JAMMY JELLYFISH

UBUNTU 22.04 SERVER

ADMINISTRATION

HANDS-ON-GUIDE

RUZANNA RAMLI
ROSLAINI IBRAHIM

HANDS-ON-GUIDE: UBUNTU 22.04 SERVER ADMINISTRATION

ABSTRACT

HANDS-ON-GUIDE: UBUNTU 22.04 SERVER ADMINISTRATION

This book introduces students with skills on managing servers in Ubuntu 22.04 environment. The emphasis is on what administrators will need to know to perform service such as DNS, NFS, Samba, Web and Email. Basic administration command also included in this book to guide beginner user learn easily. Configuration explore in this book are using para virtualization method from Oracle VM Virtualbox software.

This book contains ELEVEN (11) main sections, step by step of hands-on-guide tasks and tutorials to enhance students learning abilities. The authors hope this book can provide useful resources to students and lecturers and can also be use as independent self-learning source.

Ubuntu 22.04

JAMMY JELLYFISH

Server Administration

RUZANNA BINTI RAMLI
ROSLAINI BINTI IBRAHIM
DEPT OF INFORMATION TECHNOLOGY & COMMUNICATION
POLYTECHNIC KUALA TERENGGANU

CONTENTS

CONTENTS

CONTENTS

Table of Contents

TOPIC 1
INTRODUCTION

1.0 INTRODUCTION

1.1 Introduction to Linux.

Linux operating system was developed through the cooperation of many people around the world. It is a product of the Internet and a free operating system that fall under GNU General Public License. This meant that anyone can use, modify and redistribute the source code.

The Linux kernel was developed by Linus Torvalds in 1991. The name Linux is a combination of Linus and UNIX. Linus Torvalds began the Linux as a personal endeavor when he was a computer science student at the University of Helsinki, Finland. At that time, mostly operating systems were proprietary and expensive. Torvalds wanted to create an operating system that was freely available to anyone who wanted to use the operating system.

Nowadays, Linux is considered one of the most stable, secure and reliable operating systems and widely used in servers, supercomputers and enterprise environments. The Linux community has expanded to include thousands of developers and users who work on the creation and upkeep of the operating system. The Linux operating system also has many distributions (versions) such as Fedora, Ubuntu, Manjaro, CentOS, LinuxMint, OpenSuse, Debian, Arch and lots more.

Figure 1.1: Variety of Linux distribution operating system

1.2 Why Use Linux?

There are several reasons why someone choose to use Linux operating system:

Open-source:

Linux is open-source software, meaning that the source code is freely available for anyone to use, modify, and distribute. This allows for a large and active community of developers to contribute to the development and maintenance of the operating system.

Customizability:

Linux is highly customizable, and users can easily install and configure different software packages to suit their needs.

Stability and security:

Linux is known for its stability and security, as it is less prone to crashes and viruses than other operating systems.

Cost-effective:

Linux is free to download and use, making it a cost-effective option for individuals and businesses.

1.3 How Does Linux Work?

Linux was designed to be similar to UNIX but evolved to run hardware ranging from phones to supercomputers. All Linux based operating systems include a Linux kernel that manages hardware resources and a set of software packages that make up the rest of the operating system. Organizations can also run Linux operating system on Linux servers. There are two main components in Linux operating system:

- **Kernel:**
  This is actually a component of the Linux system as a whole. The kernel, which controls CPU, memory and peripherals, servers as the brain of the system. The operating system's kernel is at the most fundamental level.

- **Desktop Environment:**
  The user actually engages in interaction at this point. There are numerous desktop environments available such as GNOME, Cinnamon, Mate, Pantheon, Enlightenment, KDE, Xfce and so on. Every desktop environment has pre-installed programmes including file managers, configuration tools, web browsers, games, application software and etc.

1.4 Official Mascot of Linux

The official Linux mascot is a penguin named Tux. Created by artist Larry Ewing in 1996, the penguin was first used as his Linux mascot for the Linux kernel. Tux quickly became popular in the Linux community and is now one of Linux's most recognizable icons. Tux was chosen as the mascot because the penguin is a rare animal found in the wild in Antarctica, just as Linux is a unique and powerful operating system. The name Tux comes from the short form "Torvalds Unix", in honor of the creator of Linux, Linus Torvalds.
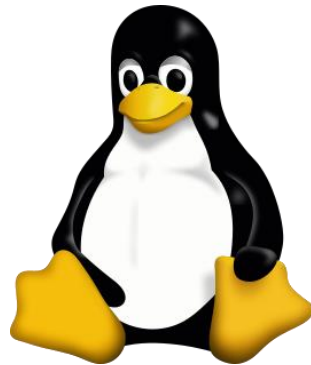
Figure 1.2: Linux official mascot, Tux the penguin.

1.5 History of Ubuntu

Ubuntu is one of a number of Linux distributions. The source code that makes up the Ubuntu distribution originates from another, much older Linux distribution known as Debian (it was started by two people named Debra and Ian). Debian is still a widely respected operating system as today. A South African internet mogul (who made his fortune selling his company to VeriSign for around $500 million) decided to make the operating system more user friendly distribution. He took the Debian distribution and worked to make it a new operating system which he called as Ubuntu.

Then, he formed a company called Canonical Ltd to promote and provide support for Ubuntu. In addition Shuttleworth has formed and funded (to the tune of $ 10 million) a foundation to guarantee the future of Ubuntu. Ubuntu has since gone from strength to strength. Dell and other hardware vendors now ship computers pre-loaded with Ubuntu and Ubuntu usually tops the chart at DistroWatch.com (a web site that tracks the popularity of the various Linux distributions).

The word "Ubuntu" is an ancient Zulu and Xhosa word that means "humanity to others". Ubuntu also means "I am what I am because of who we all are". It was chosen because these sentiments precisely describe the spirit of the Ubuntu distribution.

1.6 Official Ubuntu Version

Ubuntu releases are made semiannually by Canonical Ltd, the developers of the Ubuntu operating system, using the year and month of the release as a version number. The first Ubuntu release, for example was Ubuntu 4.10 and was released on 20 October 2004. The number will change accordingly based on the released month if it has changes.

There are two types of Ubuntu releases:

- **Interim or None-LTS:** Interim release are maintained for 9 months.
- **Long Term Support (LTS):** Each Ubuntu LTS is maintained for 10 years total (5 years of standard support + 5 years of Extended Security Maintenance (ESM).

Support in Ubuntu means that the operating system will be updated with security patches and minor updates during the support period. The support will apply to all Ubuntu flavors as well as the distro's that use Ubuntu as their base system such as Linux Mint.

1.7 Ubuntu Flavors

Ubuntu supports a number of desktop environments as official flavors. User can install a number of desktops on their based Ubuntu setup if desired. Each desktop environment has its own strong points and weak points depending on the user's needs and taste.

These are the official flavors of Ubuntu:

- Kubuntu – Ubuntu with the K Desktop environment
- Lubuntu – Ubuntu that uses LXDE Desktop environment
- Xubuntu – Ubuntu with the XFCE Desktop environment
- Ubuntu DDE – This version uses the Deepin Desktop environment
- Ubuntu Budgie – Simplicity and elegance using the Budgie Desktop environment
- Ubuntu GNOME – Ubuntu with the GNOME desktop environment
- Ubuntu Kylin – Ubuntu localized for China with its own desktop environment
- Ubuntu MATE – Ubuntu with the MATE desktop environment
- Ubuntu Studio – Designed for multimedia editing and creation
- Edubuntu – Developed in collaboration with teachers and technologist in several countries

Several version of Ubuntu operating system:

- Ubuntu 23.10 (Mantic Minotaur)
- Ubuntu 23.04 (Lunar Lobster)
- Ubuntu 22.04 LTS (Jammy Jellyfish)
- Ubuntu 20.04 LTS (Focal Fossa)
- Ubuntu 18.04 LTS (Bionic Beaver)
- Ubuntu 16.04 LTS (Xenial Xerus)
- Ubuntu 14.04 LTS (Trusty Tahr)

TOPIC 2

FILE SYSTEMS AND FILES

2.0 FILE SYSTEMS AND FILES

2.1 Directory Structure

Linux file system is based on a tree hierarchy. The file system hierarchy contains a root directory (/) and its subdirectories. All other directories can be accessed from the root directory. Each directory in Linux has their own function.
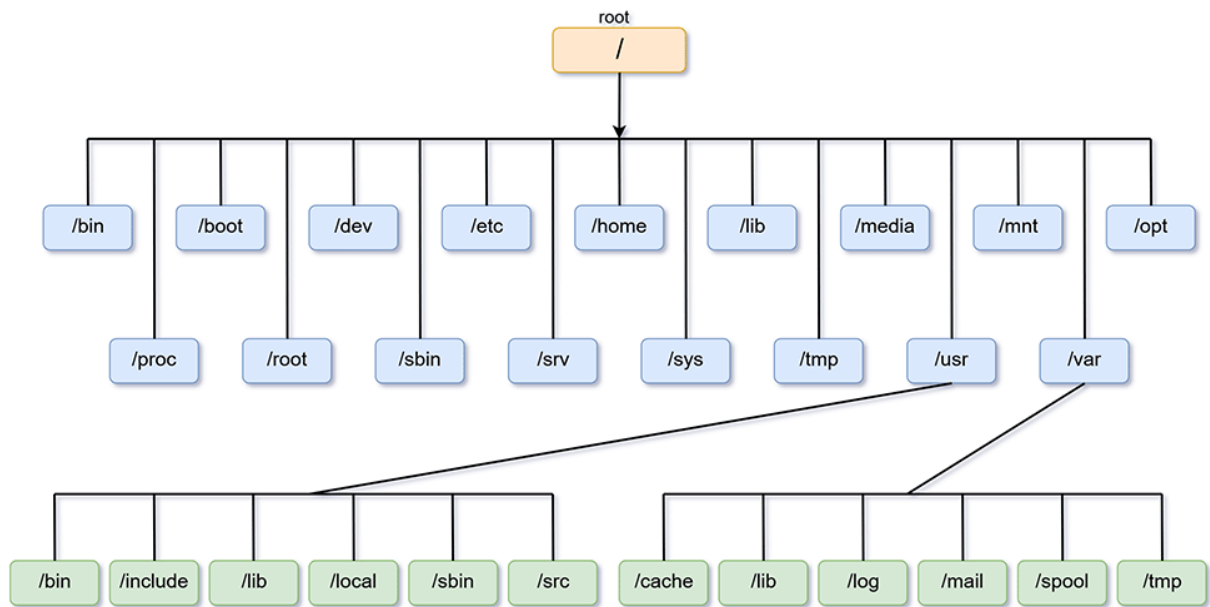
Figure 2.1: File System Hierarchy

2.2 Filename in Linux

- File names are case-sensitive. Example, the file name intro.txt, INTRO.txt and Intro.txt are considers as three different files.
- Avoid using symbol such as /, : , > , < , | and spacing when creating filenames.
- Use characters such as numbers, letters (either uppercase or lowercase), dot ".", and underscore "_" symbol is acceptable.

2.3 Path Name

A path is a unique location to a file or a folder in a file system of an Operating System (OS). A path to a file is a combination of / and alpha-numeric characters. In Linux, there are two different kinds of paths: absolute and relative.
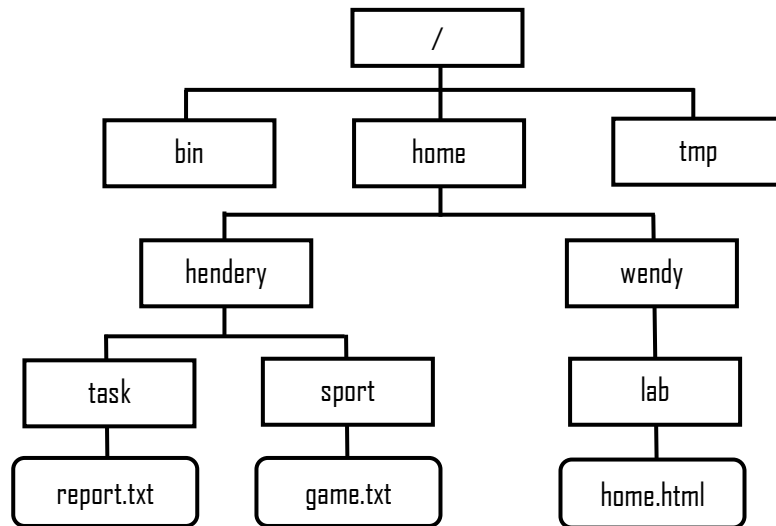
Figure 2.2: Traversing Path

2.3.1 Absolute Path Name

Traces the path from the / (root) directory. Absolute path names always begin with the slash (/) symbol. Absolute paths are useful when you need to refer to a file or directory from anywhere in the system.

