



Bicol University Polangui  
Polangui, Albay  
Computer Studies Department

IT-123

# System Administration and Maintenance

FINAL LABORATORY REQUIREMENT

Submitted by:  
Jeremy O. Rellama  
Marc Elijah N. Riofrio  
**BSIT 4A Student**

Submitted to:  
Guillermo V. Red, Jr. DIT  
**PROFFESOR**



# Table Of Contents

**Laboratory Week1** - Installing Essential Tools for System Administration **3-7**

---

**Laboratory Week2** - Installing and Configuring Windows & Linux Virtual Machines in VirtualBox **8-16**

---

**Laboratory Week 3** - User Group Management in Windows & Linux **17-27**

---

**Laboratory Week4** - File Systems and Storage Management **28-35**

---

**Laboratory Week6** - Network Configuration and Management **36-41**

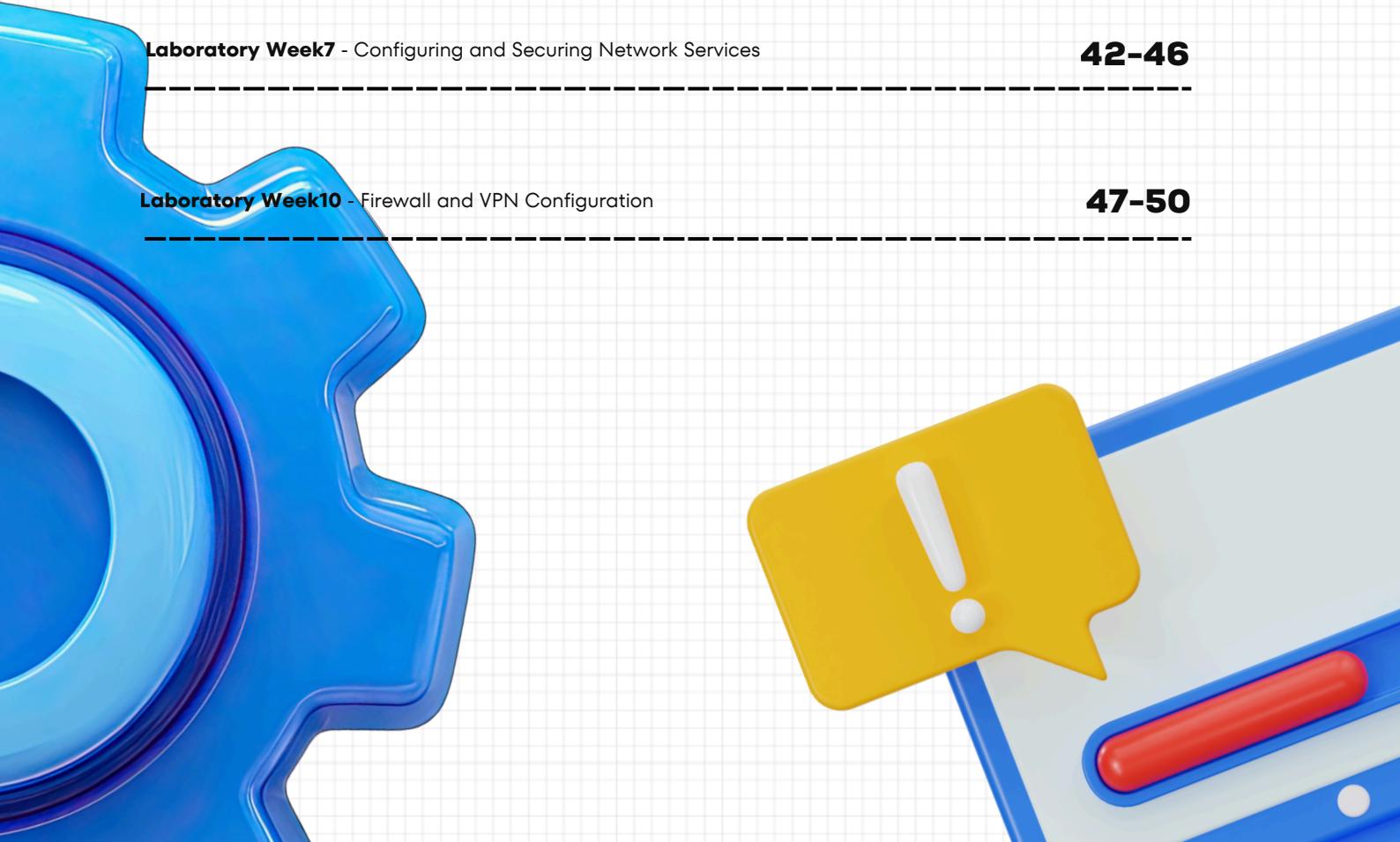
---

**Laboratory Week7** - Configuring and Securing Network Services **42-46**

---

**Laboratory Week10** - Firewall and VPN Configuration **47-50**

---





Bicol University Polangui  
Polangui, Albay  
Computer Studies Department

IT-123

# System Administration and Maintenance

Laboratory Week1 - Installing Essential Tools for System Administration

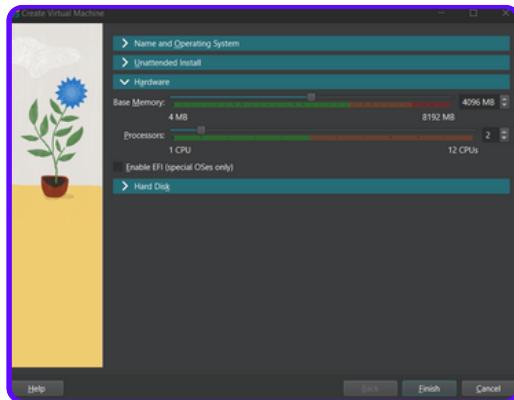
Submitted by:  
Jeremy O. Rellama  
Marc Elijah N. Riofrio  
**BSIT 4A Student**

Submitted to:  
Guillermo V. Red, Jr. DIT  
**PROFFESOR**



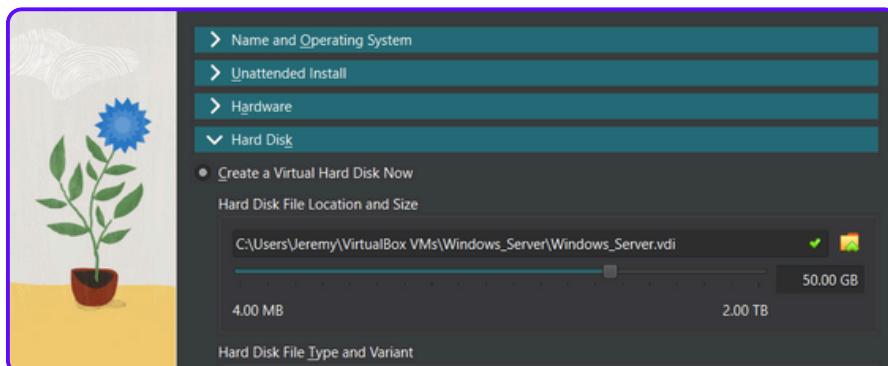


# Windows Server



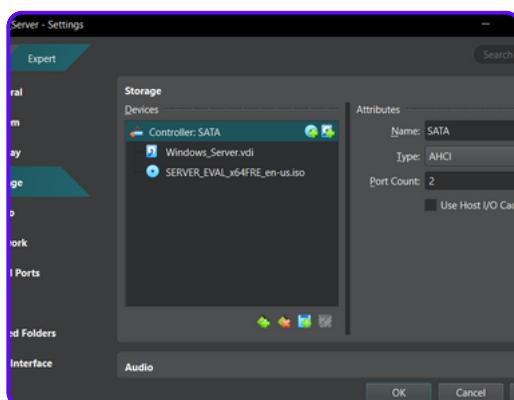
## Step 1. Create the VM shell

We create a new VM by clicking **New**. In the setup window, we type the name **Windows\_Server**, set the type to **Microsoft Windows**, and select either **Windows 2019 (64-bit)** or **Windows 2022 (64-bit)** as the version. We then allocate resources by setting the memory size to **4096 MB (4 GB)** and assigning **2 CPU cores** if available.



