**LINUX LAB GUIDE**

## 

This document serves a guide to practice what has been taught by the instructor in the training session. The course participants can use this a helper guide to practice the below, during the **student lab session** of the training. When the student practices the below material, it acts as a re-enforcer of the concepts learnt during the training.

1. **Terminology:**
2. # : Implies Root user. ( Get into the root user's environment by typing 'su -' )
   1. $ : Implies normal user . ( Exit from the root shell by typing 'exit' )

$ **cd /home**  (Go to the home directory) :

The Bold text is what the user has to type on the command line terminal. The contents inside ( ) describes the command why its used and should **not** be typed.

## **Module 1: Fundamentals of Linux**

Lab : Learning Cmd-line skills

Scenario:

You are new to Linux and learn the basic commands to get used to the Linux OS quickly. The following shall help you.

**Exercise:** Basic Linux Commands using the shell:

Task 1: Using ls :

$ ls (Normal command)

$ ls -l (Detailed list )

Task 2: Using cd :

$ cd /home (Go to the home directory)

$ cd Scripts/Day2 (Go to the specified directory)

Task 3: Using pwd:

$ pwd (Get the current working directory)

$ cd Scripts/Day2

$ pwd

Task 4: Using clear:

$ clear (Clears the screen)

Task 5: Using whoami:

$whoami (Displays the user name who is logged in)

$whoami –help (Shows the help for whoami command)

Task 6: Using man page:

$ man whoami (get the man page for whoami)

$ man lvcreate (get the man page for lvcreate)

Use upper case G to go to end of man page

Use lower case g to go to start of man page

Scroll to examples

Type /volume to highlight the string 'volume' inside the man page.

Press 'n' to go to next occurrence of the string 'volume'.

Press 'q' to exit

$ man man (Know how to use man page)

$ man -k user (Find the man page to create use)

$ man -k user | grep 8

$ man -k user | grep 8 | grep create

Task 7: Using pinfo page:

$ sudo apt install pinfo (Install pinfo software)

$ pinfo -a whoami (Scroll to the whoami info page directly)

$ pinfo (To search for whoami : Press : <|/> ie look for | , then press / to get the Regex input prompt: whoami)

Press n,u to navigate through and press q for quiting

Task 8: Using su command:

$ sudo su (Go to the root shell)

$ passwd (Set the passsword for the root user as root123)

$ exit

$ su - (Gives root access again with root users environment. Enter root passwd)

# passwd <user> (Set the password for the user eg. azureuser)

# exit (Exit from the root shell)

$ sudo passwd (Alternative way of setting user's password running the command with the root's permission.To set the passwd for the current user <azureuser>)

Task 9: Using touch,echo,cat command:

$ touch hello.txt (Creates the file hello.txt)

$ ls -l (Lists all the files in the current working directory)

$ echo hello (Prints the string 'hello' to the console)

$ echo hello > hello.txt (Writes the string 'hello' to the file)

$ cat hello.txt (Displays the contents of the file hello.txt)

Task 10: Navigate the Linux File system hierarchy:

$ su - (Get access to the root shell)

# cd / (Change to / )

# ls

# ls -l

# cd /etc (Change to /etc to see its contents )

# ls -l /etc

# cd /root (Change to /root to see its contents )

# ls (Doesn't show hidden files)

# ls -a (Shows hidden files)

# ls -lR (List files recursively traversing sub-directories)

# ls -l /home (List the content of /home directory )

# # ls -ld /home (Doesn't list the contents but the properties of /home)

#exit

Lab : **Working with Text Files**

Scenario:

Having obtained the basic skills you are now ready to do some useful work like copying, renaming, finding , creating backup archive files. The following shall help you.

**Exercise : Doing useful work with text files using Linux commands.**

Task 11: Playing with Wildcards on the cmd line:

$ cd /etc

$ ls \* (List everything inside /etc)

$ ls -d a\* ( Lists the properties of everything starting with a. Do not show file contents if its directory )

$ ls -d [a-c]\* ( Lists Files/Dirs that ends with a to c )

$ ls -d a?s\* ( Lists files/dirs that starts with a , some charcater , s character and anything )

$ ls -d a[ln]\* ( Lists files/dirs that starts with a , followed by either l or n ,and anything )

Task 12: Using copy command and delete commands :

$ cd Scripts

$ cp Day2/Module1/1\_Variables/hello.sh /tmp/ (Copies the file hell.,sh inside the

Day2/Module1/1\_Variables/ dir to /tmp dir)

$ cp -r Day2/Module1/1\_Variables/ /tmp/ ( Copies recursively directories )

$ mv /tmp/hello.sh ~/ (Moves hello.sh to the home directory)

$ rm ~/hello.sh (Deletes the file hello.sh)

$ rm -rf /tmp/1\_Variables ( Removes the whole directory tree /tmp/1\_Variables)

$ mkdir -p /tmp/example/1 ( Creates the directory trees inside /tmp )

$ rm -rf /tmp/example (Removes the whole directory tree)

Task 13: Using Symbolic links:

$ cd ~/Scripts/Day2

$ ln -s ~/Scripts/Day1 day1 (Creates a symbolic link by name 'day1' pointing to

the directory ~/Scripts/Day1/)

$ cd day1 (get inside ~/Scripts/Day1/ using the symbolic link day1 )

$ rm day1 (Delete the symbolic link)

Task 14: Using find command:

# cd /etc

# find . -name 'hosts' (Finds the file hosts inside /etc. The search path is mentioned as '.' which means the current working dir )

# find /etc -name 'hosts' (Same as above. But the search path is mentioned as /etc)

$ find . -user azureuser | more (Searches for all files created by the user 'azureuser')

$ mkdir /tmp/azureuser ; find . -user azureuser -exec cp -r {} /tmp/azureuser \;

(First create the /tmp/azureuser . ';' acts as cmd separator. Finds all files created by azureuser and then passes the result line by line to {} for the copy command)

$ find . -size +1M (Finds all files less than 1MB)

$ mkdir /tmp/azureuser ; sudo find /etc/systemd/ -exec grep -l 'root' {} \; -exec cp {} /tmp/azureuser \;

(creates the dir, finds the files , passes the files found to the 1st grep command does the filtering, passes its result to the 2nd grep command)

$ rm -rf /tmp/azureuser

$ mkdir /tmp/azureuser ; sudo find /etc/systemd/ -exec grep -l 'root' {} \; -exec cp {} /tmp/azureuser \; 2> /dev/null ( Re-direct the errors to null )

$ sudo find /etc -name '\*' -type f | xargs grep '127.0.0.1'

(Same as above but using xargs command)

Task 15: Using tar command:

Tar:

$ tar -zcvf Scripts.tgz ~/Scripts (Creates an archive of the dir Scripts and stores the result as Scripts.tgz)

$ tar -zxvf Scripts.tgz -C /tmp/azureuser (Extracts the Scripts.tgz into the /tmp/azureuser dir)

Task 16: Using more, head, tail commands:

more:

$ cat /var/log/messages | more (displays contents screen sequentially wise)

head:

$ cat /etc/passwd | head -n 5 (Displays only 5 lines from the start of the file )

$ cat /etc/passwd | tail -n 3 (Displays only 3 lines from the end of the file )

**Module 2: User , Group, Permissions Management**

Lab : Managing Users & Groups

Scenario:

Having mastered the fundamental skills in Linux, you enter into the world of administering linux machines. As part of this learning curve you are expected to manage your users of the linux systems segregating them into various groups. After user and group creation you shall give permissions to the resources they create. The following shall help you.

**Exercise : Working with users & groups**

Task 17: Displaying the user & group owner

$ cd

$ cat /etc/passwd (Displays all the users in that linux system)

$ id azureuser (Display the properties of the user azureuser)

$ touch hello

$ ls -l hello (Displays the file & group owner of the file hello)

Task 18 : Creating & Managing users with adduser, usermod

# adduser rama (Create the user rama)

# id rama (Display the uid, gid of the user rama)

# tail -n 3 /etc/passwd (Shows the uid/gid of user rama in passwd & shadow files)

# tail -n 3 /etc/shadow

Task 19 : Creating & Managing groups with groupadd, usermod

# groupadd sales ( Create the group sales)

# usermod –help

# usermod -aG sales rama (Makes rama member of sales)

# id rama

# tail -n 3 /etc/passwd

# tail -n 3 /etc/group (notice rama is in sales group)

# tail -n 3 /etc/passwd

Task 20: Set the password expiry time for new users

# vim /etc/login.defs (Change PASS\_MAX\_DAY to 99)

# ls -al /etc/skel (Displays the hidden files in /etc/skel)

# touch /etc/skel/hello

# adduser sita

# id sita

# tail -n 3 /etc/passwd

# tail -n 3 /etc/shadow (Notice the password is set 99 days)

# ls -al /home/rama (User rama doesn't contain the hello file)

# ls -al /home/sita (User sita contains the hello file)

**Exercise : Setting Permissions for your users**

Task 21: Creation of shared directories within a group

# mkdir -p /data/salesdir (Create directory tree /data/salesdir)

# ls -l /data/

# chgrp sales /data/salesdir (Change group owner for salesdir)

# ls -l /data

# chmod 770 /data/salesdir/ (Others will have no permission, rwx are set for user root & rwx for sales group)

# ls -l /data/

Task 22: Creation of files inside the shared directories for its group members

# su – rama (Rama already a member of sales group)

$ cd /data/salesdir/

$touch ramasales (user owner: rama ; group owner: rama for the file ramasales)

$ ls -l /data/salesdir (Check the properties of the files inside salesdir)

# chown rama /data/salesdir (salesdir has userowner: rama groupowner:sales)

# ls -l /data/salesdir

Task 23: Checking File permissions

# cd /home/rama

# touch hello (Hello file has userowner: root ; groupowner: root)

# ls -l (Note the permissions of the file hello)

# su – rama

$ echo hello >> hello (Gets permission denied. Because even though hello file is inside /home/rama, its permissions are root for owner & group ownership)