Example (Refer Figure 2.2):

    a. directory task: /home/hendery/task
    b. directory lab: /home/wendy/lab
    c. file game.txt: /home/hendery/sport/game.txt
    d. file home.html: /home/wendy/lab/home.html

2.3.2 Relative Path Name

Traces the path from the current print working directory (pwd). Relative paths are more convenient when you are working within a specific directory or subdirectory. It can be used as a shortcut to locate directory or file.

- The dot (.) represent current directory
- The dot dot (..) represent parent directory

Example (Refer Figure 2.2):

- Current print working directory (pwd) is at **/home/wendy**.
    a. directory lab: lab
    b. file home.html: lab/home.html
    c. directory bin: ../../bin
    d. directory sport: ../hendery/sport
    e. file report.txt: ../hendery/task/report.txt

2.4 Exercises
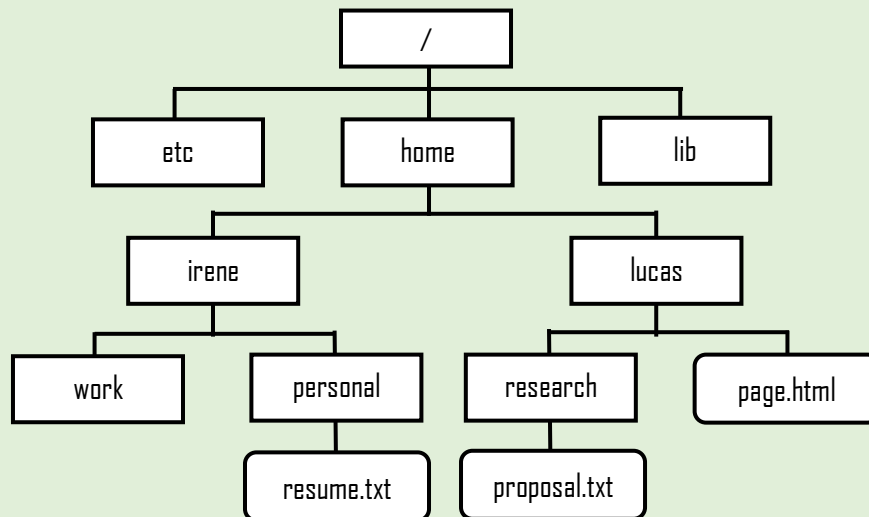
Figure 2.3: Traversing Path

Based on Figure 2.3, answer the following questions:

Assume you are log in as richard and the working directory is at **/home/lucas**.

Provide the absolute and relative path to traverse:

    a.   page.html file
    b.   proposal.txt file
    c.   work directory
    d.   personal directory
    e.   resume.txt file
    f.   etc directory
    g.   lib directory

TOPIC 3
BASIC CLI COMMAND

3.0 BASIC COMMAND LINE INTERFACE (CLI)

3.1 Print Working Directory

To check your current location in Linux system, use "pwd" command. By default, normal user always be in their home directory.

Syntax:

```
$ pwd
```

3.2 Listing Directory Content

Displays files and directories in your current directory.

Syntax:

```
$ ls [option]
```

Option:

-F   - To distinguish between files and directory.

-a   - To show hidden files starting with a dot (.).

-l   - To produce a long listing format information.

-i   - To provide a unique inode information the kernel assigns to each object in the file system.

-s   - To print block size of each file

3.3 Changing Directory

Changing from one location to another location in the terminal prompt.

Syntax:

```
$ cd directory
```

Going back to parent directory (relative path).

```
$ cd ..
```

Changing directory to / directory (absolute path).

```
$ cd /
```

Going back to home directory using shortcuts.

```
$ cd ~
```

3.4 Creating New Directory

Creating directories using command "mkdir" (Make Directory) give the owner read, write and execute permissions, while giving groups and others read and execute permissions.

Syntax:

```
$ mkdir directory
```

Option:

-p   - To create the second subdirectory from the parent's directory

Create multiple directories at the same time.

```
$ mkdir Sem1 Sem2 Sem3
```

Create a subdirectory named OSOS under Sem2 directory without change the current directory (relative path).

```
$ mkdir -p Sem2/OSOS
```

3.5 Creating Files

There are several ways to create a text file in the terminal prompt. Either using touch, cat or symbol (>).

3.5.1 Using cat >
Creating file info.txt using command "cat", together with some content. To make sure all the content capture in the file, press [ENTER] and then CTRL + C. This task will write the content into the file and back at the terminal prompt.

```
$ cat > info.txt
```

3.5.2 Using touch
Creating an empty file named nota.txt using "touch" command.

```
$ touch nota.txt
```

3.5.3 Using >

Creating empty file named reference.txt using symbol (>).

```
$ > reference.txt
```

3.6 Editing Files Using Nano

To edit a file, you can run the "nano" command and specify the name of the file to edit.

Syntax:

```
$ nano nota.txt
```

There are several shortcuts can be used to navigate thru nano editor easily.

a. Save changes [Ctrl + S]
b. Find text [Ctrl + W]
c. Replace old text with new text [Ctrl + \ ]
d. Copy text [Ctrl + 6, then highlight all text that want to copy and stop with Alt + 6]
e. Paste text [Ctrl + U]

3.7 Viewing File

To view file content in the terminal prompt, various command that can be used to achieve the result.

Use command "more" to view content one page at a time. Use <spacebar> to go to next page or <Enter> to go to next line.

Syntax:

```
$ more nota.txt
```

Use command "head", to read at the beginning lines of the file content. The default number of lines is 10 if the option –[number] not use when the command execute.

Syntax:

```
$ head -[number] nota.txt
```

Use command "tail", to read at the end of last few lines of the file content.

Syntax:

```
$ tail -[number] nota.txt
```

Use command "less", to allows both forward and backwards movement when view file in terminal prompt. To exit the navigation screen, press :q .

Syntax:

```
$ less nota.txt
```

Use command "cat", to view content at the terminal prompt.

Syntax:

```
$ cat nota.txt
```

3.8 Copying File/Directory

Allows user to make a copy of file or directory using cp (copy) command in the terminal prompt.

Syntax:

```
$ cp [option] source destination
```

Option:

-i   – To enable interactive mode before overwrite an existing file.

-p   – To preserve file access or modification times of the original file.

-R   – To recursively copy contents of a directory to another location

Copy file passwd to current directory.

```
$ cp /etc/passwd .
```

Copy file passwd to Sem2/OSOS directory.

```
$ cp /etc/passwd Sem2/OSOS
```

3.9 Moving and Rename File/Directory

By using mv (move) command, it enables user to move or rename a file or directory.

Syntax:

```
$ mv source destination
```

Move file nota.txt to Sem2/OSOS

```
$ mv nota.txt Sem2/OSOS
```

Rename file nota.txt to task1.txt

```
$ mv Sem2/OSOS/nota.txt Sem2/OSOS/task1.txt
```

3.10 Deleting File/Directory

Use the rm (remove) command to delete files.

Syntax:

```
$ rm [option] filename
```

Option:

-i   – To delete file with some precautions and enable interactive mode.

-r   – To delete non-empty directory. This will delete the directory and all its content.

-f   – To force remove and the user will not be prompt to confirm the delete task.

Use the rmdir (remove directory) command to delete empty directory.

Syntax:

```
$ rmdir directoryname
```

3.11 Exercises
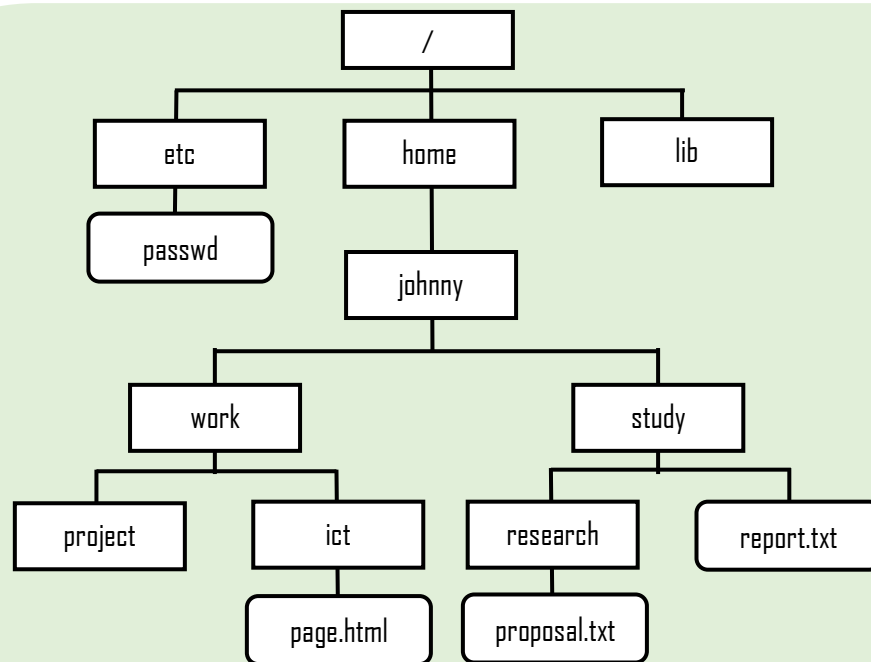
Figure 3.1: Linux Directory Structure

Based on Figure 3.1, perform the following task step by step:

    a.   Create the directories and files in the johnny's home directory.
    b.   List all the created files and directories.
    c.   Change current directory to / (root).
    d.   Check the working directory.
    e.   Copy the file passwd into project directory and rename as newpwd.
    f.   Go back to johnny's home directory.
    g.   Insert some new text into report.txt.
    h.   Rename file page.html as home.html.
    i.   Move file report.txt to research directory.
    j.   Show the usage of cat, head, tail, more and less in viewing file newpwd.
    k.   Delete file proposal.txt.
    l.   Delete directory project.

TOPIC 4

USER & GROUP MANAGEMENT

4.0 USER AND GROUP MANAGEMENT

4.1 Users

Users must authenticate to any system they need to use. This authentication provides access to resources and a customized, user-specific environment. The user's identity is based on their user account.

User account information is stored in the /etc/passwd file. This information includes the account name, home directory location, and default shell, among other values. Linux sysadmins should be able to recognize these fields.

Each field is separated by a colon (:) character, and not all fields must be populated, but you must delineate them.

Here's an example of the /etc/passwd fields:

```
<Username>:<Password>:<UID>:<User-Details>:<Home-Directory>:<Login Shell>
```

4.2 Creating Users

Syntax:

```
$ sudo adduser [username]
```

Example:

```
$ sudo adduser hana
```

Change user password by root administrator, if necessary.

Syntax:

```
$ sudo passwd [username]
```

Example:

```
$ sudo passwd hana
```

Check user id to check uid, gid and groups of the created user.

Syntax:

```
$ id [username]
```

or

```
$ tail -10 /etc/passwd
```

4.3 Groups

A group in Linux is a collection of accounts that can be given special permissions on the system. In Linux there are two types of groups; primary group and secondary group.

Primary group is also known as private group. Primary group is compulsory. Every user must be a member of a primary group and there can be only one primary group for each member.

Secondary group is optional. As per requirement, a user can be added in a single secondary group or in multiple secondary groups.

Here's an example of the /etc/group fields:

```
<GroupName>:<Password>:<GID>:<Member>
```

4.4 Creating Groups

Syntax:

```
$ sudo addgroup [groupname]
```

Example:

```
$ sudo addgroup jke
$ sudo addgroup dee
```

Check group id of the created group.

Syntax:

```
$ more /etc/group
```

4.5 Modifying Users

4.5.1 Setting the Primary Group
User can only have one primary group at one time.

Syntax:

```
$ sudo usermod -g [groupname] [username]
```

Example:

```
$ sudo usermod -g jke hana
```

4.5.2 Setting the Secondary Group
User can have multiple secondary groups at one time.

Syntax:

```
$ sudo usermod -G [groupname] [username]
```

Example:

```
$ sudo usermod -G dee hana
```

4.6 Deleting Users and Group

4.6.1 Delete User
Delete user and its home directory.

Syntax:

```
$ sudo deluser --remove-home [username]
```

Example:

```
$ sudo deluser --remove-home hana
```

4.6.2 Delete Group

Syntax:

```
$ sudo delgroup [groupname]
```

Example:

```
$ sudo delgroup dee
```

4.7 Exercises
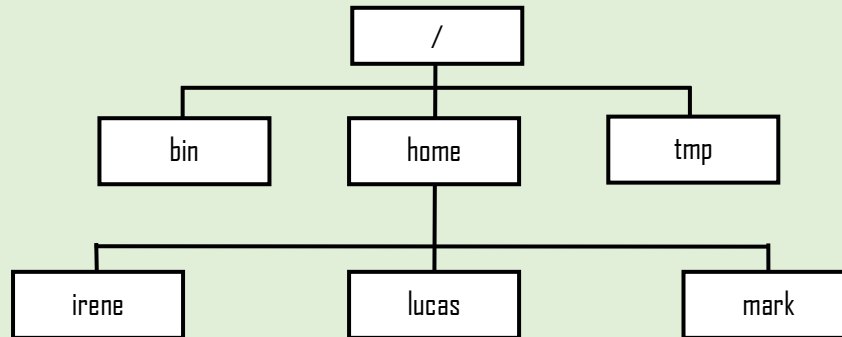
Figure 4.1: Linux Directory Structure

Table 4.1: Directory and Files Details

User	Primary Group	Secondary Group
irene	jtmk	dit
lucas	pkt	ddt
mark	pkt	jtmk

Based on Figure 4.1 and Table 4.1:

1. Create new groups; pkt, jtmk and dit
2. Create new user; irene, lucas and mark with their appropriate password.
3. Set the primary and secondary group for each user as table 4.1
4. Log in to each user.