## Step 2. Setting Up Virtual Hard Disk

We create a new virtual hard disk in **VDI** format, select **Dynamically allocated** so it only uses space as needed, and set the size to **50 GB**.

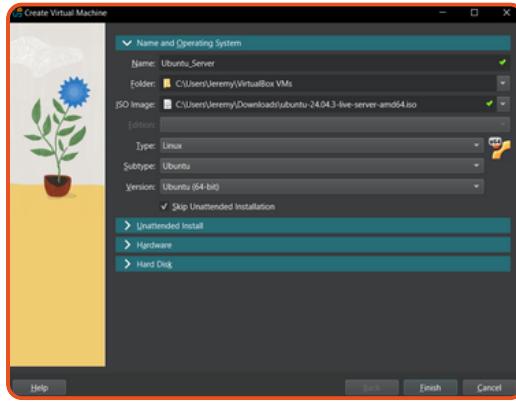


## Step 3: Attaching ISO File

After the VM is created, we go to **Settings → Storage**, select the empty optical drive, and load the Windows Server ISO file.

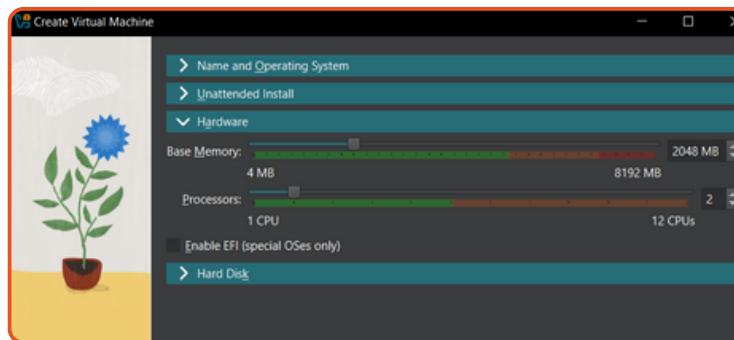


# **Ubuntu Server**



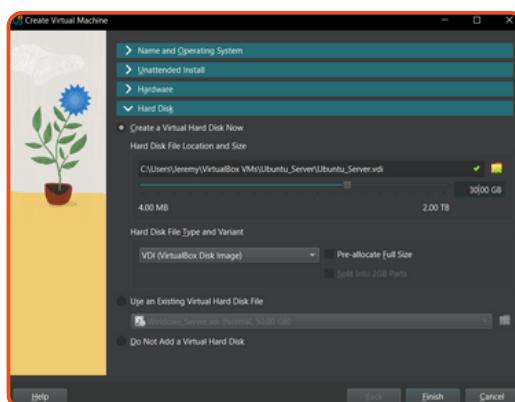
## Step 1: Creating a New Virtual Machine

Next, we open Oracle VM VirtualBox and click on New to create our virtual machine. We give it the name Ubuntu\_Server, set the type to Linux, and choose Ubuntu (64-bit) as the version.



## Step 2: Allocating System Resources

Next, we open Oracle VM VirtualBox and click on New to create our virtual machine. We give it the name Ubuntu\_Server, set the type to Linux, and choose Ubuntu (64-bit) as the version.



## Step 3: Creating the Virtual Hard Disk

We create a virtual hard disk by choosing VDI (VirtualBox Disk Image) as the file type, selecting Dynamically allocated so it grows only as needed, and setting the disk size to 30 GB. This gives the server enough space for the operating system and additional tools.



Bicol University Polangui  
Polangui, Albay  
Computer Studies Department

IT-123

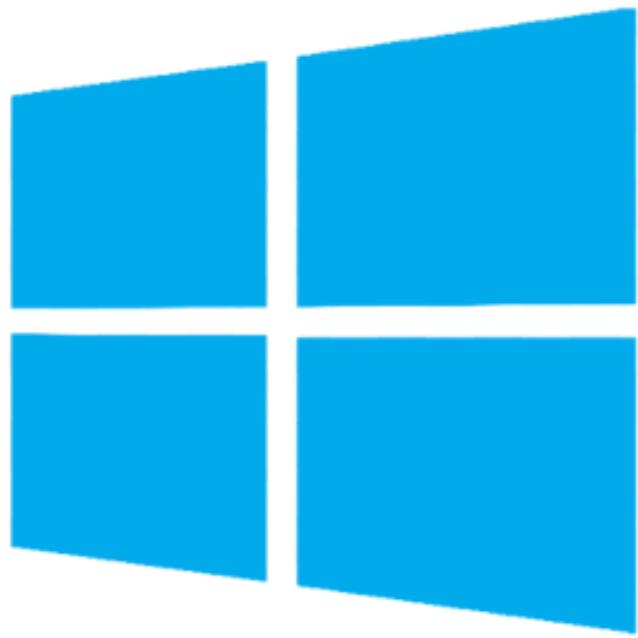
# System Administration and Maintenance

Laboratory Week2 - Installing and Configuring Windows & Linux Virtual Machines in VirtualBox

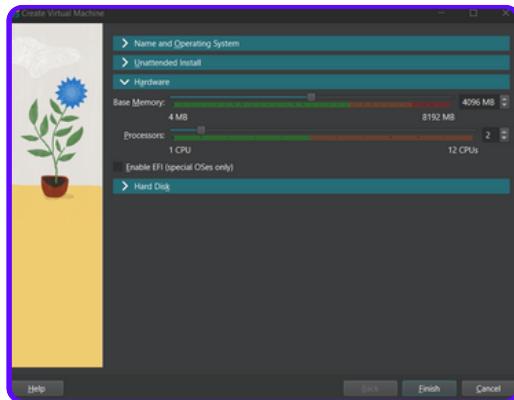
Submitted by:  
Jeremy O. Rellama  
Marc Elijah N. Riofrio  
**BSIT 4A Student**

