**Red Hat Certification System Administrator**

# Linux Architecture

The kernel is linked directly to the hardware components. The shell is a higher level than the kernel and we have the highest level that is the closest to the user which is the user interface.

# Check status of services

**systemctl** **status** sshd : This command will check the status of the service sshd which is the service responsible for remote connections on servers (OpenSSH)

# Check ip address

**ip addr** : To see the interfaces and ip addresses.

# Relative path vs Absolute path

Absolute path is when we start from the root directory and pass to another directory (/home/elie)

♣ Commands are case sensitive

# Open a file and write in it

**cat >** firstFile

Here we can write some text but if there were some info in the file before it will be overridden.

**cat >>** firstFile

Here we will not override we will just add.

**cat -n >>** firstFile

Will add the number for each line

# Traverse between directories

**cd -** : Go to the last directory.

**cd ~** : Go to home directory.

# Work with directories

**mkdir -p** 1/2/3/4 : Here we created 4 direcotires nested in each other so the the directory 1 is the first parent.

**mkdir** {2008..2021}-{01..12}

**rmdir 6/ :** Removes the directory 6 but in condition that the directory is empty otherwise we should add the parameter **-f**

**rmdir -rf** 4/ : ‘r’ means recursif so here it will delete nested directories. We can also add the **-i** that will ask for confirmation before deleting.

# Check the history of commands

**history** : Show all executed commands and each command will be listed on a line with the number of the command, let’s say I want to execute the command of the line 71 I can do : **!71**

**!!** : Will execute the last command

**history -c** : Will delete the history

# Check info about machine and os

**uname -a**

# File ownership

Owner : Group : World : Path

4 read (r)

2 write (w)

1 execute (x)

7 = 4+2+1

6 = 4+2

**ls -l** file ou **ll** file:

- rw- rw- r--

The - determines a file or directory or link.

**chmod** 554 file : Will give the following access rules for the file.

**chmod +x** : Add execute permission to owner and group of file.

**chmod u+rw** : Add r/w to user.

**chmod -R+x** : Add execute permissions to files and directories permissions recursively.

**chmod g-rw** : Removes read and write for group

**chmod ugo+rwx** : Adds r/w/x for user,group and others.

**chmod -R 770** Downloads/ : Changes recursively the permissions for this directory.

**chown** root file : Change the owner group of file so root is owner of group

**chown** root : root file : Changes the owner and group of file.

**chgrp** root : file : Change the group to root for the file.

# Work with files

**touch** file-‘date +%d-%M-%Y’

Result : file-4-12-2021

**touch** file{1..100}.txt

Here we will create files going from 1 to 100

# Copy some files

**cp** .txt Downloads

Will copy all text files to the Downloads directory.

**cp -pv** file Downloads/

The ‘-p’ is used to preserve parameters so for example the modification date of the copied file will be same as before copying. The ‘-v’ means verbose which will show us a sentence explaining what happened.

**cp -R** Desktop/ Downloads/

The ‘-R’ is used to copy a directory.

**scp** backup.tar.gz root@192.168.2.140:/opt/ : Will copy file to opt directory on the remote server.

# Move some files

**mv** file othernamefile : Renames the file.

**mv** file Downloads/ : Moves to Downloads.

# List some files

**ls -m** /home/elie : ‘-m’ is used to list files separated with a comma. Usually used when exporting.

**ls -r** : list subdirectories also.

**less** : Display file content 1 page at a time

# Create a link

**ln -s** Kenlm/LICENSE : Create link to that file with same name

# Compare two files

**diff** file1 file2 : Verify if two files are equal

# Output some data

**head** : By default, prints first 10 lines of file

**head -n** 15 : First 15 lines

**tail** : last 10 lines

**tail -n** 20 /var/log/messages **>** /home/aravi/logfile

# Arithmetic expression

**echo $**((7+3)) : Will output 10

# Brace expansion

**echo** abc{elie,joe}xyz : Will output 2 things : abceliexyz and abcjoexyz

**echo** {090..100} : 090 091 092 … 100

**echo** a{A{1,2},B{3,4}}b : aA1b aA2b aB3b aB4b

# Work with echo

**echo $**{USER} : Will output the value of the variable USER

**echo \**$USER : Outputs \USER

**echo** my name is \\$USER : Will output my name is elie because the \\ will cancel the one the other.

# Work with partitions

**df -h** : Will show us partitions and directories with some details.

**du -sh** /etc/\* : Shows directories size inside of etc directory.

# Work with users

**who** : Used to see the users currently logged.

# Work with processes

**ps -aux** : Show all running processes.

**ps -U** elie : Show the processes running by the user elie.

# Get help for a certain command

**ls --help**

**man ls**

# Search for files

**locate** or **find** : Locate is faster but need database update. Find is used when we know where the file is located.

**updatedb** : Will do a database update

**find** /home/ **-i -name** “file1” : The ‘-i’ will ignore the case sensitive.

**locate -i** fil : If you know a fragment of file

**locate -ib** fil : Without the ‘-b’ it will take the path as a search parameter so it will output everything in it. The ‘-b’ will force to use the last thing we wrotes in the command.

# Archieve and compress

Archieving is different than compressing. Archieving means putting files into one file without reducing size so without compressing.

We have 3 types of zip methods :

- gzip

- bzip

- zip

**tar -cvf** archieve.tar varlog.tar.gz anothervar.rpm : The “-c” means create, the “-v” means archeive and the “-f” means specify files. So here we archieve files into one file without compressing.