TOPIC 5

FILE PERMISSION

5.0 FILE PERMISSIONS

5.1 Introduction of Linux File Permissions

Linux is a clone of UNIX, the multi-user operating system which can be accessed by many users simultaneously. Linux can also be used in mainframes and servers without any modifications. But this raises security concerns as an unsolicited or malign user can corrupt, change or remove crucial data. For effective security, Linux divides authorization into 2 levels.

- Ownership
- Permission

5.2 Types of Permissions in Linux

5.2.1 Permission Groups
Each file and directory have three users based on permission groups:

Owner (u) – The Owner permissions apply only to the owner of the file or directory; they will not impact the actions of other users. Also known as group user.

Group (g) – The Group permissions apply only to the group that has been assigned to the file or directory, they will not affect the actions of other users.

Others (o) – The Others permissions apply only to other users who are not the owners of the file and do not belong to the group that has been assigned to the file or directory, they will not affect the actions of other users.

All users (a) – The All Users permissions apply to all other users on the system, this is the permission group that you want to watch the most.

5.2.2 Permission Types
Each file or directory has three basic permission types:

Read (r, 4) – The file can be opened, and its content viewed. For a directory, the read permission allows you to list the contents of the directory.

Write (w, 2) – The file can be modified; for a directory, the write permission allows you to create, delete, and rename files within the directory.

Execute (x, 1) – The file can be executed as a program. For a directory, the execute permission allows you to access, or traverse into, the directory, and access any of its contents.

5.3 Changing File Permissions

Permissions file in Linux can be change using "chmod" command. The "chmod" (stands for "Change Mode") command is used to change the permissions of file or directory in Linux.

There are 2 ways to use the command:

   a. Absolute (numeric) mode
   b. Symbolic mode

Syntax:

```
$ chmod permissions filename
```

5.3.1 Absolute (Numeric) Mode in Linux

In this mode, file permissions are not represented as characters but a three-digit octal number. The table below gives numbers for all for permissions types.

Number	Permission Type	Symbol
0	No Permission	—
1	Execute	–x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r–
5	Read + Execute	r-x
6	Read +Write	rw-
7	Read + Write +Execute	rwx

Figure 5.1: Linux Directory Structure

To set read, write, and execute permissions to the owner, and read and execute permissions to the group and others for 'file.txt':

```
$ chmod 755 file.txt
```

To assign read and write permissions to the owner, and only read permissions to the group and others for 'file.txt':

```
$ chmod 644 file.txt
```

To give all permissions (read, write, execute) to the owner, and no permissions to the group and others for 'file.txt':

```
$ chmod 700 file.txt
```

To give read and execute permissions to everyone for 'file.txt':

```
$ chmod 555 file.txt
```

To give read, write and execute permissions to the group for 'file.txt':

```
$ chmod 070 file.txt
```

For more details, refer chmod manual page by entering the following command:

```
$ man chmod
```

5.3.2 Symbolic Mode in Linux

In the Absolute mode, you change permissions for all three owners. In the symbolic mode, you can modify permissions of a specific owner. It makes use of mathematical symbols to modify the Unix file permissions.

To set or change permissions, we can use the **plus** (+) and **minus** (–) and **equal to** (=) operators in chmod command.

Here's a brief explanation of how the +, –, and = operators work in the chmod command:

- The + operator adds permissions to a file or directory without changing the existing permissions. For example, if you want to add execute (x) permission to the user (u) on a file, you would use chmod u+x filename.

- The – operator removes permissions from a file or directory without changing the remaining permissions. For example, if you want to remove write (w) permission from the group (g) on a file, you would use chmod g-w filename.

- The = operator sets the permissions exactly as specified, disregarding the current permissions. For example, if you want to set the user (u) permissions to read and write (rw) only, removing execute if it's there, you would use chmod u=rw filename.

To add read, write, and execute permissions to the owner of the file named 'file.txt':

```
$ chmod u+rwx file.txt
```

To remove write permission from the group and others for 'file.txt':

```
$ chmod go-w file.txt
```

To add execute permission to the group for 'file.txt':

```
$ chmod g+x file.txt
```

To set the permissions so that the user can read/write, the group can read, and others can't access 'file.txt':

```
$ chmod u=rw,g=r,o= file.txt
```

To add read permission to all (user, group, others) for 'file.txt':

```
$ chmod a+r file.txt
```

5.4 Changing Ownership and Group

The "chown" (Change Ownership) command use the change the owner of the file. While the command "chgrp" (Change Group) are used to change group ownership.

For changing the ownership of a file/directory, you can use the following syntax:

```
$ chown [username] [filename]
```

Example:

```
$ chown hana file.txt
```

In case you want to change to change group-owner only, use the following syntax:

```
$ chgrp [group_name] [filename]
```

Example:

```
$ chgrp jke file.txt
```

In case you want to change the user as well as group for a file or directory use the following syntax:

```
$ chown [username:group_name] [filename]
```

Example:

```
$ chown hana:jke file.txt
```

5.5 Exercises
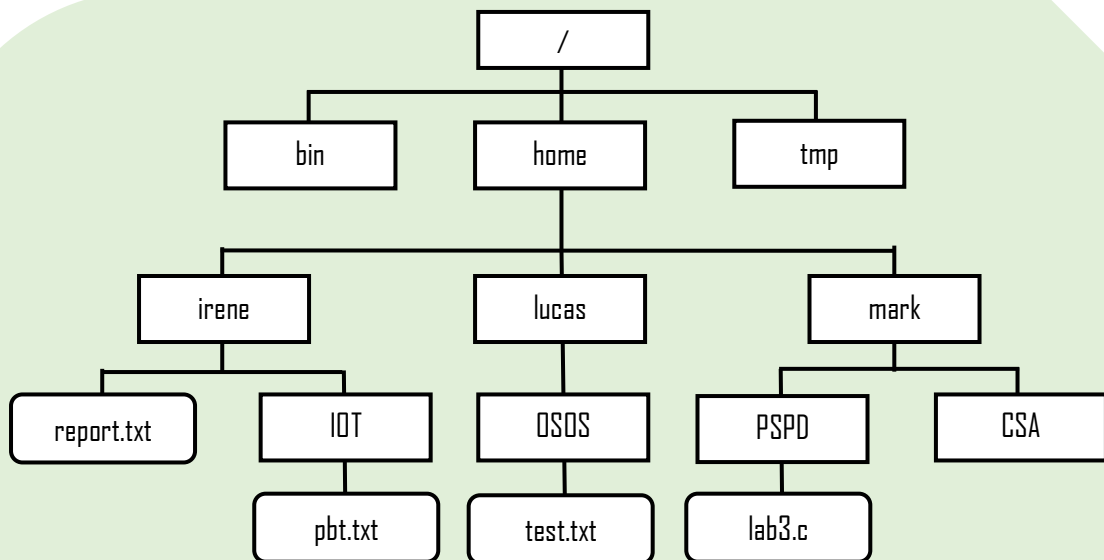
Figure 5.2: Linux Directory Structure

Table 5.1: Directory and Files Details

File/Directory Permission	File/Directory Name	User	Primary Group	Secondary Group
-rw-r--r--	report.txt	irene	jtmk	dit
-rwxr-xr--	pbt.txt	irene	jtmk	dit
-rw-rw-r--	test.txt	lucas	pkt	ddt
-rwxr-xr-x	lab3.c	mark	pkt	jtmk
drwxrwxr--	CSA	mark	pkt	jtmk

Based on Figure 5.2 and Table 5.1:

Noted: Number 1 – 3 were made in the exercises chapter 3.

1. Create new groups; pkt, jtmk and dit
2. Create new user; irene, lucas and mark with their appropriate password.
3. Set the primary and secondary group for each user as table 5.1
4. Log in as the users and create the files and folders.
5. Change the permissions of files and directories as table 5.1.
6. Change the owner of CSA directory to lucas, and the group of CSA directory to ddt.

TOPIC 6

BASIC NETWORKING

6.0 BASIC NETWORKING

6.1 Understanding Oracle Virtual Box Network Environment

In Oracle Virtual Box the choice of networking mode depends on how the user uses their virtual machine (as client or server). Each networking mode has its own uses, behaviors and features:

- **Network Address Translation (NAT)**. If all you want is to browse the Web, download files, and view email inside the guest, then this default mode should be sufficient for you, and you can skip the rest of this section. Please note that there are certain limitations when using Windows file sharing.

- **NAT Service Network**. A NAT network is a type of internal network that allows outbound connections.

- **Bridged networking**. This is for more advanced networking needs, such as network simulations and running servers in a guest. When enabled, Oracle VM VirtualBox connects to one of your installed network cards and exchanges network packets directly, circumventing your host operating system's network stack.

- **Internal networking**. This can be used to create a different kind of software-based network which is visible to selected virtual machines, but not to applications running on the host or to the outside world.

- **Host-only networking**. This can be used to create a network containing the host and a set of virtual machines, without the need for the host's physical network interface. Instead, a virtual network interface, similar to a loopback interface, is created on the host, providing connectivity among virtual machines and the host.

Mode	VM→Host	VM←Host	VM1←VM2	VM→Net/LAN	VM←Net/LAN
Host-only	+	+	+	–	–
Internal	–	–	+	–	–
Bridged	+	+	+	+	+
NAT	+	Port forward	–	+	Port forward
NAT service	+	Port forward	+	+	Port forward

Table 6.1: Overview of Networking Modes

6.2 Network Configuration and Troubleshooting Tools

Command line tools are necessity to configure and troubleshoot a variety of different network conditions.

6.2.1 ip

The ip command in Linux is a powerful tool that not only displays the current IP address of a system but also allows you to view and manage the current configuration of network interfaces, IP addresses, routes, and ARP tables.

a. **ip address show** is used to show the IP address and information about network interfaces:

```
$ ip address show
```

b. **ip address show <interfaces>** is used to show the IP address and information about specific network interfaces:

```
$ ip address show enp0s3
```

c. **ip link set dev <interface-name> down** is used to disable network interfaces, keeping it in state where it cannot transfer or receive data:

```
$ sudo ip link set dev enp0s3 down
```

d. **ip link set dev <interface-name> up** is used to activate network interfaces, keeping it in state where it can transfer or receive data:

```
$ sudo ip link set dev enp0s3 up
```

e. **ip route** is used to show the kernel routing table:

```
$ ip route
```

6.2.2 ifconfig

The ifconfig command commonly used command line utility to configure network, but it is not installed by default. The ifconfig utility is provided by net-tools package.

a. To install the net-tools package:

```
$ sudo apt install net-tools
```

b. **ifconfig** is used to show the IP address and information of all active network interfaces:

```
$ sudo ifconfig
```

c. **ifconfig <interfaces>** is used to show the IP address and information about specific network interfaces:

```
$ ifconfig enp0s3
```

d. **ifconfig down <interface–name>** is used to deactivate network interfaces, keeping it in state where it cannot transfer or receive data:

```
$ sudo ifdown enp0s3
```

e. **ifconfig up <interface–name>** is used to activate network interfaces, keeping it in state where it can transfer or receive data:

```
$ sudo ifup enp0s3
```

6.2.3 ping

The **ping** command in Linux is a utility that helps to test connectivity between two devices on a network. This command sends a request to a specified device and waits for a response. The response from device helps us to determine whether device is available or not.

```
$ ping www.pkt.edu.my
```

6.2.4 traceroute

The **traceroute** command is a network diagnostic tool that allows users to track the route that a packet takes from the source computer to the destination. This tool is widely used to troubleshoot network issues such as high latency, packet loss, and connectivity problems.

```
$ traceroute www.google.com
```

6.2.5 netstat

The **netstat** (network statistics) command provides information about network connections, ports in use, and the processes using them. There are many netstat variables to display the different output. Listed some ways of using netstat with different variable:

```
$ netstat -a
```

```
$ netstat -au

$ netstat -lt

$ netstat -nr
```

6.2.6 host

The host command used to perform DNS lookup operation. Host can be utilized to find the IP address of a specific domain name.

```
$ host www.google.com
```

6.3 How to Configure Static IP Address on Ubuntu

After installing Ubuntu 22.04 the default network interface assigns an IP address using the DHCP server. Also, the wireless network will be active and enable networking over the Wi-Fi network.

You can also configure the network interface with static IPv4 addresses. Ubuntu 22.04 uses the Netplan as a network manager.