Submitted to:  
Guillermo V. Red, Jr. DIT  
**PROFFESOR**



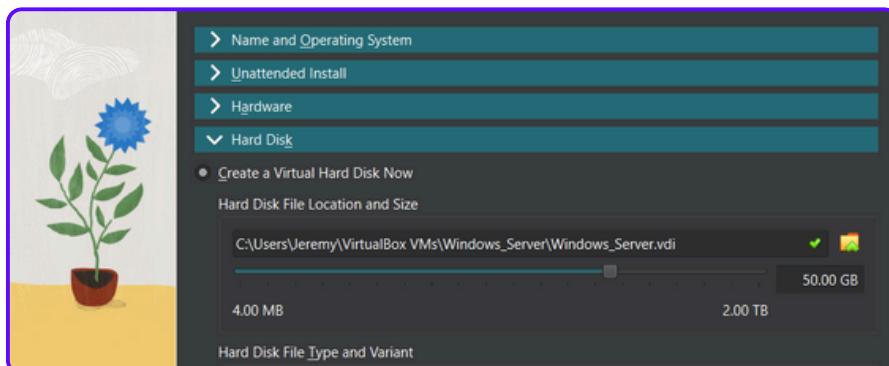


# Windows Server



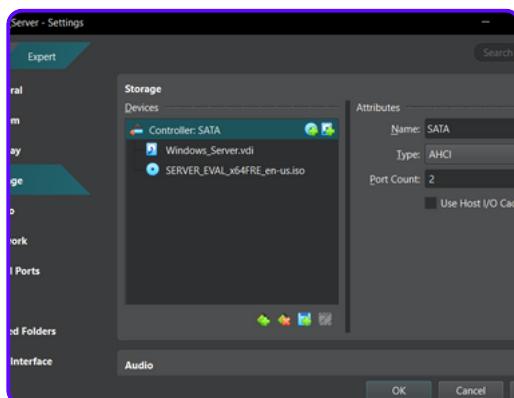
## Step 1. Create the VM shell

We create a new VM by clicking New. In the setup window, we type the name Windows\_Server, set the type to Microsoft Windows, and select either Windows 2019 (64-bit) or Windows 2022 (64-bit) as the version. We then allocate resources by setting the memory size to 4096 MB (4 GB) and assigning 2 CPU cores if available.



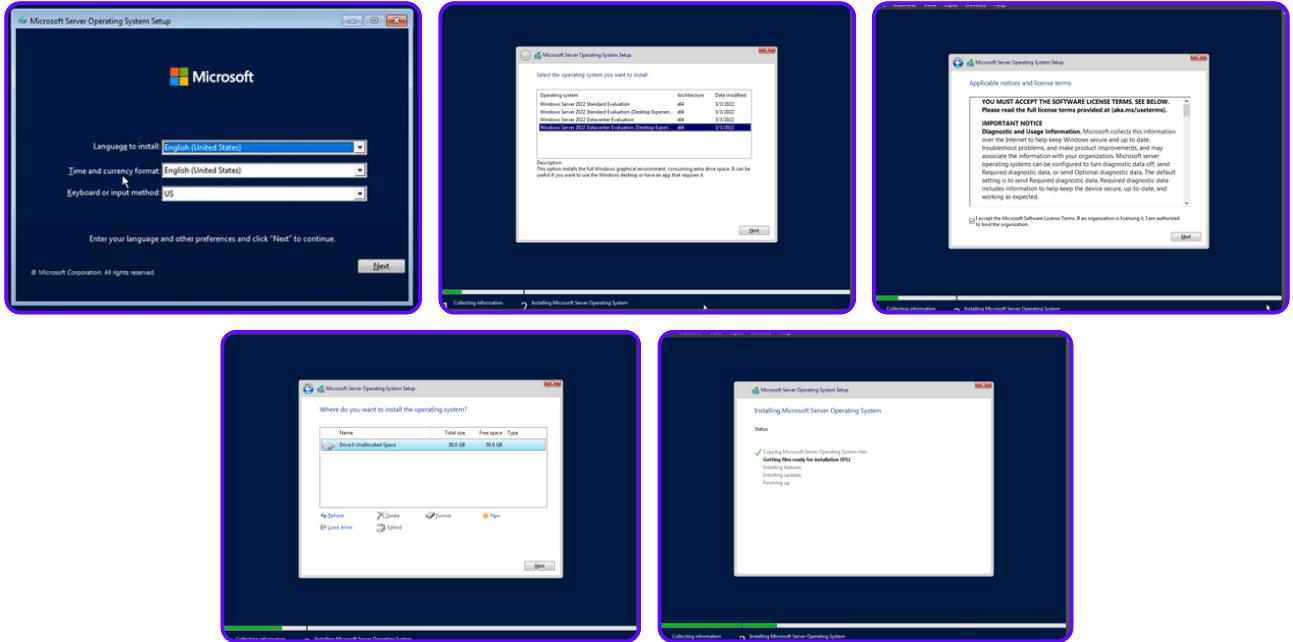
## Step 2. Setting Up Virtual Hard Disk

We create a new virtual hard disk in VDI format, select Dynamically allocated so it only uses space as needed, and set the size to 50 GB.



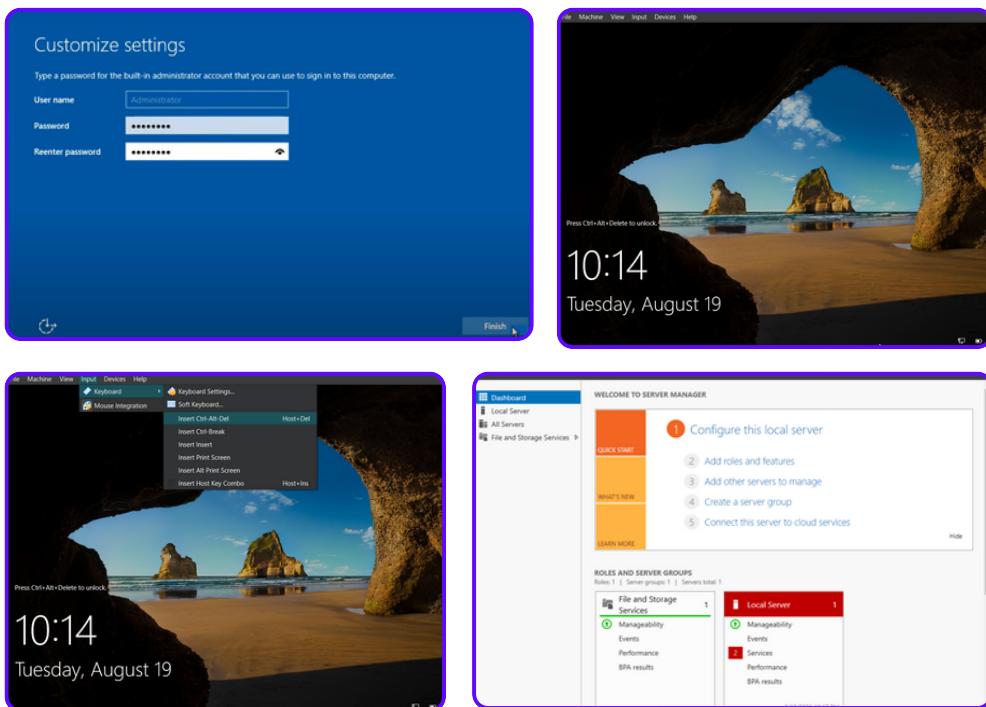
## Step 3: Attaching ISO File

After the VM is created, we go to Settings → Storage, select the empty optical drive, and load the Windows Server ISO file.



