:39:14--  http://website.com/files/file.zip
Resolving website.com (website.com)... 65.61.198.201
Connecting to website.com (website.com)|65.61.198.201|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://www.website.com/files/file.zip [following]
--2016-09-12 18:39:15--  http://www.website.com/files/file.zip
Resolving www.website.com (www.website.com)... 65.61.198.201
Reusing existing connection to website.com:80.
```

Q :- Mount a block / partition / external HDD

Ans:# mount

Q :-Configuring Network Interface

Ans:# ifconfig

Q :- Viewing custom Network Related information

Ans:#ip addr show

```
o/p:   1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
```

Q :- Digging DNS (how to check IP address of specific URL. for ex: to get IP address of www.google.com)

Ans:#nslookup google.com

```
o/p:  Server: 192.168.0.105
Address: 192.168.0.105#74
```

Q :- Knowing Your System uptime

Ans:# uptime

```
o/p:  17:51:36 up 3 days,  4:48,  1 user,  load average: 0.00, 0.01, 0.05
```

Q :- Renaming a file

Ans:# mv

o/p: mv newfile3 newfile6

o/p: -rw-rw----. 1 root root   0 Jul 15 16:54 newfile6

Q :- Text File editors like vi, emacs and nano

Ans:# vim /etc/exports (for configuring nfs)

O/p:

Q :- Copy folder from one location to another location with all subdirectories and files

Ans:# cp –R source/destination/

o/p: cp -R video newmoon

```
[root@localhost ~]# ls newmoon
watched
```

Q :- check free and available memory

Ans:#free

o/p:

```
     total      used      free    shared    buffers     cached
Mem:      1010924     889356     121568       240        20     174784
-/+ buffers/cache:     714552     296372
Swap:     2129916      1732    2128184
```

Ext2

- Ext2 stands for second extended file system.
- It was introduced in 1993. Developed by Rémy Card.
- This was developed to overcome the limitation of the original ext file system.
- Ext2 does not have journaling feature.
- On flash drives, usb drives, ext2 is recommended, as it doesn't need to do the over head of journaling.
- Maximum individual file size can be from 16 GB to 2 TB
- Overall ext2 file system size can be from 2 TB to 32 TB

Ext3

- Ext3 stands for third extended file system.
- It was introduced in 2001. Developed by Stephen Tweedie.
- Starting from Linux Kernel 2.4.15 ext3 was available.
- The main benefit of ext3 is that it allows journaling.
- Journaling has a dedicated area in the file system, where all the changes are tracked. When the system crashes, the possibility of file system corruption is less because of journaling.
- Maximum individual file size can be from 16 GB to 2 TB
- Overall ext3 file system size can be from 2 TB to 32 TB
- There are three types of journaling available in ext3 file system.
  - Journal – Metadata and content are saved in the journal.
  - Ordered – Only metadata is saved in the journal. Metadata are journaled only after writing the content to disk. This is the default.
  - Writeback – Only metadata is saved in the journal. Metadata might be journaled either before or after the content is written to the disk.
- You can convert a ext2 file system to ext3 file system directly (without backup/restore).

Ext4

- Ext4 stands for fourth extended file system.
- It was introduced in 2008.
- Starting from Linux Kernel 2.6.19 ext4 was available.
- Supports huge individual file size and overall file system size.
- Maximum individual file size can be from 16 GB to 16 TB

- Overall maximum ext4 file system size is 1 EB (exabyte). 1 EB = 1024 PB (petabyte). 1 PB = 1024 TB (terabyte).
- Directory can contain a maximum of 64,000 subdirectories (as opposed to 32,000 in ext3)
- You can also mount an existing ext3 fs as ext4 fs (without having to upgrade it).
- Several other new features are introduced in ext4: multiblock allocation, delayed allocation, journal checksum. fast fsck, etc. All you need to know is that these new features have improved the performance and reliability of the filesystem when compared to ext3.
- In ext4, you also have the option of turning the journaling feature "off".

Here is a short list of advantages of Ext4 over Ext3:

- extents (reduce overhead for large files, reduce fragmentation and improve performance)
- flexible block groups (fast fsck)
- support for huge files, huge total filesystem size
- more subdirectories
- journal checksumming

Overall, Ext4 provides better performance, reliability and scalability.