This tutorial helps you to configure network interfaces on Ubuntu 22.04 Desktop and Ubuntu 22.04 Server editions. The desktop users can use any one of the below methods but the server users that have CLI only access use the second method to edit network interface configuration files.

6.3.1 Configuring Static IPv4 Address on Ubuntu 22.04 Desktop with GUI

The Ubuntu Desktop editions have a graphical interface for configuring the network interfaces. Follow the below steps to configure a static IP address on Ubuntu 22.04 Desktop system.

➢ Click the network icon in the upper-right corner.
➢ Then expand the Wired Connected dropdown.
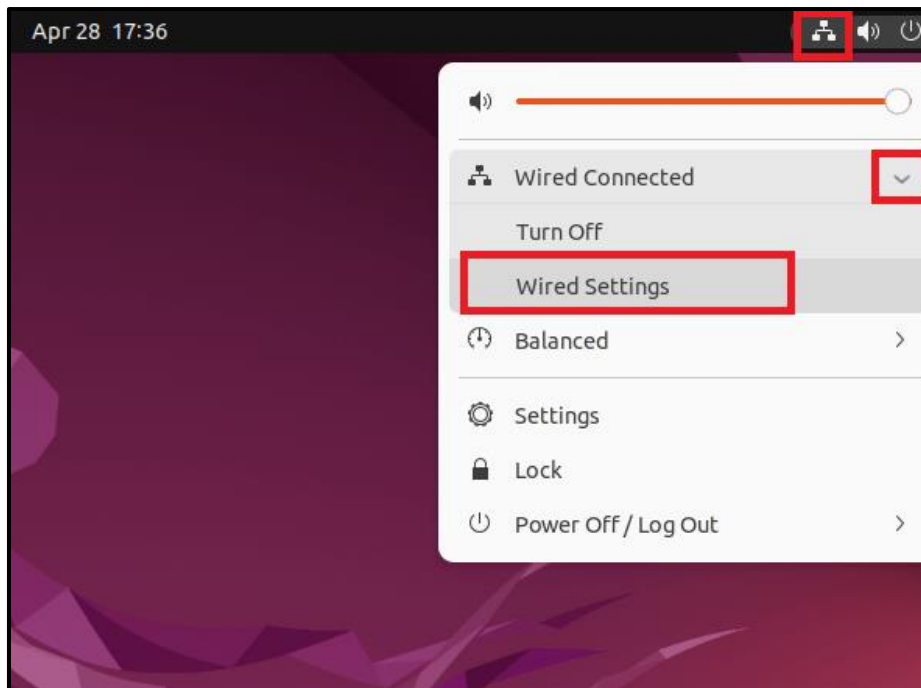➢ Now, click on Wired Setting as shown below image.

Figure 6.1: Edit Wired Interface Settings

- A network settings dialog box will appear.
- Now, click **"Network"** in left sidebar.
- Under the **Wired** section, click the **Gear icon** as showing in below image.
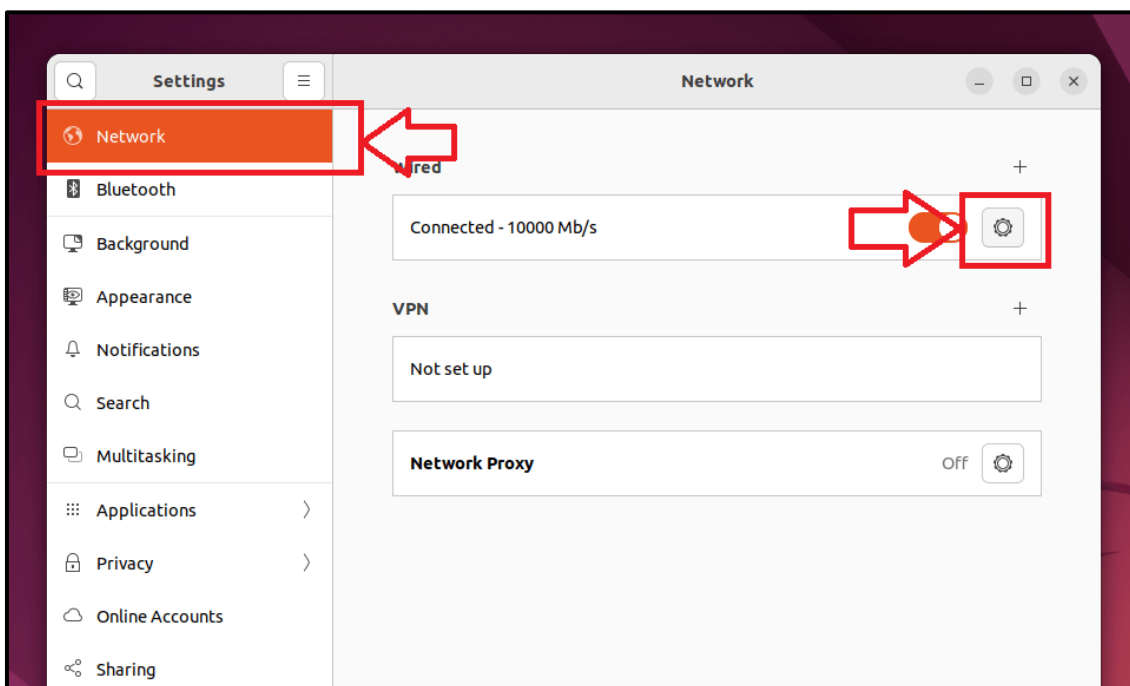
Figure 6.2: Edit Network Interface

➢ A new Wired dialog box will appear.
➢ Now, click on **"IPv4"** tab.
➢ Set **IPv4 Method** to Manual.
➢ Input a valid IP address, Netmask and Gateway address.
➢ Set the DNS server (optional).
➢ Click **Apply** button to save changes.

Figure 6.3: Configuring Static IP on Ubuntu

➢ Finally disable and then again enable networking to apply changes.

Figure 6.4: Disable and Enable Networking

➢ All done. Ubuntu Desktop system IP address is now changed.

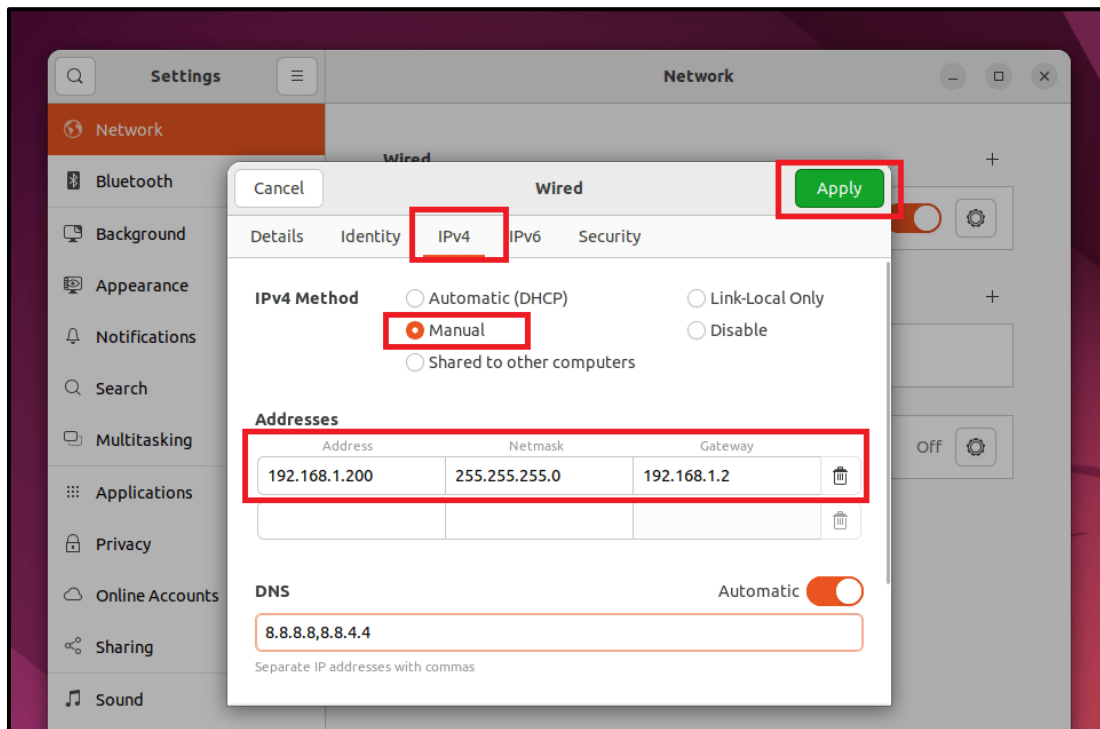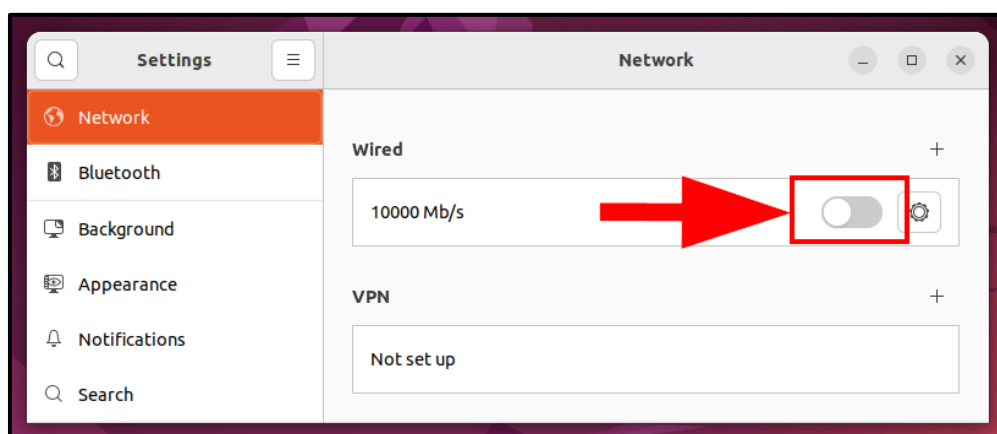## 6.3.2 Configuring Static IPv4 Address on Ubuntu 22.04 Server with CLI

Server editions are installed without any graphical interface. In that case, you need to configure the IP address using the command line.

Ubuntu uses the Netplan as a network manager that is responsible for configuring the network interfaces. Follow the below steps:

1. First, find out the network interface name in your system. The interface name can differ based on the installation type and system hardware. To find the interface name type:

   ```
   $ sudo ip a
   ```
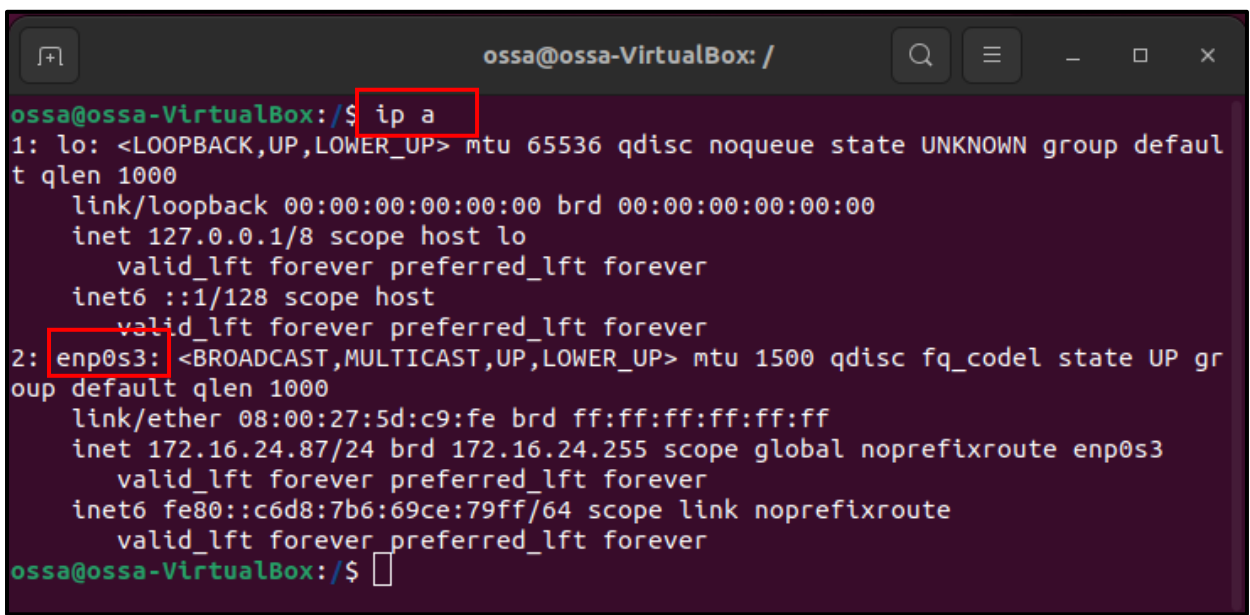

Figure 6.5: Checking the network interface name

   The above output shows that the system is configured with the network interface name **enp0s3**. This can be different on your system.