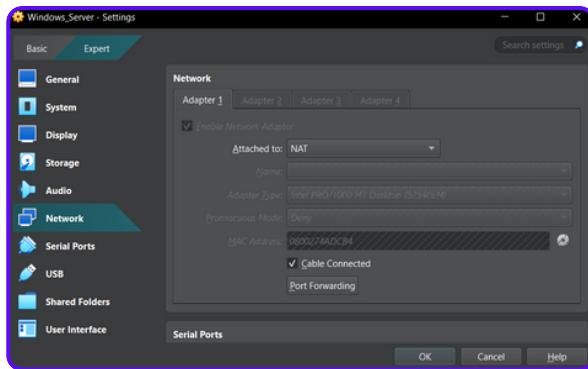
## Step 4: Installing Windows Server

We start the VM, and it boots from the ISO. We follow the Windows Server installation wizard by selecting language and region, clicking install. We accept the license, choose custom installation, and install the OS on the 50 GB virtual hard disk.



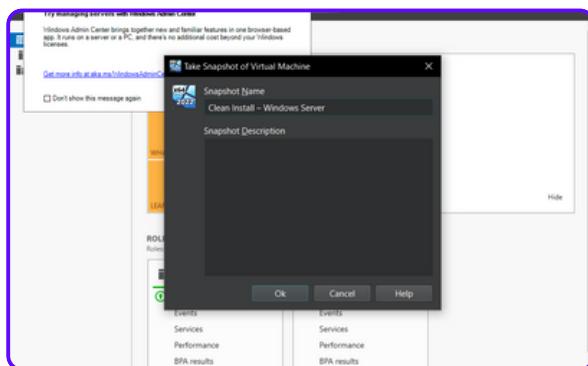
## Step 5: Setting Administrator Account

Once installation is complete, we set the Administrator password to P@ssw0rd123. To log in, we use Input → Keyboard → Insert Ctrl+Alt+Del in VirtualBox, then enter the password to access the Windows Server desktop.



## Step 6: Enabling Network

We also configure networking under Settings → Network, choosing either NAT for basic internet or Bridged Adapter to make the VM act like a real computer on the network.

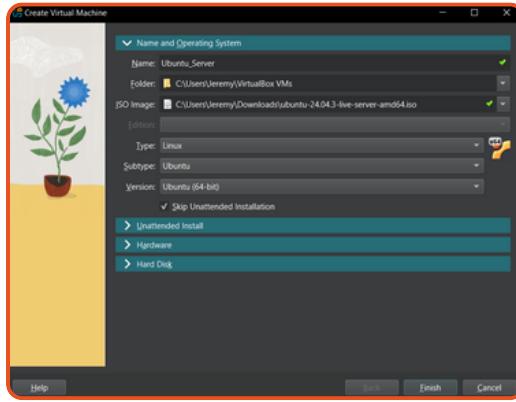


## Step 7: Taking a Snapshot

After the installation finishes and the VM is working correctly, we take a VirtualBox snapshot. We go to Machine → Take Snapshot, name it Clean Install – Windows Server, and save it. This snapshot allows us to quickly roll back to a fresh installation state whenever we need it for future labs.

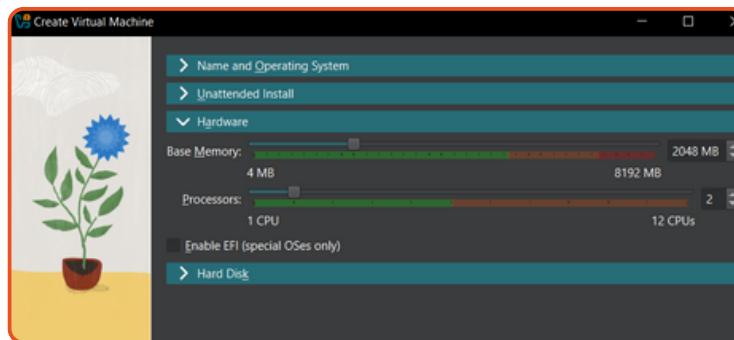


# **Ubuntu Server**



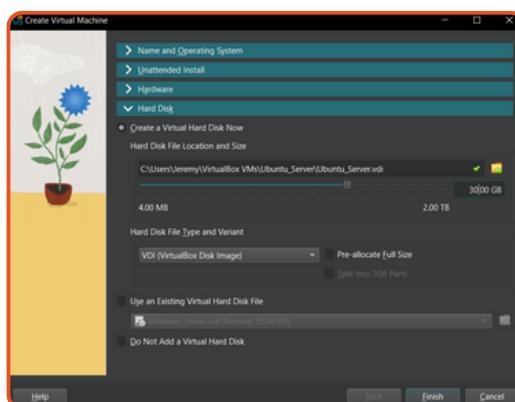
## Step 1: Creating a New Virtual Machine

Next, we open Oracle VM VirtualBox and click on New to create our virtual machine. We give it the name Ubuntu\_Server, set the type to Linux, and choose Ubuntu (64-bit) as the version.



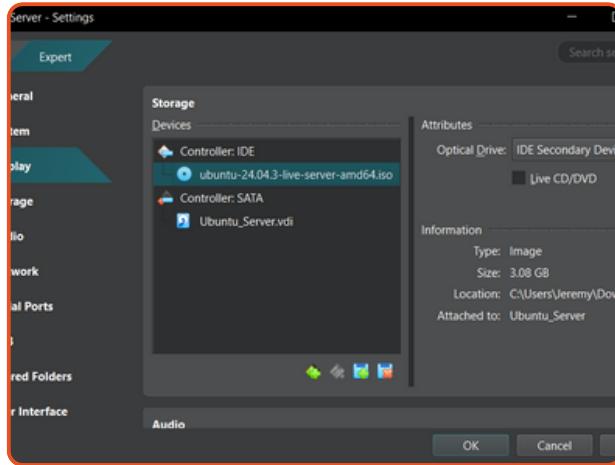
## Step 2: Allocating System Resources

Next, we open Oracle VM VirtualBox and click on New to create our virtual machine. We give it the name Ubuntu\_Server, set the type to Linux, and choose Ubuntu (64-bit) as the version.



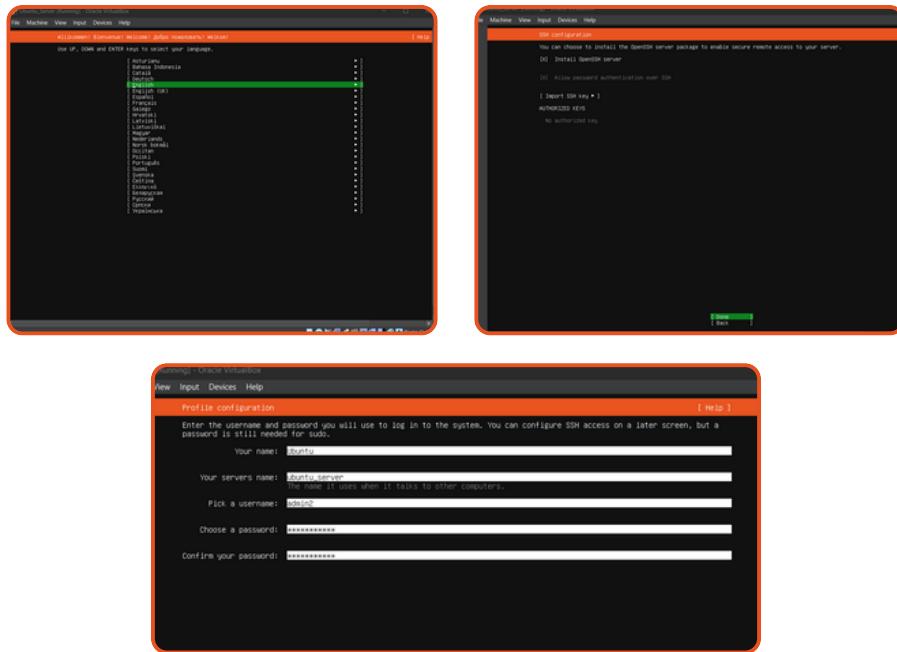
## Step 3: Creating the Virtual Hard Disk

We create a virtual hard disk by choosing VDI (VirtualBox Disk Image) as the file type, selecting Dynamically allocated so it grows only as needed, and setting the disk size to 30 GB. This gives the server enough space for the operating system and additional tools.