Task 24: Deleting Files inside a directory

$ rm -f hello (Deletion is possible .. Why see below)

$ ls -ld . (/home/rama directory has rwx set for its user ownership. Hence deletion is possible. Hence creation / deletion of files is possible for user rama)

Task 25: Insufficient user permission even though group permission available:

# cd /data/salesdir

# touch file2

# ls -l file2 (user: root ; group:root)

# chown rama:sales file2 ( Set owner: rama group:sales for the file2)

# chmod 464 file2 (Set permissions: r for user, rw for group, r for others )

# ls -l file2

# su – rama

$ cd /data/salesdir

$ echo hello >> file2 (Permission denied.. because file2 has only read permissions only for rama. Order of checking is: 1st check for user permission )

$ exit

Task 26: Rectify the permissions for the file file2 so that rama can modify the file

# cd /data/salesdir

# chmod 644 file2 (Now file2 has write permissions for its user rama)

# su – rama

$ echo hello >> /data/salesdir/file2 (Modification possible its rw permission set for its owner)

**Exercise : Setting special Permissions**

Task 27: Setting Stickybit permissions

# ls -l /data/salesdir (Has the ramasales file userowner: rama groupowner:rama)

# su – sita

$ rm -f /data/salesdir/ramasales (Sita can delete the ramasales file because rwx is set for the groupownership of salesdir. Sita is a member of sales group. So she can modify the files inside salesdir directory )

# chmod +t /data/salesdir (Setting stickybit for the salesdir directory)

# ls -ld /data/salesdir (stickybit is set now. As indicated by T)

# su – sita

$ cd /data/salesdir

$ rm -f file2 (Operation NOT permitted. Sita is not the owner of file2 and NOR the owner of the directory salesdir which contains file2. Hence cannot delete file file2)

**Exercise : Creating default file & directory permissions using umask**

Task 28: Seeing default umask permissions

base is 777 for directories & 666 for files

Subtract umask from the base setting of umask to get the actual permissions

# cd /root

# umask (Returns 022 as answer because its set in /etc/profile)

# touch 123 (Sets 644 as its permission to the file 123)

# mkdir hellodir

# ls -l hellodir (Sets 755 as its permission to the hellodir directory)

Task 29: Changing umask permissions on the shell

# umask 027 (Change umask to 027)

# touch 456 (Sets 640 as its permission to the file 456)

# rm -rf hellodir

# mkdir hellodir (Sets 750 as its permission)

Task 30: Delete users,groups with deluser, groupdel

# cd /root

# rm -rf hellodir 123 456 /home/rama/hello

# deluser rama

# deluser sita

# groupdel sales

**Module 3: Operating Running systems**

Lab : Maintaining software on running linux systems

Scenario:

Now having created users & groups in the linux machines, you wanted to install software so that the users can make using them. Once the users starts writing programs using the software you have provided, you need to monitor the processes they create. Sometimes you need to login into the system to install packages. Hence we shall also look into ssh. This module will help you do so.

**Exercise : Software installation using repositories**

Task 31: Check the repository files

# cd /etc/apt

# cat /etc/apt.source.list (Default repository used by ubuntu)

# ls /etc/apt/source.list.d (Lists all additional repositories. Chrome & vscode in our system uses them)

Task 32: Show info about a package

# apt show nmap

Task 33: Search and install packages

# apt search nmap (List all packages which relates to nmap)

# apt install nmap

Task 34: Remove packages

# apt remove nmap (Removes nmap and also show if any additional remove commands needs to be done)

# apt autoremove (Additional packages not used will also be removed)

**Exercise : Managing processes and jobs**

Task 35: Start the job dd if=/dev/zero of=/dev/null . What do you see? Stop the job

# dd if=/dev/zero of=/dev/null (Console doesn't return)

# Ctrl-Z (Returns the console)

# bg <job id> (Put job id in background)

# jobs (Shows the job running in background)

Task 36: Start the multiple jobs dd if=/dev/zero of=/dev/null in background

# dd if=/dev/zero of=/dev/null &

# dd if=/dev/zero of=/dev/null &

# jobs (Shows 3 jobs running in background)

Task 37: kill job id 2

# fg 2 (Brings the job id 2 to foreground. Gives the console)

# Ctrl-C (Kills the job)

# jobs (Now shows only 2 jobs)

Task 38: Options in ps cmd

# ps aux (Shows process info)

# ps -ef (Shows process info along with pid & ppid)

# ps fax (Shows process info in tree form)

Task 39: Kill using top & ps

# dd if=/dev/zero of=/dev/null &

# dd if=/dev/zero of=/dev/null &

# dd if=/dev/zero of=/dev/null &

# top (Press k and select the pid)

# ps aux | grep dd (List all processes running dd command)

# kill <PID> (Kill the selected process)

# killall dd (Kills all dd processes )

**Exercise : Login into system using ssh**

Task 40: Login using ssh

$ ssh -i AzureNavin\_key.pem azureuser@40.71.227.104 (Logs

into Linux VM having IP address 40.71.227.104 as azureuser with the mentioned SSH key )