2. Now, create or edit the network configuration file under the **/etc/netplan** directory. Create a configuration file and edit it in an editor:

   ```
   $ sudo vi /etc/netplan/01-netcfg.yaml
   ```

Add the network configuration in YAML format as below:

```
 1 network:
 2     version: 2
 3     renderer: networkd
 4     ethernets:
 5         enp0s3:
 6             addresses:
 7                 - 172.16.24.5/24
 8             nameservers:
 9                 addresses: [8.8.8.8, 8.8.4.4]
10             routes:
11                 - to: default
12                   via: 172.16.24.1
```

In the above configuration:

- **enp0s3** – is the network interface name
- **addresses** – is used to configure IPv4 address on an interface. Make sure to define CIDR. You can add multiple addresses here
- **nameservers** – Set the name servers here. 8.8.8.8 and 8.8.4.4 are Google's name servers
- **routes** – This is used to set gateway on your system.

Make sure the IPv4 address belongs to the system network and has the correct gateway IP address. Do not use tab in this file instead use space bar only and the indentation as above should be follows. Once confirmed, save file content and close it.

3. You can check for any error in the file configuration by running the following command:

```
$ sudo netplan try
```

4. Apply the changes by running the following commands. Now, execute the following command to apply the changes:

```
$ sudo netplan apply
```

That's it. The static IP address is configured on your Ubuntu system.

TOPIC 7
DOMAIN NAME SYSTEM

7.0 DOMAIN NAME SYSTEM (DNS)

A Domain Name System (DNS) is an internet service used to resolve a domain name to IP address and vice versa. When configuring DNS server, there are two files require to create which is Forward or Reverse Lookup Zone. Forward Lookup Zone (FLZ) let user's lookup up the IP address resolution of a domain from a DNS server. While the Reverse Lookup Zone (RLZ) let user's lookup the domain name from IP address.
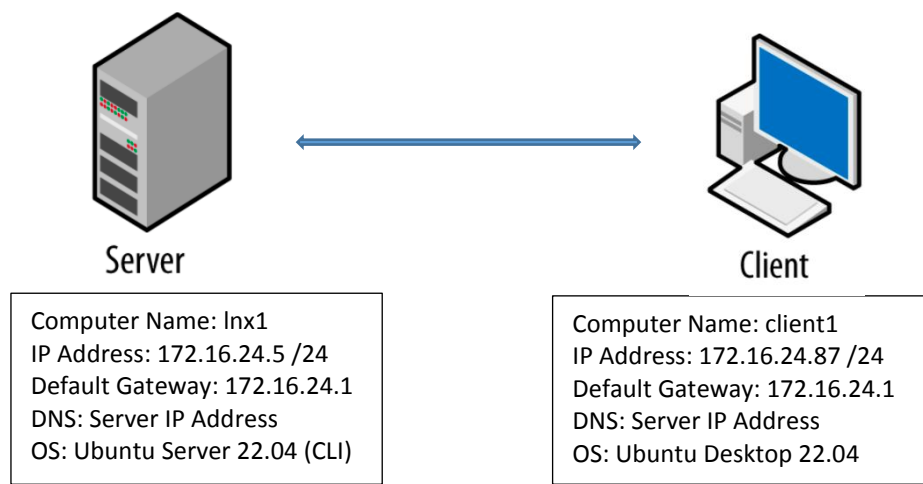
7.1 Set Up Environment

Server	Client
Computer Name: lnx1 IP Address: 172.16.24.5 /24 Default Gateway: 172.16.24.1 DNS: Server IP Address OS: Ubuntu Server 22.04 (CLI)	Computer Name: client1 IP Address: 172.16.24.87 /24 Default Gateway: 172.16.24.1 DNS: Server IP Address OS: Ubuntu Desktop 22.04

Figure 7.1: Virtual Machine (VM) environment

Let's create a DNS server for cille.com.my domain for testing purpose in this VM environment.

Domain Name: cille.com.my
DNS server Name: lnx1.cille.com.my
IP address: 172.16.24.5

Configure the hostname of the server to lnx1.
```
$ sudo hostnamectl set-hostname lnx1
```

7.2 Configuring DNS Server

7.2.1 Install DNS Server
The package of the DNS server in Ubuntu operating system is bind9. User can use the apt command to install the bind9 package along with other utilities packages.

```
$ sudo apt-get install -y bind9 bind9utils
```

7.2.2 Creating DNS Zone and Zone Files

Creating DNS Zone

The /etc/bind will be created after successful installation of bind9 service. This location is the main directory that holds configuration files and zone for DNS server.

For the zone creation, use the /etc/bind/named.conf.local file as we are configuring the dns in the control environment in VM. The global configuration file for DNS is /etc/bind/named.conf.

Let's begin editing the DNS zone for your domain.

```
$ sudo nano /etc/bind/named.conf.local
```

Creating Forward Zone

The following is the forward zone entry for the cille.com.my domain in the named.conf.local file. This forward zone information will translate a fully qualified domain name (FQDN) into an IP address.

```
zone "cille.com.my" IN {
        type master;
        file "/etc/bind/fw.cille"; // Forward zone file
        allow-update { none; };
};
```

Creating Reverse Zone

The following is the reverse zone entry for the 172.16.24.0/24 network in the named.conf.local file. This zone translates an IP address into a fully qualified domain name (FQDN).

```
zone "24.16.172.in-addr.arpa" IN {
        type master;
        file "/etc/bind/rv.cille"; // Reverse zone file
        allow-update { none; };
};
```

Creating Zone Files

Once the dns zone complete, create dns zone files for forward and reverse zones.

Creating Forward Zone file

Create a zone file /etc/bind/fw.cille for the forward zone.

```
$ sudo cp /etc/bind/db.local /etc/bind/fw.cille
```

```
$ sudo nano /etc/bind/fw.cille
```

Type of record in the zone files:

- SOA – Start of Authority
- NS – Name Server
- A – Record for IPv4 Address
- AAAA – Record for IPv6 Address
- MX – Mail for Exchange
- CN – Canonical Name

Domain names should end with a dot (.).

```
$TTL    604800
@       IN      SOA     cille.com.my. root.cille.com.my. (
        200101          ; Serial
        604800          ; Refresh
         86400          ; Retry
       2419200          ; Expire
        604800 )        ; Negative Cache TTL

; Name server record
@       IN      NS      lnx1.cille.com.my.

; A record for name server
@       IN      A       172.16.24.5
lnx1    IN      A       172.16.24.5

; Mail Exchanger record
@       IN      MX 10 mail.cille.com.my.

; A record hostname to IP address
www     IN      A       172.16.24.5
mail    IN      A       172.16.24.5
client1         IN      A       172.16.24.87

; CNAME record
ftp     IN      CNAME www.cille.com.my.
```

Creating Reverse Zone file

Create a zone file /etc/bind/rv.cille for the reverse zone.

```
$ sudo cp /etc/bind/db.255 /etc/bind/rv.cille
```

```
$ sudo nano /etc/bind/rv.cille
```

Type of record in the zone files:

- SOA – Start of Authority
- PTR – Pointer

Domain names should end with a dot (.).

```
$TTL    604800
@       IN      SOA     cille.com.my. root.cille.com.my. (
        200102          ; Serial
        604800          ; Refresh
         86400          ; Retry
       2419200          ; Expire
        604800 )        ; Negative Cache TTL

; Name server record
@       IN      NS      lnx1.cille.com.my.
@       IN      A       172.16.24.5

; Reverse lookup for Name Server
5       IN      PTR     lnx1.cille.com.my

; PTR Record IP address to hostname
5       IN      PTR     www.cille.com.my.
5       IN      PTR     mail.cille.com.my.
87      IN      PTR     client1.cille.com.my.
```

7.3 Validating DNS Syntax

To identify the dns configuration file and zone file free from errors, it is recommended to use this command.

a. named-checkconf - to check the syntax of DNS configuration files for any errors.
```
$ cd /etc/bind
$ sudo named-checkconf
```
Command will return to the shell if there are no errors.

b. named-checkzone – to check the syntax errors in zone files.
```
$ sudo named-checkzone cille.com.my fw.cille
```

Output:
```
zone cille.com.my/IN: loaded serial 200101
OK
```

```
$ sudo named-checkzone 24.16.172.in-addr.arpa rv.cille
```

Output:
```
zone 24.16.172.in-addr.arpa/IN: loaded serial 200102
OK
```

To reload or restart the configuration files and zones after modify use the command below:

- Reload configuration file
  ```
  $ sudo rndc reload
  ```

- Restart dns service
  ```
  $ sudo service bind9 reload
  ```

- Check dns service status
  ```
  $ sudo service bind9 status
  ```

If the configuration file for DNS queries unable to obtain the answer, you can modify the /etc/resolv.conf file.

```
$ sudo nano /etc/resolv.conf
```

Edit the file by input the information below at the end of the file:

```
nameserver 172.16.24.5
domain cille.com.my
search cille.com.my
```

7.4 Verify DNS Server

Use the dig, nslookup or ping command to verify the DNS server by looking up records.

- Verify the forward lookup answer with the nslookup command
  ```
  $ nslookup www.cille.com.my
  ```

- Verify the reverse lookup answer with the nslookup command
  ```
  $ nslookup 172.16.24.5
  ```

- Verify the forward lookup answer with the dig command
  ```
  $ dig www.cille.com.my
  ```

Figure 7.2: Forward Lookup Zone Output

- Verify the reverse lookup answer with the dig command
  $ dig –x 172.16.24.5

Figure 7.3: Reverse Lookup Zone Output

TOPIC 8
NETWORK FILE SYSTEM

8.0 NETWORK FILE SYSTEM (NFS)

Network File System Server or NFS Server is a powerful technology that enables file sharing and data access between computers on a network. It allows multiple users to access files stored on a central server as if the location of directory were located on their own local machines. By implementing NFS server, it simplifies file management, improves collaboration, and enhances data security.
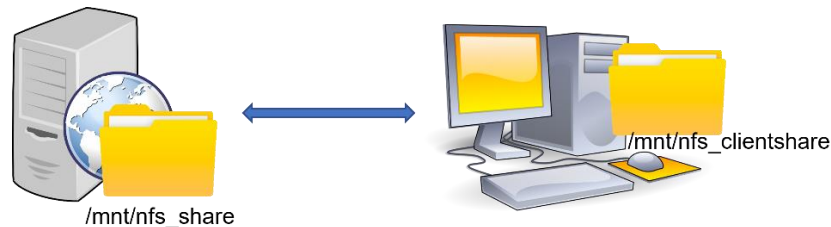

/mnt/nfs_clientshare

/mnt/nfs_share

Figure 8.1: NFS sharing environment

8.1 Install NFS Server

To install an NFS server in Ubuntu 22.04, carry out the following command from the CLI terminal.

```
$ sudo apt-get install -y nfs-kernel-server
```

Verify the status of the service by prompt the following command:

```
$ sudo service nfs-kernel-server status
```

If the package or service are failed to download, user need to update the system package first before continue the download.

```
$ sudo apt update
```

8.2 Configure NFS Server

Follow the step by step configuration to set up a sharing folder thru NFS server on Ubuntu 22.04.

Step 1: Create a shared NFS directory

Create a directory at selected location, usually directory created at /mnt or / directory. First create a directory called "nfs_share" at /mnt directory.

```
$ sudo mkdir -p /mnt/nfs_share
```

Step 2: Set directory permissions and file permissions

Modify the default permission of "nfs_share" directory so that all user may readily access it.

```
$ sudo chown -R nobody:nogroup /mnt/nfs_share
```

Next, set the file permission in "nfs_share" directory to have given the read, write and execute access to the users.

```
$ sudo chmod -R 777 /mnt/nfs_share
```

Step 3: Grant NFS access

The user will be given access to the NFS server thru this step. The main file that needs to be edit is exports file which is located in the /etc directory. This file indicates all directories that a server exports to its clients.

Edit the /etc/exports file in the nano editor.

```
$ sudo nano /etc/exports
```

Then define the access that will be given to the user, either for a single user, a group of users or the entire subnet. As an example, we will allow 172.16.24.0/24 as a whole subnet to access NFS share.

```
/mnt/nfs_share 172.16.24.0/24(rw,sync,no_subtree_check)
```

Note:

- **rw** – Share as read and write permission.
- **ro** – Share as read-only.
- **sync** – file data changes are made to disk immediately.
- **no_subtree_check** – prevent subtree checking.
- **root_squash** – prevent root users connected remotely and having root privilege.

Use Ctrl + X to save and exit from the nano editor.