## Step 4: Attaching the Ubuntu ISO

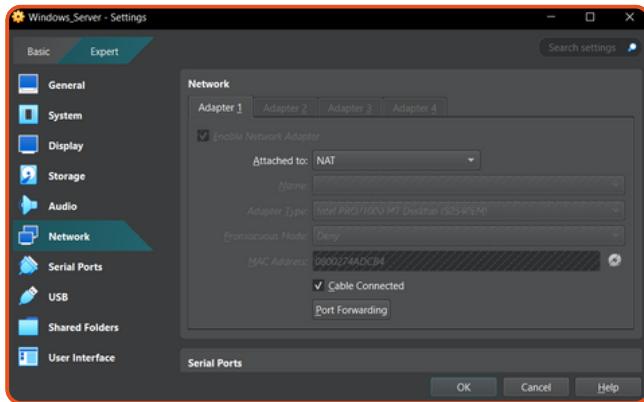
Once the VM is created, we open its Settings, go to the Storage section, and select the empty optical drive. We then choose our downloaded ISO file (ubuntu-22.04-live-server-amd64.iso) so that the VM can boot from it during startup.



## Step 5: Starting the VM and Installing Ubuntu Server

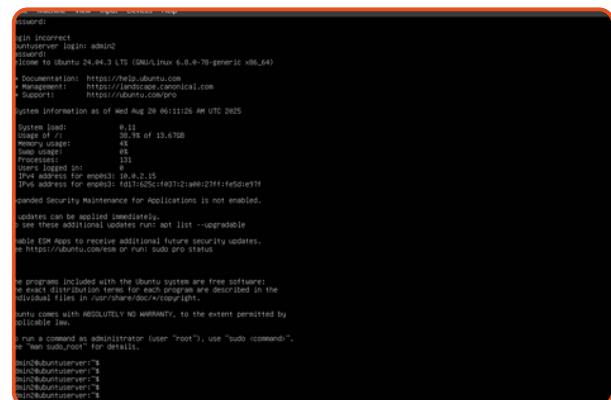
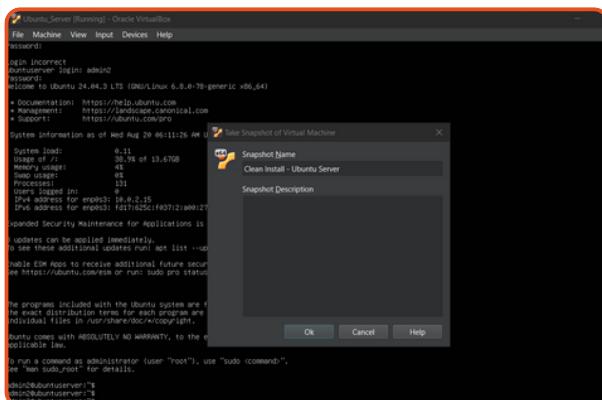
We now start the VM, and it boots into the Ubuntu Server installer. We select English as the language and proceed with the guided installation. During the process, we make sure to select the option to Install OpenSSH Server, which will allow us to connect to the VM remotely in later labs. When prompted, we create a user account with the following credentials:

- Username: admin
- Password: P@ssw0rd123



## Step 6: Starting the VM and Installing Ubuntu Server

The network configuration is handled the same way as in our Windows Server VM. In VirtualBox, we can either leave the adapter on NAT for basic internet access.



## Step 7: Taking a Snapshot

After the installation finishes and the VM is working correctly, we take a VirtualBox snapshot. We go to Machine → Take Snapshot, name it Clean Install – Ubuntu Server, and save it. This snapshot allows us to quickly roll back to a fresh installation state whenever we need it for future labs.