**tar -cvzf** archieve.tar varlog.tar.gz anothervar.rpm : The “-z” will zip the resulting file.

**tar -uvf** archieve.tar \*.txt : We can update the contents of the archieve file so for example if we made changes in the folder and want to recompress it we can just update. Here we are saying to check all text files that has been changed to update the zip folder in the end.

**tar -tf** archieve.tar : The “-t” will list the files that are present in the archieve folder.

**tar -xvf**  archeive.tar : Extracts files from archieve.

**zip** -9 **-r** test1.zip /home/aravi : Here we used zip so it is another method and “-9” let us do high compression and here we are compressing all subfiles into one archieve which is tes1.zip

**zip -d** test1.zip home/aravi/ioen/testing.txt : The “-d” is used to delete specific file from the archieve.

**zip -fr** test1.zip \* : Here the test1.zip will be updated.

**unzip** test1.zip : To unzip file

# Redirection of errors

Standard input : 0

Standard output : 1

Standard error : 2

**ls** /root 2**>** /temp/errors : Will insert the error line into errors file and the ‘2’ refers to the standard error value.

**find** /etc **-name** passwd **>** /tmp/output 2 **>** /tmp/error : Will redirect the output file to /tmp/output and if there is an error, the /tmp/error will be filled with the error line.

**ls** /home/aravi **>>** /tmp/errors 2 **>** &1 : The **>>** will let insert multiple things instantly so here we insert the output and the error if there is one.

# Pass output as input for a command

**grep** aravi **<** /tmp/output : Will give the output of /tmp/output to the command as an input

# Grep command

**grep -e** “First” **-e** “last” demofile : So the ‘-e’ must be used if we want to search multiple strings

**grep -v** “First” demofile : The ‘-v’ is used to unmatch the word so we will output everything except the word “First”.

**grep** “First” **-A** 2 file : Will output line containing the word “First” and 2 lines after.

**grep** “First” **-B** 2 file : Will output line containing the word “First” and 2 lines before.

**grep** “All” **-C** 1 file : Will output 1 line before and 1 line after.

**grep -ril** /home/aravi/ : The “-l” is used when searching files or directories.

**grep -l** demo\* : Will output the files having “demo” included in their name.

**grep -c** first demofile : Will output the number of times the word “first” is found in demofile.

**grep** “demo**$**” demofie : Will output the line that ends with the word “demo”.

**cat** /var/log/messages **| grep** “^Sep 26 03:26:41” : Will search for lines beginning (^) with date.

# Vi&Vim editor

Vi is the best editor to use in Linux. It is used to modify config files.

The difference between Vi and Vim is that Vim is an imoproved version of Vi where output is colored.

# Manual page

**man -s** 5 passwd : Will give us passwd(5) which is the 5th sub category.

**man -k** printf : Shows us how many related command are there. So all commands related to printing.

**man -s** 5 **-k** passwd : Show the command linked to passwd in the 5th sub category.

♣ In the manual page I can enter / and search.

**pinfo** ls : Similar to man command but different the way and content.

**info** ls : Exactly similar to man command.

# whereis and whatis command

**whereis** python : Show us where it is located

**whatis** ls : Brief description of command

# SSH Servers

Here we are going to explore OpenSSH server&client configuration so how to securely log in to the server and how the client is authentified. OpenSSH use port 22 by default for communicating between client and server. Public and private keys are used for cryptographics purposes.

- Firstly we must install the OpenSSH package :

**yum install** openssh

♣ Difference between yum and rpm : Yum is a package manager and rpms are the actual packages. With yum you can add or remove software. The software itself comes within a rpm.

**rpm -qa | grep** openssh : Will show us the installed version of openssh, so here if we are on the server side we will see the installed package.

- Now we have to configure the config files and SSH keys on server side and authenticate the client.

**systemctl status** sshd : Will show us the status of the service openssh.

**systemctl enable** sshd : To enable the service openssh if it is not enabled.

- Now we must verify the config files :

**vi** /etc/ssh/sshd-config : Now when we do access of the file we can add some code

a) Add the protocol type responsible for a secured environnement :

Protocol 2

UsePAM yes

AllowGroups sshusers (This group only will have access to the ssh server)

- Now we will create the group

**groupadd** sshusers

**usermod -Ag** sshusers aravi : Add user aravi to group sshsuers

**id** aravi : To verify that aravi is part of the group sshusers.

- Now we will try to connect to the server

**ssh** aravi@192.168.2.140 : Here he will ask for passwd of aravi and we will be able to connect to ssh server.

♣ I can also use Putty and choose the connection type as SSH and he will also ask for user’s password.

# SCP (Secured copy)

In this section, we will learn how to securely copy files from a source to another.

- Source IP Server : 192.168.2.140

- Destination IP Server : 192.168.2.42

**scp** file1 root@192.168.2.42:/root/ : The ip address and location in orange is the ip address of the destination server and the location where we want to store the file.

We can add multiple files simulatenously

**scp -r** directoryName root@192.168.2.42:/root/ : Here we used the “-r” to copy entire directory.

**scp -C** filename root@192.168.2.42:/root/ : The “-C” is used to compress and send data.

In fact, in this situation the timestamp will be the actual one when we make the copy that’s why we can use the parameter “-p” to preserve parameters.

**scp -vp** filenameroot@192.168.2.42:/root/ : To preserve parameters

**scp -l** 500 filename root@192.168.2.42:/root/ : To limit the bandwidth allocated for the copy.

# Linux Processes