Step 4: Exporting NFS directory

Apply the changes made in the /etc/exports file to the NFS service.

```
$ sudo exportfs -a
```

Discover the exports directory in terminal.

```
$ sudo showmount -e
```

Step 5: Restart NFS Server

Use the following command to restart the NFS server.

```
$ sudo service nfs-kernel-server restart
```

8.3 Install NFS Client

To allow the user to access the NFS service provided by the server, user need to download the package that will allow those access.

```
$ sudo apt -y install nfs-common
```

8.4 Testing NFS share

Now, let's put a coordination between "nfs_share" directory at server to synchronize between client share folder. Create a mount point on the NFS user machine to accomplish this:
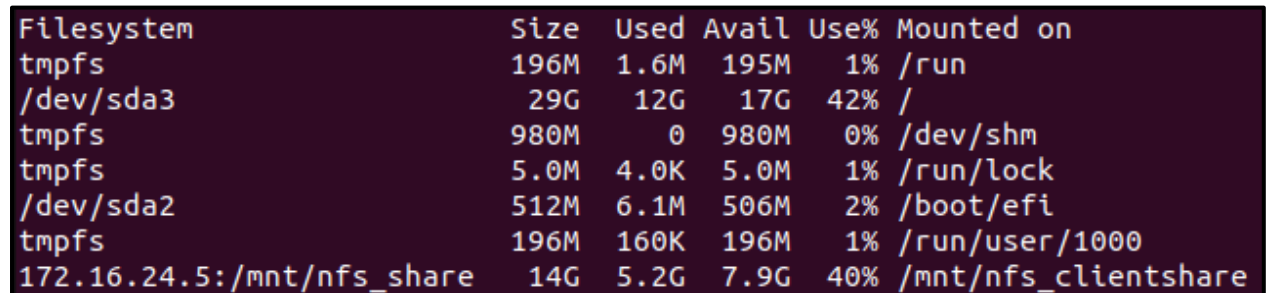
```
$ sudo mkdir -p /mnt/nfs_clientshare
```

Mount the NFS share after that on the client terminal

```
$ sudo mount 172.16.24.5:/mnt/nfs_share /mnt/nfs_clientshare
```

Verify the mount share either successful or not to be mount in client system. The share directory is successful when it's appeared in the disk filesystem.

```
$ df -h
```

```
Filesystem                    Size  Used Avail Use% Mounted on
tmpfs                         196M  1.6M  195M   1% /run
/dev/sda3                      29G   12G   17G  42% /
tmpfs                         980M     0  980M   0% /dev/shm
tmpfs                         5.0M  4.0K  5.0M   1% /run/lock
/dev/sda2                     512M  6.1M  506M   2% /boot/efi
tmpfs                         196M  160K  196M   1% /run/user/1000
172.16.24.5:/mnt/nfs_share     14G  5.2G  7.9G  40% /mnt/nfs_clientshare
```

Figure 8.2: NFS mount connection on client system

For the testing purpose, create a few files in the /mnt/nfs_share directory on the NFS server location to enable access from the client system.

```
$ cd /mnt/nfs_share/
```

```
$ touch test1.txt test2.txt
```

Next, go to the client system to see the list of files that are present in the "nfs_clientshare" directory. You also can test to create or edit file in the directory.

```
$ ls -l /mnt/nfs_clientshare
```

TOPIC 9

SERVER MESSAGE BLOCK

9.0 SERVER MESSAGE BLOCK (SAMBA)

Samba is a widely used open source software suites that enables file sharing and networking between different operating systems. It allows seamless communication between Windows, Linux and macOS systems. By using samba, users can share files, printers and other resources over a network.

Samba service is using Common Internet File System (CIFS) protocol. CIFS is the standard way that computer users share files across corporate intranets and the Internet. Samba consists of two key programs, which is:
- smbd – file and print services and authentication and authorization
- nmbd – name resolution and service announcement

9.1 Install Samba Server
Samba can be found at official Ubuntu repositories. Install Samba package using this command:

```
$ sudo apt install -y samba
```

After installation the Samba service will be operative. To verify the Samba service status either the service run smoothly or not, use the command below:

```
$ sudo service smbd status

$ sudo service nmbd status
```

9.2 Create Samba Users and Directory Structure
Step 1: Create Directory Structure

To manage easily each and every Samba directory and data will be located in the /samba directory. In order to use the directory, we need to create the directory first using the following command:

```
$ sudo mkdir /samba
```

Modify the directory group ownership is set to sambashare. While the Samba installation, this group is created and later you will add all Samba users here.

```
$ tail -5 /etc/group

$ sudo chgrp sambashare /samba
```

Step 2: Create Samba Users

Samba utilizes Linux users and a group permission system. Although it has an authentication mechanism of its own as well. One may create users through the standard Linux useradd tool and later set the user password via smbpasswd utility.

a) Creating a new user named lily to access samba service:

```
$ sudo useradd –M -d /samba/lily –s /usr/sbin/nologin –G
sambashare lily
```

The useradd options mean the following:

- **-M** – do not create the user's home directory, you will manually create this directory
- **-d /samba/lily** – set the user's home directory to /samba/lily
- **-s /usr/sbin/nologin** – disable shell access to this user
- **-G sambashare** – add the user to the sambashare group

b) Creating a user's home directory and set the directory ownership to user lily and group sambashare.

```
$ sudo mkdir /samba/lily

$ sudo chmod 2770 /samba/lily

$ sudo chown lily:sambashare /samba/lily
```

c) Add user account lily to be added to the Samba database and can be done by setting the samba user password.

```
$ sudo smbpasswd –a lily
```

d) Activate the user Samba account by running the following command:

```
$ sudo smbpasswd –e lily
```

9.3 Configure Samba Shares

The samba configuration file is located at /etc/samba/smb.conf. Before you edit this file, it's recommended to create a backup first for reference in future.

```
$ sudo cp /etc/samba/smb.conf{,.backup}
```

Next, edit the Samba configuration file by append the following section with the information below:

```
$ sudo nano /etc/samba/smb.conf

...
# input at the end of the file

[lily]
      path = /samba/lily
      browseable = no
      read only = no
```

```
        force create mode = 0660
        force directory mode = 2770
        valid users = lily @sambashare
```

These options have the meanings given below:

- **[lily]** – The names of the shares to use when logging in. This name will appear when it open to access.
- **path** – The path to the share.
- **browseable** – If the share is to be listed in the list of the available shares. By setting to no, other users won't be able to see the share.
- **read only** – If the users specified in the valid users list can write to this share.
- **force create mode** – Sets the permissions for the files which are newly created.
- **force directory mode** – Sets the permissions for the newly created directories in this share.
- **valid users** – A list of users and groups that are allowed to access the share.

Testing the Samba configuration file to check for any errors.

```
$ sudo testparm
```

Restart the Samba services.

```
$ sudo service smbd restart

$ sudo service nmbd restart
```

9.4 Connect to a Samba Share from Linux

Linux users can access the samba share from the command line, using the file manager or mount the Samba Share.

This method will show to access Samba share using the default file manager in Gnome that has built-in option to access the share directory.

1) Open Files and click on "Other Locations" in the sidebar.
2) In "Connect to server", put the address of the Samba share in this format smb://samba_hostname_or_server_ip/sharename as an example [**smb://172.16.24.5/lily**]
3) Then, click on "connect"
4) The next step is to select "Registered User", put the Samba username and password, and hit "Connect"
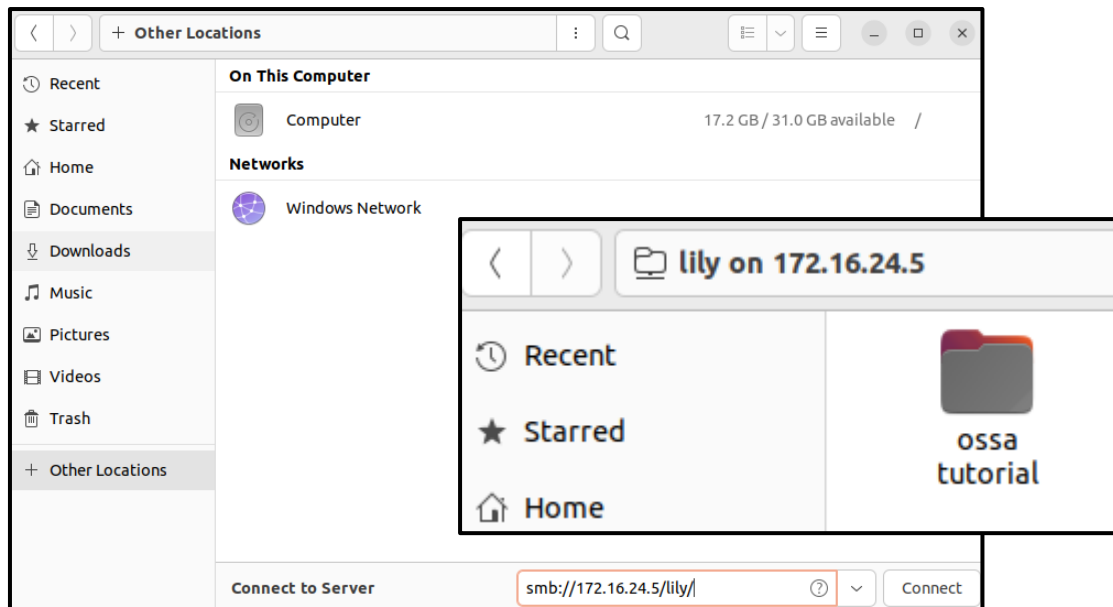5) The files on the Samba server will appear.

Figure 9.1: Samba share thru Linux client

9.5 Connect to a Samba Share from Windows

Windows users can also connect to the Samba share from both command line and Graphical User Interface (GUI). These are the steps to access the share utilizing the Windows File Explorer.

1) Open the "File Explorer" and on the left-panel right-click on "This PC".
2) Find and select the option which says "Choose a custom network location" and then hit "Next".
3) In "Internet or network address", put the address of the Samba share in this format: \\samba_hostname_or_server_ip\sharename as an example [**\\172.16.24.5\lily**]
4) After hitting the option "Next" and you need to enter the login credentials.
5) Then, in the following window, one may type a custom name associated with the network location. The Samba server will choose the default one.
6) Hit "Next" for moving the last screen on the connection setup wizard.
7) Finally, click on "Finish" then Samba server filed will appear.

TOPIC 10
WEB SERVER

10.0 WEB SERVER (HTTP)

A web server stores and delivers the content for a website to clients that request it thru the web browser. A web server can host a single website or multiple websites using the same software and hardware resources. In Linux, the package uses to install and manage web server is known as LAMP stack. LAMP is defined as Linux operating system, Apache web server, MySQL database and PHP Programming Language. A LAMP stack is a group of open source software that is typically installed together in order to enable a server to host dynamic websites written in PHP.

10.1 Installing Apache

Installing Apache web server which is among the most popular web server in the world. To install Apache run this command:

```
$ sudo apt install -y apache2
```

Testing the Apache default web page from client web browser by prompt the information in url http://your_server_ip.

If the web page can be view, your web server is correctly installed and accessible thru the firewall.
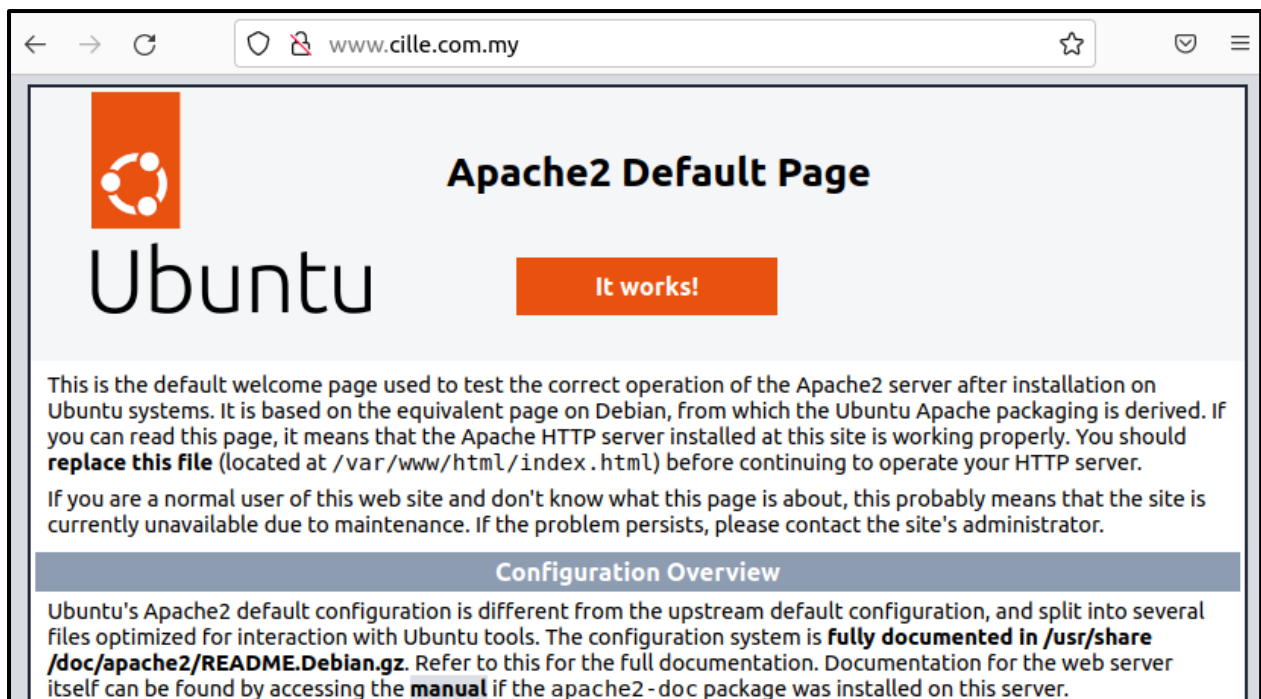

Figure 10.1: Apache Default Web Server Interface.