Bicol University Polangui  
Polangui, Albay  
Computer Studies Department

IT-123

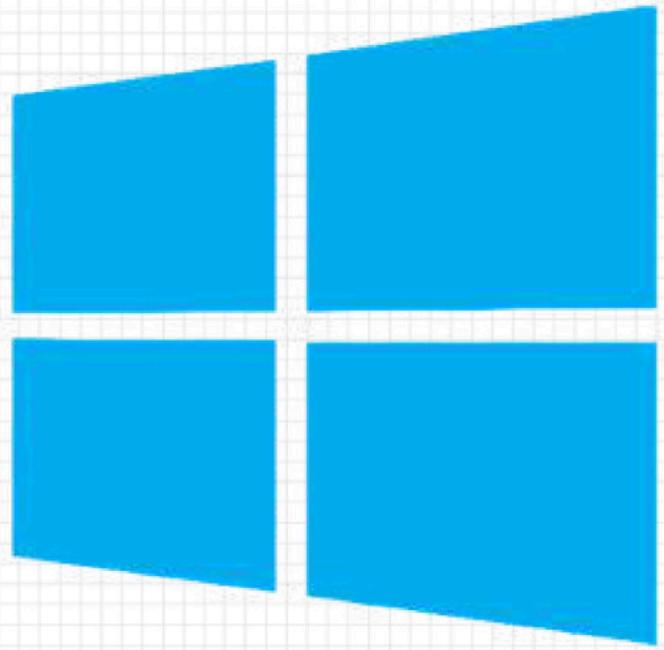
# System Administration and Maintenance

Laboratory Week3 - User and Group Management in Windows & Linux

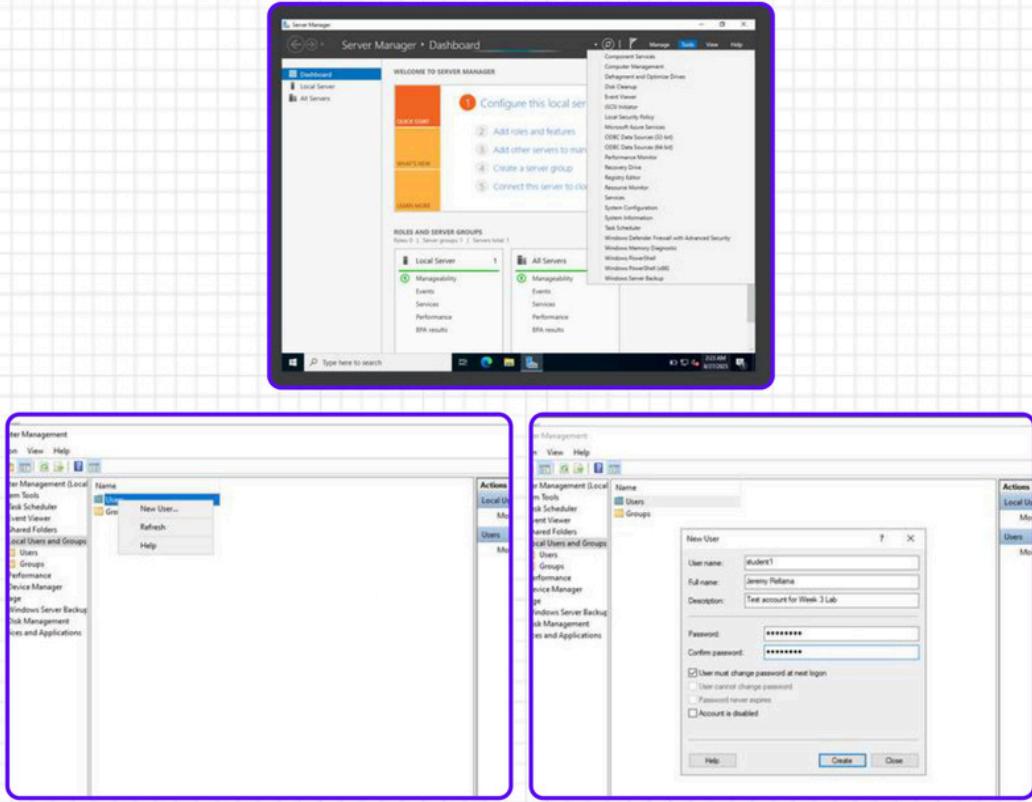
Submitted by:  
Jeremy O. Rellama  
Marc Elijah N. Riofrio  
**BSIT 4A Student**

Submitted to:  
Guillermo V. Red, Jr. DIT  
**PROFFESOR**



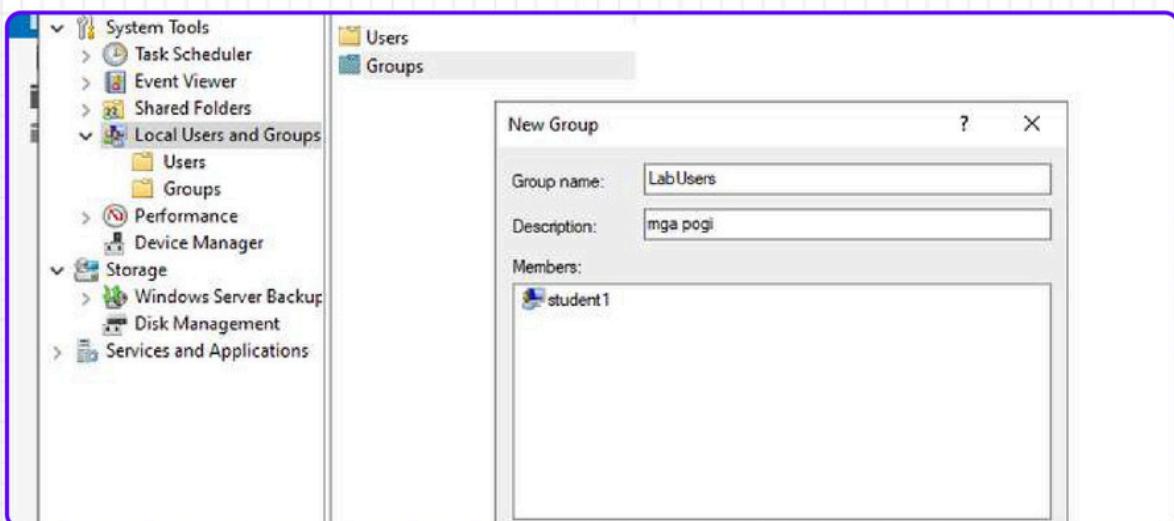


# Windows Server



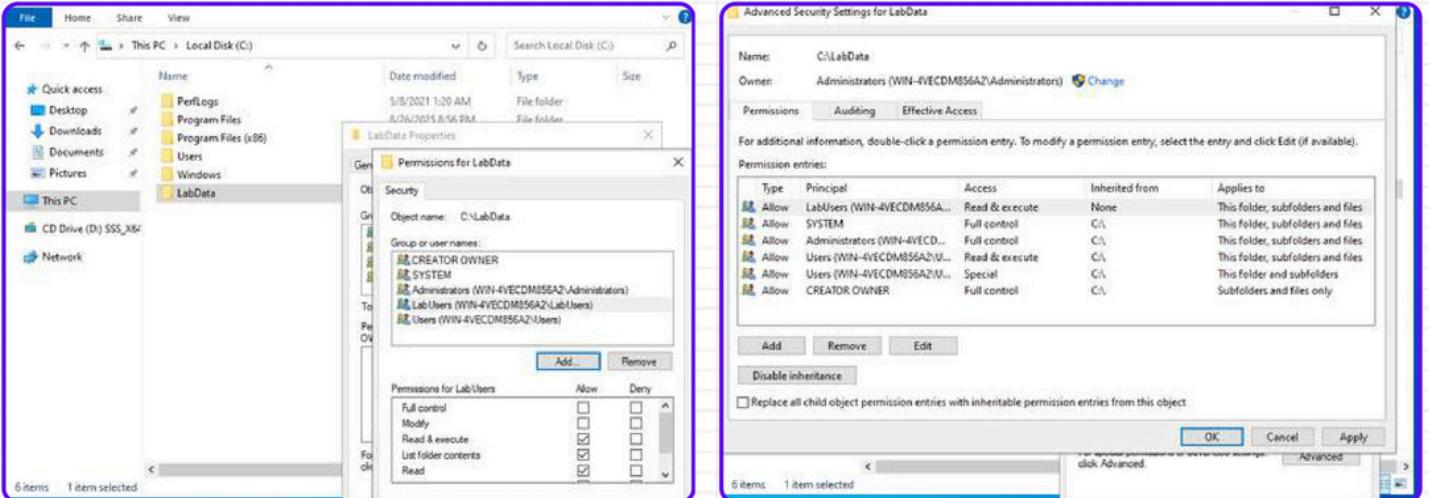
## Step 1: Creating & Modifying a New User

We begin by opening Server Manager → Tools → Computer Management. Inside the Computer Management console, we expand Local Users and Groups → Users. From there, we right-click on the Users folder and select New User. We create a new account with Username: student1 and Password: User@123. In the description field, we add: "Test account for Week 3 Lab". We also enable the option User must change password at next logon.



