LAB 3 – MANAGING UBUNTU SERVER – USERS – PERMISSIONS AND REMOTE CONNECTIONS

3.1 Users in Ubuntu server:

Users of Ubuntu server are divided into three types including root user, regular users and system users. There is just only one root user per system. It is defined in /etc/passwd. This is the superuser with unlimited control. By default, root account is created but disabled for login. Its password is locked. Instead, sudo is used for administrative tasks.

3.1.1 Enable root user and set its password:

Practice: Login into the system with root user:

Step 1: Login to your regular account, set password and enable root account

sudo passwd root

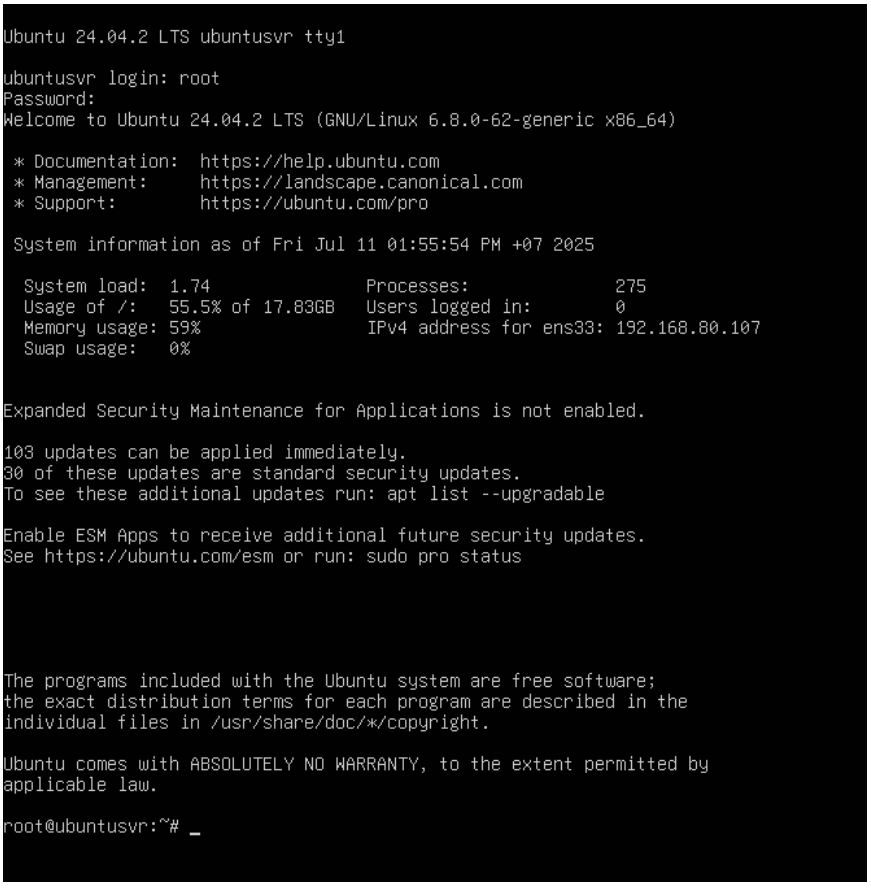

Image 3.1 Login into the system with root super user.

3.1.2 Create new user and add sign it to sudo group with administrative privileges:

a/ Create a new user:

Command syntax: sudo adduser <user_name>

Example: sudo adduser nvcuong

Right after creating nvcuong, the system created the home directory /home/nvcuong and default shell bin bash defined in /bin/bash. User nvcuong belongs to nvcuong group and could not use sudo for administrative privileges.

```
nvcuong@ubuntusvr:/$ cd home
nvcuong@ubuntusvr:/home$ ls
nvcmis  nvcuong
nvcuong@ubuntusvr:/home$ cd nvcmis
-bash: cd: nvcmis: Permission denied
nvcuong@ubuntusvr:/home$
```

Image 3.2 User nvcuong could not view the nvcmis folder.

```
nvcuong@ubuntusvr:/home$ mkdir myfolder
mkdir: cannot create directory ‘myfolder’: Permission denied
nvcuong@ubuntusvr:/home$ sudo apt update
[sudo] password for nvcuong:
nvcuong is not in the sudoers file.
nvcuong@ubuntusvr:/home$
```