10.2 Installing MySQL

Now, let's install the database system to be able to store and manage data for your website. MySQL is a popular database management system used within PHP environments. To install this software run the following command:

```
$ sudo apt install -y mysql-server
```

Testing the mysql service by prompt:

```
$ sudo mysql
```

To exit the prompt, just type exit at the MySQL console.

You can secure the MySQL connection by prompt the following commands:

```
$ sudo mysql_secure_installation
```

- Password validation: Yes
- Choose password level: Low
- Remove anonymous users: Yes
- Disallow root login remotely: Yes
- Remove test database: Yes
- Reload privilege tables: Yes

10.3 Installing PHP

PHP is stand for Hypertext Preprocessor and known as the language that is powerful for making dynamic and interactive web pages. Several PHP modules will be installed together to ensure the connection between Apache, MySQL and email service that will be installed later can run smoothly. To install these packages, run the following command:

```
$ sudo apt install -y php libapache2-mod-php php-mysql php-fpm
php-gd php-common php-json php-imagick php-imap php-xml php-
mbstring php-curl php-zip php-bz2 php-intl php-ldap
```

Once the installation is finished, run the following command to confirm your PHP version:

```
$ php -v
```

10.4 Creating a Virtual Host

Apache on Ubuntu 22.04 has one virtual host enabled by default that is configured to server documents from the /var/www/html directory. This directory structure can be customs if we were hosting multiple sites instead of single site.

Create a directory for your website as follows:

```
$ sudo mkdir /var/www/cille
```

Next, assign ownership of the directory to "www-data":

```
$ sudo chown -R www-data:www-data /var/www/cille
```

Create a new configuration in Apache's sites-available directory using your preferred command-line editor. Here, we will use nano:

```
$ sudo nano /etc/apache2/sites-available/cille.conf
```

This will create a blank file. Add the following configuration in the file:

```
<VirtualHost *:80>
      ServerName cille.com.my
      ServerAlias www.cille.com.my
      ServerAdmin webmaster@localhost
      DocumentRoot /var/www/cille
      ErrorLog ${APACHE_LOG_DIR}/error.log
      CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Now, use command "a2ensite" to enable the new virtual host:

```
$ sudo a2ensite cille.conf
```

You might need to disable the default website that comes with Apache because it may override your virtual host if you are not using a custom domain name. To disable Apache's default website, run the following:

```
$ sudo a2dissite 000-default.conf
```

To test either there is any error in your file configuration and to reload Apache so the changes can take effect:

```
$ sudo apache2ctl configtest
```

```
$ sudo service apache2 reload
```

Your website is now active, but the web root "/var/www/cille" is still empty. Create an index.html file in that location to test the virtual host works as expected:

```
$ sudo nano /var/www/cille/index.html
```

Include the following HTML code content in this file:

```
<html>
    <head>
        <title>cille.com.my</title>
    </head>
    <body>
      <h1> Hello Everyone! </h1>
      <p> This is the landing page of <strong>cille.com.my</strong>.</p>
    </body>
</html>
```

Save and close the file and the content at your client browser by prompt the domain name or IP address in url:

```
http://www.cille.com.my

OR

http://172.16.24.5
```


Figure 10.2: Output from testing page created.

10.5 Testing PHP Processing on Web Server

Create a new file named **info.php** inside your custom web root folder:

```
$ sudo nano /var/www/cille/info.php
```
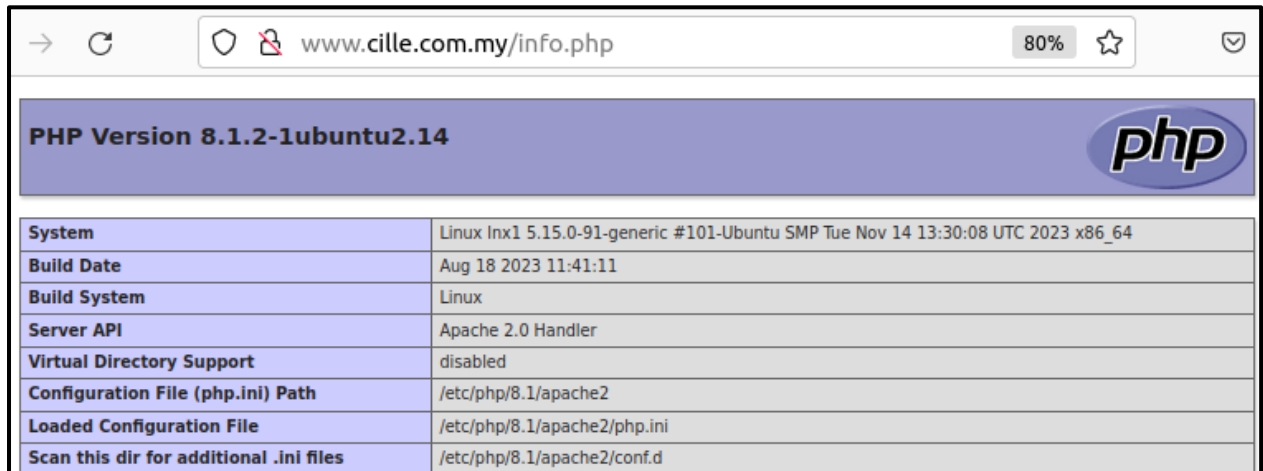
Input PHP valid code in the blank file that you have just created:

```
<?php

phpinfo();

?>
```

When finished, save and close the file.

To test the script, open your client web browser and access your **server's domain name** or **IP address** by typing the following URL:

```
http://www.cille.com.my/info.php
```


Figure 10.3: PHP information page.

This page is providing information about your server from the perspective of PHP. It is useful for debugging and to ensure the settings are being applied correctly. After checking the relevant information, it is best to remove the file you created as it contains sensitive information about your PHP environment.

Use command below to remove the info.php file:

```
$ sudo rm /var/www/cille/info.php
```

10.6 Testing Database Connection from PHP (optional)

Now, let's create a database and a table to test whether PHP is able to connect to MySQL and execute database queries. You can create a test table with test data and query for its contents from a PHP script.

Create a database named **cilledb** and a user named **cille_user**. You can replace these names with any names that you like.

First, connect to the MySQL console using the root account.

```
$ sudo mysql
```

Create database in MySQL prompt:

```
> CREATE DATABASE cilledb;
```

Create user in MySQL prompt:

```
> CREATE USER 'cille_user'@'%' IDENTIFIED BY 'Skills39';
OR
> ALTER USER 'root'@'localhost' IDENTIFIED WITH
mysql_native_password BY 'password';
> GRANT ALL ON cilledb.* TO 'cille_user'@'%';
> exit
```

Test user to login into MySQL.

```
$ mysql -u cille_user -p
```

Add table in MySQL prompt:

```
> SHOW DATABASES;
> CREATE TABLE cilledb.todo_list (
    item_id INT AUTO_INCREMENT,
    content VARCHAR(255),
    PRIMARY KEY(item_id)
    );
```

Insert several test data into the table created.

```
> INSERT INTO cilledb.todo_list (content) VALUES ("Configure
DNS Service");
> INSERT INTO cilledb.todo_list (content) VALUES ("Configure
NFS Service");
> INSERT INTO cilledb.todo_list (content) VALUES ("Configure
LAMP Service");
```

Verify data from MySQL queries in terminal prompt.

```
> SELECT * FROM cilledb.todo_list;
> exit
```

Create file to use PHP to read data in MySQL server

```
$ sudo nano /var/www/cille/todo_list.php
```

Input the code into the file todo_list.php. You can change the information in the script based on your database name, username, password and table by referring to the names that you created in MySQL database.

```php
<?php
$user = "cille_user";
$password = "Skills39";
$database = "cilledb";
$table = "todo_list";

try {
  $db = new
PDO("mysql:host=localhost;dbname=$database",$user,$password);
  echo "<h2>TODO</h2><ol>";
  foreach($db->query("SELECT content FROM $table") as $row) {
    echo "<li>" . $row['content'] . "</li>";
  }
  echo "</ol>";
} catch (PDOException $e) {
  echo "Error!: " .$e->getMessage() . "<br/>";
  die();
}
?>
```

Now, let's test the output either successful or not in the web browsers.

```
http://cille.com.my/todo_list.php
```


Figure 10.4: Output from todo_list.php in a web browser.

TOPIC 11
EMAIL SERVER

11.0 EMAIL SERVER

Mail server is best when you need a private email communication that you can use to send and read emails in your network and it also can be access thru the Internet. In this topic, we will configure an email server from Roundcube Webmail.

Roundcube is a free open-source, full-featured webmail client written in PHP. A webmail is a mail client in your browser. Instead of reading and sending emails from a desktop mail client like Mozilla Thunderbird, Microsoft Outlook, Zimbra or IBM Lotus; you can access the email from a web browser.

To follow this tutorial, it's assumed that you already done install:

- DNS service (refer to topic 6: DNS)
- LAMP stack (refer to topic 10: Web Server)

Now let's proceed to configure email server.

11.1 Configure Postfix SMTP server

Postfix is a Mail Transfer Agent (MTA) use to send email to the client mailbox from email client software to its intended recipient. The MTA agent communicate with one another using the Simple Mail Transfer Protocol (SMTP) protocol port 25.

First, installed the postfix package using this command:

```
$ sudo apt install -y postfix
```

Postfix file configuration is located in **/etc/postfix/main.cf**. To reconfigure or modify the configuration file in postfix service, you can run this command:

```
$ sudo dpkg-reconfigure postfix
```

- Choose > Internet Site
- System mail name: cille.com.my
- Root and postmaster mail recipient: postmaster
- Other destinations to accept mail: cille.com.my, lnx1.cille.com.my, mail.cille.com.my, localhost
- Force synchronous updates on mail queue: No
- Local networks: 172.16.24.0/24

Other's config just use default.

Restart service postfix from terminal after done editing the configuration file.

```
$ sudo service postfix reload
```

Add two user to test the email service.

```
$ sudo adduser josh

$ sudo adduser chia
```

Verify the email agent Mail Transfer Agent (MTA) are running successfully using terminal prompt. Test SMTP-AUTH protocols by telnet the mail address domain.

Figure 11.1: Testing SMTP protocol communication from CLI.

11.2 Configure Dovecot IMAP server

Dovecot act as a Mail Delivery Agent (MDA) in email service. This agent handles a message for delivery to the point where it can be read by an email client application or in proper user's mailbox. It supports two protocol to receive email message in client mailbox which is Internet Message Access Protocol (IMAP) using port 143 and Post Office Protocol version 3 (POP3) using port 110.

At the command prompt, type the following to begin installation of postfix:

```
$ sudo apt install -y dovecot-common
$ sudo apt install -y dovecot-imapd dovecot-pop3
```

After installation, we will edit certain file in the dovecot configuration file located at **/etc/dovecot/conf.d** to allow the service run smoothly.

```
$ cd /etc/dovecot/conf.d
```

Suggested to backup file 10-master.conf and 10-auth.conf before editing. Use the following command to do a backup.

```
$ sudo cp 10-master.conf{,.bk}
$ sudo cp 10-auth.conf{,.bk}
```

Edit the Dovecot 10-master.conf file by removing the hash symbol (#) and add several information as follows.

```
...
 service imap-login {
   inet_listener imap {
     port = 143
   }
...
 # Postfix smtp-auth
 unix_listener /var/spool/postfix/private/auth {
    mode = 0660
    user = postfix
    group = postfix
 }
```

Edit the Dovecot 10-auth.conf entry as follow:

```
auth_mechanisms = plain login
```

Restart service dovecot from terminal after done editing the configuration file.

```
$ sudo service dovecot restart
```

11.3 Create a database and user for Roundcube

Now, let's create a database and user for Roundcube webmail in MySQL. This is required to allow the email service to store record email in the database.

```
$ sudo mysql -u root -p
```

Create a database name **roundcubedb** in the MySQL. You can set any name but remember to replace it in the command:

```
> CREATE DATABASE roundcubedb;
```

Next, create a user that will be assign to access the database and allow the user the privilege to modify the database. The user that will be given access to roundcubedb is "**roundcb**" and its password is "**Skills39**". You can change the user and the password respectively.