## Step 2: Creating a Group and Adding the User

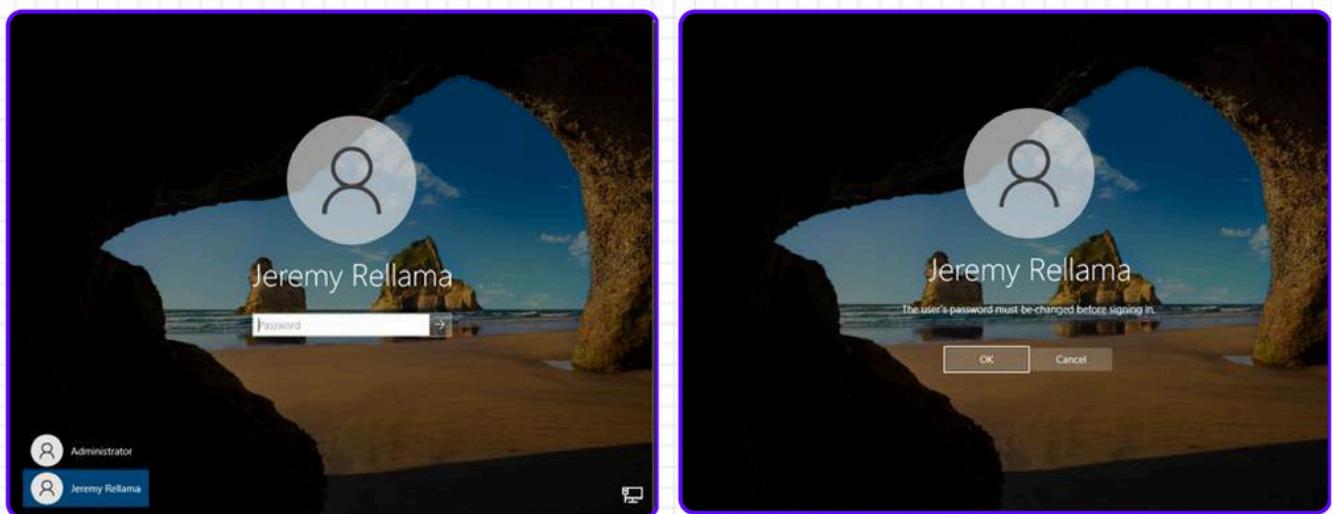
Next, we move to Local Users and Groups → Groups. We right-click the Groups folder and create a new group named LabUsers. Once the group is created, we add student1 as a member.



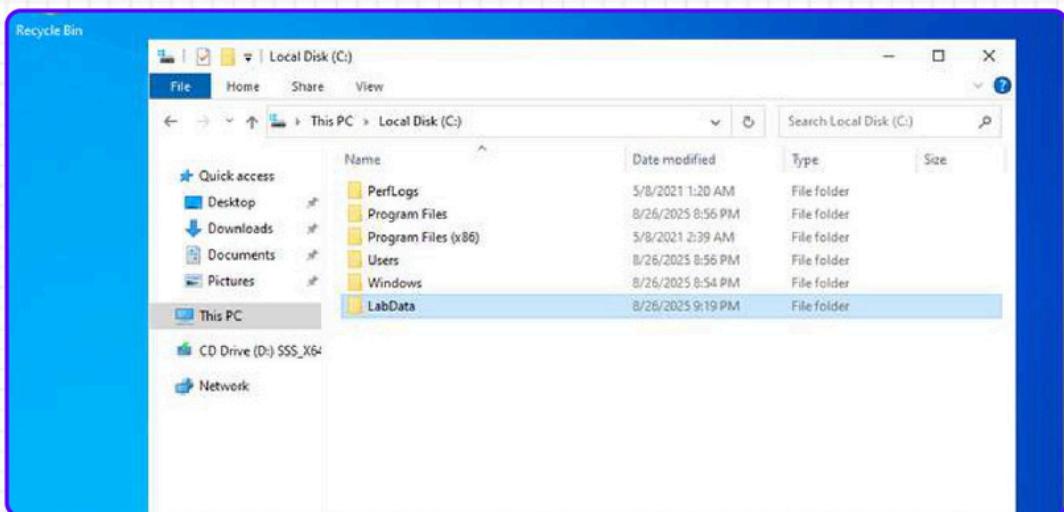
## Step 4: Applying Permissions to a Folder

We now create a folder called C:\LabData on the system drive. Right-click the folder, choose Properties → Security → Edit, and add the LabUsers group. We grant this group Read & Execute permissions only, which allows members to open and view files but prevents them from modifying content.

# VERIFICATION & TROUBLESHOOTING

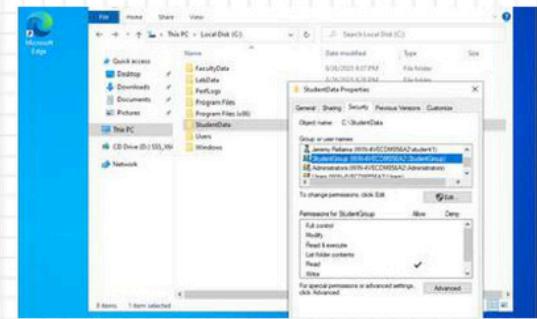
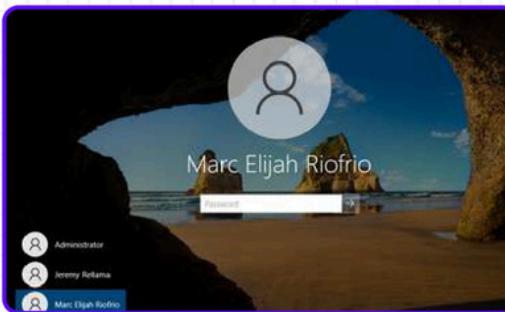


To verify our configuration, we first log out of the administrator account and log in as student1. Since we required a password change at first logon, the system will prompt us to enter a new password. This confirms that the account policy was applied correctly.

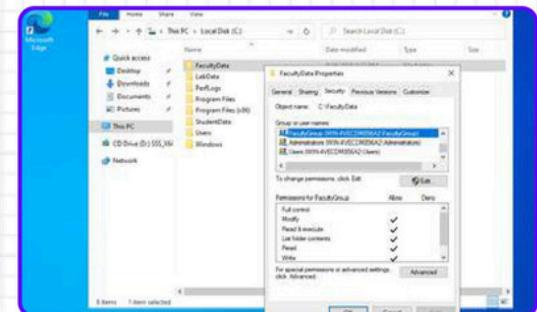
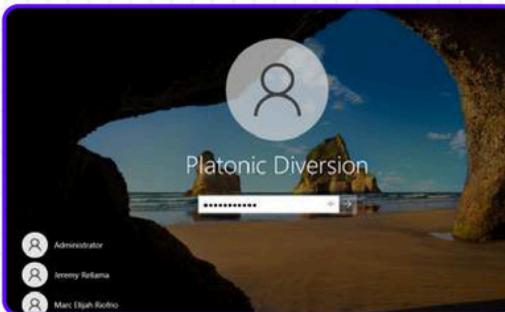


After logging in as student1, we navigate to the C:\LabData folder. The user should be able to open the folder and view its contents because the LabUsers group was granted Read & Execute permissions.