Image 3.3 User nvcuong is not allowed to create folder in home and use sudo keyword

```
nvcuong@ubuntusvr:~/dailytask$ ls
11_07_2025.txt
nvcuong@ubuntusvr:~/dailytask$ cat 11_07_2025.txt
Finish lab2 - Installing and managing ubuntu server
Write lab3 - Managing ubuntu server - User - Privileges
nvcuong@ubuntusvr:~/dailytask$
```

Image 3.4 User nvcuong have full privileges to create, edit folders and files in /home/nvcuong folder.

b/ Assign the user to sudo group:

If the user could make admistrative tasks such as installing software, changing password of root or other users, taking ownership of a specific file or folder, delete a user, changing permissions of a file or folder, the user must be assigned to sudo group.

Syntax of the command to assign to sudo group:

sudo usermod -aG sudo <user_name>

```
nvcmis@ubuntusvr:~$ sudo usermod -aG sudo nvcuong
[sudo] password for nvcmis:
nvcmis@ubuntusvr:~$ _
```

Image 3.5 Assign nvcuong to sudo group

```
nvcuong@ubuntusvr:~$ sudo apt update
[sudo] password for nvcuong:
Hit:1 http://vn.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://vn.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://vn.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://vn.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,237 kB]
Get:5 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [28.5 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:8 http://vn.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [253 kB]
Get:9 http://vn.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [163 kB]
Get:10 http://vn.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [1,418 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [984 kB]
Get:12 http://vn.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [303 kB]
Get:13 http://vn.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:14 http://vn.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1,105 kB]
Get:15 http://vn.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [281 kB]
Get:16 http://vn.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:17 http://vn.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:18 http://vn.archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [39.9 kB]
Get:19 http://vn.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7,052 B]
Get:20 http://vn.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:21 http://vn.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [28.0 kB]
Get:22 http://vn.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [28.3 kB]
Get:23 http://vn.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:24 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [175 kB]
Get:25 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.5 kB]
Get:26 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [208 B]
Get:27 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [871 kB]
Get:28 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [191 kB]
Get:29 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.2 kB]
Get:30 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Fetched 7,994 kB in 4s (1,965 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
100 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg
on in apt-key(8) for details.
nvcuong@ubuntusvr:~$ _
```

Image 3.6 User nvcuong was able to run sudo to update apt completely as root user. With sudo privileges, nvcuong is even able to delete user root.

c/ Limit a user to the specific administrative tasks:

It’s extremely dangerous for the system if many users were assigned to sudo group, because they will be able to do everything even when they disable, change password or delete root and the other users. Therefore, in realistic, you have to assigned the specific tasks for each users.

Practice script:

1/ Create user lhttung

2/ Grant him the following tasks:

- Restart only nginx
- Reboot the server
- Run a script only

3/ Login as lhttung and test the result

User guide: after creating user, edit sudoers like

sudo visudo

At the bottom of the file add the following line:

lhttung ALL=(ALL) NOPASSWD: /bin/systemctl restart nginx, /usr/bin/apt update

Use Case	`sudoers` Rule Example
Restart only Nginx	`username ALL=(ALL) NOPASSWD: /bin/systemctl restart nginx`
Tail logs	`username ALL=(ALL) NOPASSWD: /usr/bin/journalctl -u nginx`
Reboot the server	`username ALL=(ALL) NOPASSWD: /sbin/reboot`
Run a script only	`username ALL=(ALL) NOPASSWD: /opt/scripts/restart-docker.sh`

Image 3.7 Some common limited task in the reality.

3.2 Managing privileges on a folder and file.

3.2.1 View the permissions on a folder and file:

Permissions of Linux includes three types read (r), write (w) and execute (x). These permissions are applied to three types of objects including owner (u), group (g) and others (o).