```
> CREATE USER 'roundcb'@'localhost' IDENTIFIED BY 'Skills39';

> GRANT ALL PRIVILEGES ON roundcubedb . * TO
'roundcb'@'localhost';

> FLUSH PRIVILEGES;

> exit
```

11.4 Install and Configure Roundcube Webmail

After we done with the prerequisite required for Roundcube, let's go ahead and install the latest version of Roundcube Webmail. It is recommended to always check the latest stable version of Roundcube from https://github.com/roundcube/roundcubemail/releases/.

Download Roundcube tarball on the server:

```
$ cd /var/www

$ wget https://github.com/roundcube/roundcubemail/releases
/download/1.6.0/roundcubemail-1.6.0-complete.tar.gz
```

Extract the tarball package and rename it to roundcube. Make sure you are in the /var/www directory.

```
$ sudo tar -xvf roundcubemail-1.6.0-complete.tar.gz

$ sudo mv roundcubemail-1.6.0-complete roundcube
```

Change the ownership of the directory to www-data user.

```
$ sudo chown -R www-data:www-data /var/www/roundcube
```

Now, we will initiate the database by executing the following command:

```
$ sudo mysql -u roundcb -p roundcubedb <
/var/www/roundcube/SQL/mysql_initial.sql
```

11.5 Create Apache Virtual Host for Roundcube

Now, we will create the virtual host for Roundcube Webmail in the apache service.

```
$ sudo nano/etc/apache2/sites-available/roundcube.conf
```

Input the following in the file roundcube.conf.

```
<Virtualhost *:80>

    ServerName cille.com.my

    ServerAlias mail.cille.com.my

    DocumentRoot /var/www/roundcube


    ErrorLog ${APACHE_LOG_DIR}/roundcube_error.log

    CustomLog ${APACHE_LOG_DIR}/roundcube_access.log combined


    <Directory />

        Options FollowSymLinks

        AllowOverride All

    </Directory>


    <Directory /var/www/roundcube/>
```

```
        Options FollowSymLinks MultiViews

        AllowOverride All

        Order allow,deny

        Allow from all

    </Directory>
</Virtualhost>
```

After save and exit from the file, use the following command to enable the virtual host:

```
$ sudo a2ensite roundcube.conf
```

Restart or reload the apache2 service as instruction to activate the new configuration:

```
$ sudo service apache2 reload
$ sudo service apache2 restart
```

11.6 Testing Email

Now, open the client browser and type the following url: mail.cille.com.my/installer.

You also can use ip address of the server as an option. Ensure to identify if anything is prompt as "**NOT OK**" instead of "**OK**". You need to troubleshoot the problem first before proceed to the next configuration.
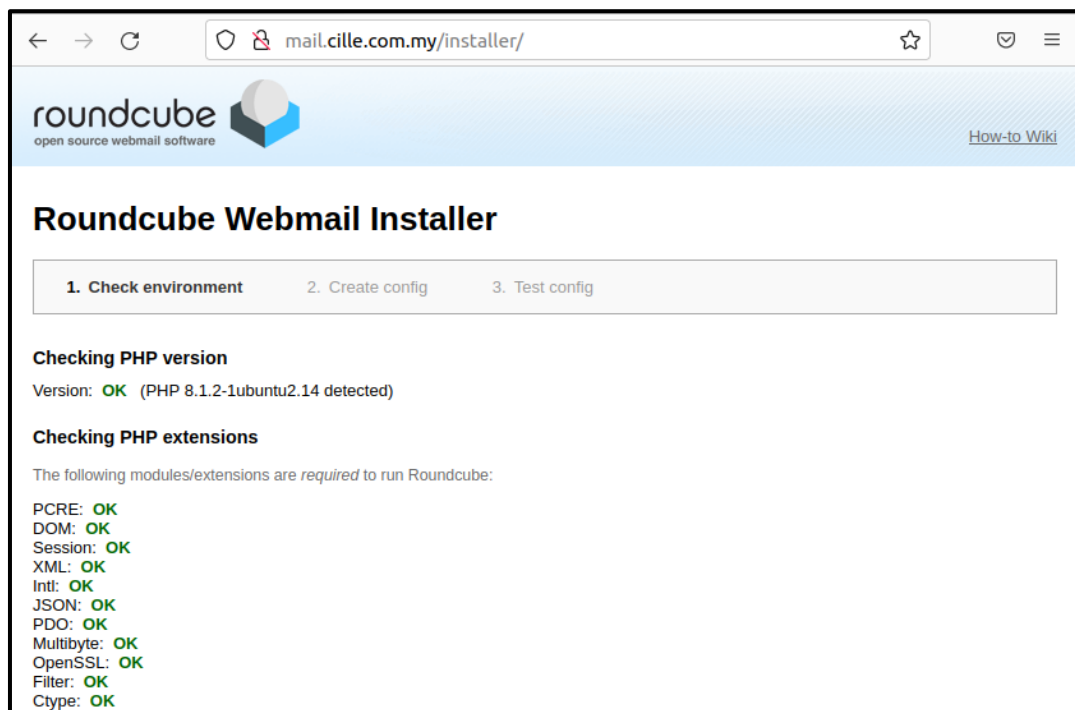

Figure 11.2: Configure the roundcube from web browser.

Input the following information for database setup by referring to your previous information.

Figure 11.3: Configure the database setup from web browser.

After complete the installation it is recommended to delete the installer page. This is to ensure that unauthorized user will not able to modify the configuration that has been successfully running.

```
$ sudo rm -rf /var/www/roundcube/installer
```

You have successfully installed and configured Roundcube webmail on your Ubuntu Server 22.04. Please verify the service by testing send and receive email from users.
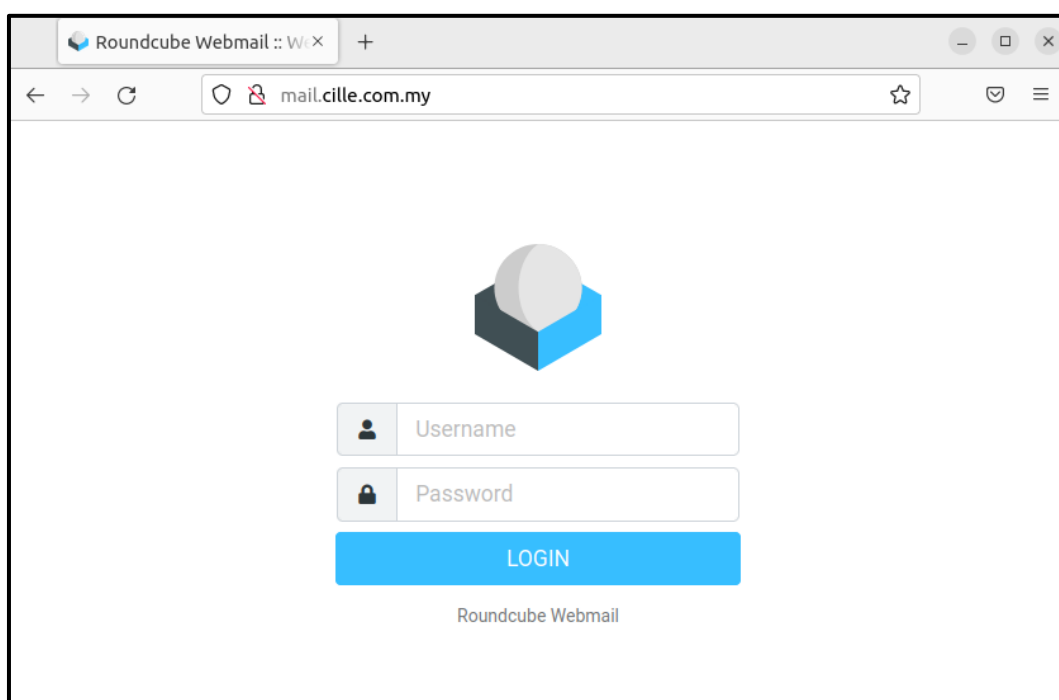

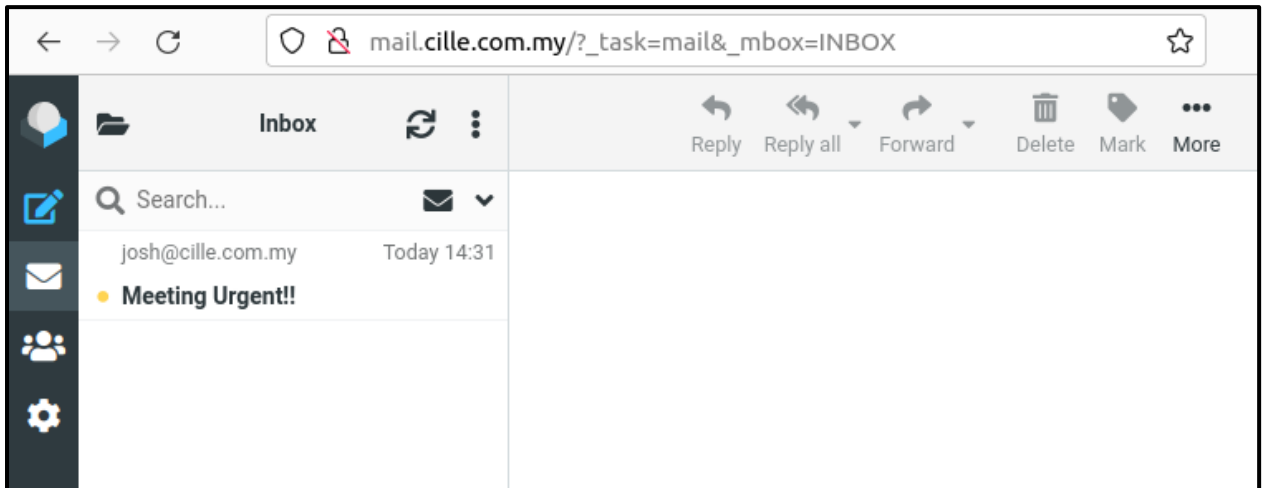**Figure 11.4: Main page of roundcube email before user login.**

Figure 11.5: Interface of the roundcube email went user login and read mail.

TOPIC 12
REFERENCE

12.0 REFERENCE

[1]   Brent, M. (2023, December 11). Basics linux/unix commands with examples & syntax
         (list). Guru99. https://www.guru99.com/must-know-linux-commands.html

[2]   Brent, M. (2023, December 30). File Permissions in Linux/Unix: how to read, write, &
         change. https://www.guru99.com/file-permissions.html

[3]   Ostechnix (2023, August 10). A Beginners Guide to Understanding Linux File
         Permissions. https://ostechnix.com/linux-file-and-directory-permissions/

[4]   Linuxize (2021, April 30). Understanding Linux File Permissions.
         https://linuxize.com/post/understanding-linux-file-permissions/

[5]   McBrien, S. (2023, January 30). Linux file permissions explained.
         https://www.redhat.com/sysadmin/linux-file-permissions-explained

[6]   The Linux Foundation (2022, January 6). Classic SysAdmin: Understanding Linux File
         Permissions. https://www.linuxfoundation.org/blog/blog/classic-sysadmin-
         understanding-linux-file-permissions

[7]   Jun, C.M. (2023, May 2). The /etc/resolv.conf. https://www.baeldung.com/linux/etc-
         resolv-conf-file

[8]   Patel, S. & Daga, D. (2023, July 19). How to install and configure samba on Ubuntu 22.04.
         https://vegastack.com/tutorials/how-to-install-and-configure-samba-on-ubuntu-22-
         04/

[9]   Ndungu, F. (2020, July 10). How to Install Postfix, Dovecot, and Roundcube on Ubuntu
         20.04. https://docs.vultr.com/how-to-install-postfix-dovecot-and-roundcube-on-
         ubuntu-20-04

[10]   Rahul (2023, January 19). How to configure static ip address on Ubuntu 22.04.
         https://tecadmin.net/how-to-configure-static-ip-address-on-ubuntu-22-04/

[11]   Md. Shah, W. & Anawar, S. (2019). Debian 9 Stretch: Basic Administration. UTeM
         Publication.

[12]   Kumar. M. (2020). Linux Guide for Beginners: The Best Guide to the Linux Operating
         System and Linux Commands. Kindle. (ASIN: B08MCLWCSS)

[13]   Christopher. N. (2019). Linux ® Bible (10th Edition). New York, US: John Wiley & Sons Inc.
         (ISBN: 1119578884)

[14]   Rahul (2023, January 3). How to install roundcube webmail on Ubuntu 20.04.
         https://tecadmin.net/how-to-install-roundcube-webmail-on-ubuntu-20-04/

[15]   GeeksforGeeks (2023, May 22). Linux history. https://www.geeksforgeeks.org/linux-
         history/

[16]   Berg, N. (2020, June 14). Introduction to Ubuntu linux.
         https://www.ubuntudocs.com/intro-linux/

HANDS-ON-GUIDE: UBUNTU 22.04 SERVER ADMINISTRATION