To view permissions on a specific file or folder, the syntax: **ls -l <file_name>**

```
nvcuong@ubuntusvr:/var/www/html/nvcmis$ ls -l index.html
-rw-r--r-- 1 nvcmis nvcmis 81 Jul 11 09:55 index.html
nvcuong@ubuntusvr:/var/www/html/nvcmis$
```

I

Image 3.8 Permissions on the file index.html

The first – represent types of objects as the following image:

The **first character** indicates the **type of file**:

Symbol	Meaning
-	Regular file
d	Directory
l	Symbolic link
c	Character device
b	Block device
s	Socket
p	Named pipe (FIFO)

Image 3.9 List of object types

From the image 3.8

-rw-r--r-- means:

- means regular file, not folder and the other types

rw- means the owner of this file has permissions of read and write. Why doesn't the owner has execute permissions because this is html file not executable file.

r-- means the group has permissions of read only.

The last r-- means the others has permissions of read only.

-rw-r--r-- 1 : what does 1 mean? 1 represent number of hard links to this file, in this context, there is only one link to this file that it is itself.

-rw-r--r--1 nvcmis nvcmis

The first nvcmis represent the user who is the owner of the file.

The second nvcmis represent the group, when creating user nvcmis, the group nvcmis created at the same time.

3.2.2 Change permission on a folder and file:

Command to change permissions on the file or folder: chmod xyz <file_name>

x, y, z are octal digit number with read (r) has value 4, write (w) has value 2 and execute (x) has value 1

Example: if you want to allow group and others has read and write permission means that 4 + 2 = 6 so that you write the command as the following:

chmod 666 index.html

```
nvcuong@ubuntusvr:/var/www/html/nvcmis$ chmod 666 index.html
chmod: changing permissions of 'index.html': Operation not permitted
nvcuong@ubuntusvr:/var/www/html/nvcmis$ sudo chmod 666 index.html
[sudo] password for nvcuong:
nvcuong@ubuntusvr:/var/www/html/nvcmis$ ls -l index.html
-rw-rw-rw- 1 nvcmis nvcmis 81 Jul 11 09:55 index.html
nvcuong@ubuntusvr:/var/www/html/nvcmis$ _
```

Image 3.10 Permissions on index.html after executing chmod 666

3.3 Managing remote connection via ssh:

3.3.1 Check if ubuntu server allowed ssh or not:

Is the SSH server installed? which sshd

If it returns something like /usr/sbin/sshd, then **SSH is installed**.

Or install ssh server via commands:

sudo apt update

sudo apt install openssh-server

Is the SSH service running?

sudo systemctl status ssh

If the services has not run yet, start it by command:

sudo systemctl start ssh

Check If the Server Is Listening on Port 22

sudo ss -tuln | grep :22

Check Firewall Rules (UFW): sudo ufw status

If not allowed, enable it with: sudo ufw allow ssh | 22

Check SSH Configuration File: sudo nano /etc/ssh/sshd_config

After all the above configurations have been set correctly, try to connect from remote computer via ssh by command: ssh username@server_ip

```
nvcuong@ubuntusvr:~$ which sshd
/usr/sbin/sshd
nvcuong@ubuntusvr:~$ sudo systemctl status ssh
[sudo] password for nvcuong:
Warning: The unit file, source configuration file or drop-ins of ssh.service changed o
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
     Active: inactive (dead)
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
nvcuong@ubuntusvr:~$
```

Image 3.11 Check if ssh ready for use

The image 3.11 has shown that ssh server installed but not running

```
nvcuong@ubuntusvr:~$ sudo systemctl start ssh
Warning: The unit file, source configuration file or drop-ins of ssh.service changed on disk. Ru
nvcuong@ubuntusvr:~$ sudo systemctl status ssh
Warning: The unit file, source configuration file or drop-ins of ssh.service changed on disk. Ru
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
     Active: active (running) since Fri 2025-07-11 17:17:16 +07; 32s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 3557689 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 3557691 (sshd)
      Tasks: 1 (limit: 4548)
     Memory: 4.4M (peak: 4.5M)
        CPU: 17ms
     CGroup: /system.slice/ssh.service
             └─3557691 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jul 11 17:17:16 ubuntusvr systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Jul 11 17:17:16 ubuntusvr sshd[3557691]: Server listening on :: port 22.
Jul 11 17:17:16 ubuntusvr systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
nvcuong@ubuntusvr:~$
```

Image 3.12 Start ssh services and display the status

Now, we have completed configurations ssh for remote connects, so it's time for us to test remote connection via ssh from another machines.

3.3.2 Connect ubuntu server from a remote machine:

Firstly, in the remote machine, ping ip of the ubuntu server as the image:

```
nvcmis@nvcmiss-MacBook-Pro ~ % ping 192.168.80.107
PING 192.168.80.107 (192.168.80.107): 56 data bytes
64 bytes from 192.168.80.107: icmp_seq=0 ttl=64 time=93.023 ms
64 bytes from 192.168.80.107: icmp_seq=1 ttl=64 time=5.483 ms
64 bytes from 192.168.80.107: icmp_seq=2 ttl=64 time=6.103 ms
64 bytes from 192.168.80.107: icmp_seq=3 ttl=64 time=3.318 ms
64 bytes from 192.168.80.107: icmp_seq=4 ttl=64 time=3.546 ms
64 bytes from 192.168.80.107: icmp_seq=5 ttl=64 time=8.520 ms
64 bytes from 192.168.80.107: icmp_seq=6 ttl=64 time=7.284 ms
64 bytes from 192.168.80.107: icmp_seq=7 ttl=64 time=7.451 ms
64 bytes from 192.168.80.107: icmp_seq=8 ttl=64 time=3.990 ms
```

Image 3.13 Ping to test connection to remote ubuntu server

That's ok for connection

Secondly, make a connection using nvcmis user

```
nvcmis@nvcmiss-MacBook-Pro ~ % ssh nvcmis@192.168.80.107
The authenticity of host '192.168.80.107 (192.168.80.107)' can't be established.
ED25519 key fingerprint is SHA256:6ChCtPbfyLtmJM7b6zkqEREW14nRcfNyRw3+3bPLRAo.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:2: 192.168.80.199
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.80.107' (ED25519) to the list of known hosts.
nvcmis@192.168.80.107's password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-62-generic x86_64)
```

Image 3.14 Connect to ubuntu server via ssh and user nvcmis

The last steps, we make some commands in the ubuntu server via ssh

```
Last login: Tue Jun 24 09:44:44 2025 from 192.168.80.192
nvcmis@ubuntusvr:~$ cd ../
nvcmis@ubuntusvr:/home$ ls
nvcmis  nvcuong
nvcmis@ubuntusvr:/home$
```

Image 3.15 Test some commands after making ssh connection.

3.3.3 Copy a file from remote machine to ubuntu server:

In previous labs, we refered to touch, vi, nano to create and edit a file content in ubuntu enviroment. However, it's much more inconvenient because if you wanna write code c++, php or python, no IDE supported. Therefore, you should make the content of a file in windows or MacOS with visual code, visual studio… the copy it to ubuntu server.

The most convenient solution is to enable ssh on ubuntu server, then use scp command at remote machine as the following syntax:

scp <source_file_path>
<user_name>@<ubuntu_ip_address>:/home/<user_name>

Example: scp C:\Users\YourName\Desktop\main.cpp
ubuntu@192.168.1.100:/home/ubuntu/

Practice exercise:

1/ Write C++ code to solve quadratic in set of real numbers in just one file quadratic.cpp

2/ Copy it to /home/<your_user_name> folder of your ubuntu server. After that, use cat or vi command to view the content of the quadratic.cpp.

3/ Install g++ compiler on Ubuntu server

4/ Compile quadratic.cpp to executable file and run it.

5/ Make configurations to allow all other user could run this executable file.

6/ Sign out of the system and login again with the others account and test executing quadratic program.

```
nvcmis@ubuntusvr:~$ ls
Documents  index.html~  Test3105.cpp
nvcmis@ubuntusvr:~$ g++ Test3105.cpp -o Test
nvcmis@ubuntusvr:~$ ./Test
Press 1 to solve exercise 1
Press 2 to solve exercise 2
Press 3 to solve exercise 3
Press 0 to quit 0
Press your choice: 2
Press your choice: 1
Enter student Id:1
Enter student full name: John Smith
Enter mark of student:8.5
Press your choice: 1
Enter student Id:2
Enter student full name: Mary
Enter mark of student:7.9
Press your choice: 2
1       John Smith
2       Mary
Press your choice: 0
nvcmis@ubuntusvr:~$
```

Image 3.16 App C++ running on Ubuntu server