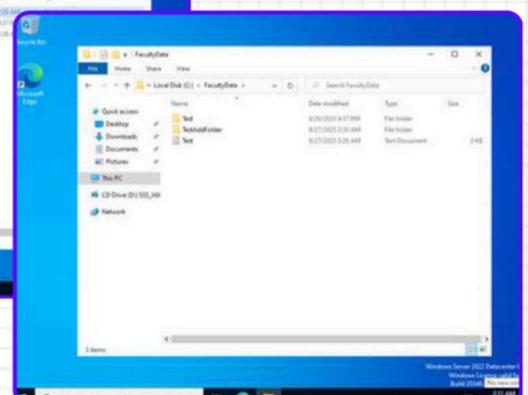
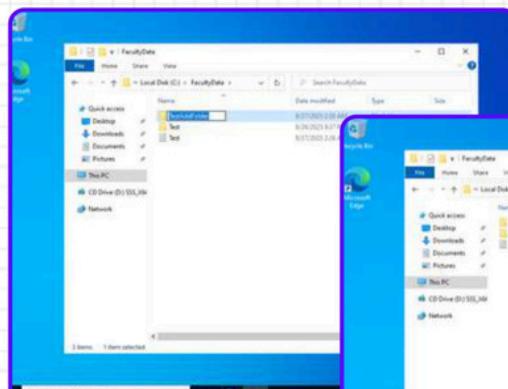
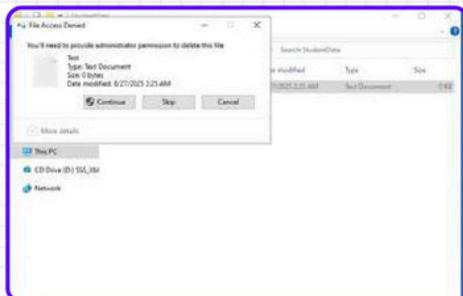
**student3**



**faculty1**



After configuring the accounts, we create two folders in the system drive: C:\FacultyData and C:\StudentData. Using the Security tab in folder properties, we grant Modify rights only to FacultyGroup for C:\FacultyData, allowing faculty members to create and edit files. For C:\StudentData, we assign Read rights only to StudentGroup, ensuring students can view the contents but cannot make changes.



**student3**

Finally, we test the configuration by logging in as each user. Faculty1 should be able to create and edit files in C:\FacultyData, but will be restricted to read-only access in C:\StudentData. Meanwhile, student3 should only be able to view files in C:\StudentData and will be denied when trying to create or modify files. These tests confirm that permissions are applied correctly.

**faculty1**



# Ubuntu Server

```
admin@rellamariofrio:~$ sudo adduser student2
[sudo] password for admin:
info: Adding user `student2' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `student2' (1001) ...
info: Adding new user `student2' (1001) with group `student2' (1001) ...
info: Creating home directory `/home/student2' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
password: password updated successfully
Changing the user information for student2
Enter the new value, or press ENTER for the default
    Full Name []: Jeremy Rellama
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [y/n]
```

## Step 1: Creating a New User

We log into Ubuntu Server and create a new user with: **sudo adduser student2**. We then set the password to **User@123**.

```
Is the information correct? [Y/n] y
info: Adding new user `student2' to supplemental / extra groups `users' ...
info: Adding user `student2' to group `users' ...
admin@rellamariofrio:~$ sudo usermod -c "Test account for week 3 Lab" student2
admin@rellamariofrio:~$ sudo groupadd labusers
admin@rellamariofrio:~$ sudo usermod -aG labusers student2
admin@rellamariofrio:~$
```

## Step 2: Modifying the User and Creating a Group

After creating the user student2, we add a description to the account using: **sudo usermod -c "Test account for Week 3 Lab" student2**. Next, we create a new group named labusers with: **sudo groupadd labusers** and then assign student2 to this group using: **sudo usermod -aG labusers student2**

```
admin@rellamariofrio:~$ sudo usermod -aG labusers student2
admin@rellamariofrio:~$ sudo mkdir /labdata
admin@rellamariofrio:~$ sudo chown root:labusers /labdata
admin@rellamariofrio:~$ sudo chmod 770 /labdata
admin@rellamariofrio:~$ _
```

## Step 3: Set Permissions

We create the /labdata directory with **sudo mkdir /labdata**, set its ownership to root:labusers using **sudo chown root:labusers /labdata**, and assign permissions 770 with **sudo chmod 770 /labdata**. This allows only root and members of labusers to access the folder.

# VERIFICATION & TROUBLESHOOTING

```
student2@rellamariofrio:~/labdata$ ls /labdata
student2@rellamariofrio:~/labdata$ touch testfile.txt
student2@rellamariofrio:~/labdata$ ls /labdata
testfile.txt
student2@rellamariofrio:~/labdata$
```

On Linux, we switch to the student2 account using: **su - student2**. From this account, the user should be able to access /labdata and create files inside, confirming that group permissions are working.

```
admin@rellamariofrio:~$ sudo chmod 750 /labdata
admin@rellamariofrio:~$ su - student2
Password:
student2@rellamariofrio:~$ cd /labdata
student2@rellamariofrio:/labdata$ touch newtextfile.txt
touch: cannot touch 'newtextfile.txt': Permission denied
student2@rellamariofrio:/labdata$ _
```

Next, we adjust the directory's permissions with: **sudo chmod 750 /labdata**. After this change, student2 should still be able to view the folder but no longer write new files inside it.

# EXERCISE

```
admin@rellamariofrio:~$ sudo adduser student4
info: Adding user 'student4' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group 'student4' (1004) ...
info: Adding new user 'student4' (1004) with group 'student4 (1004)' ...
info: Creating home directory '/home/student4' ...
info: Copying files from '/etc/skel' ...
New password:
Retype new password:
password: password updated successfully
Changing the user information for student4
Enter the new value, or press ENTER for the default
  Full Name []: Marc Elijah Riofrio
  Room Number []: 101
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
info: Adding new user 'student4' to supplemental / extra groups 'users' ...
info: Adding user 'student4' to group 'users' ...
admin@rellamariofrio:~$
```

```
admin@rellamariofrio:~$ sudo adduser faculty2
info: Adding user 'faculty2' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group 'faculty2' (1003) ...
info: Adding new user 'faculty2' (1003) with group 'faculty2 (1003)' ...
info: Creating home directory '/home/faculty2' ...
info: Copying files from '/etc/skel' ...
New password:
Retype new password:
password: password updated successfully
Changing the user information for faculty2
Enter the new value, or press ENTER for the default
  Full Name []: Faculty
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
info: Adding new user 'faculty2' to supplemental / extra groups 'users' ...
info: Adding user 'faculty2' to group 'users' ...
```

```
admin@rellamariofrio:~$ sudo usermod -aG facultygrp faculty2
admin@rellamariofrio:~$ sudo usermod -aG studentgrp student4
```

On the Ubuntu Server, we start by creating two new users, **faculty2** and **student4**, using the **adduser** command. We then create two groups: **facultygrp** and **studentgrp**, assigning **faculty2** to **facultygrp** and **student4** to **studentgrp**.

```
admin@rellamariofrio:~$ sudo mkdir /facultydata
admin@rellamariofrio:~$ sudo mkdir /studentdata
admin@rellamariofrio:~$ sudo chown :facultygrp /facultydata
admin@rellamariofrio:~$ sudo chown :studentgrp /studentdata
```

```
admin@rellamariofrio:~$ sudo chmod 770 /facultydata
admin@rellamariofrio:~$ sudo chmod 750 /studentdata
```

Next, we create two directories: **/facultydata** and **/studentdata**. For **/facultydata**, we assign ownership to **root:facultygrp** and set permissions to **770**, which grants full read/write/execute access to faculty members while restricting others. For **/studentdata**, we assign ownership to **root:studentgrp** and set permissions to **750**, allowing studentgrp members read-only access while preventing them from modifying files.

```
admin@rellamariofrio: ~$ sudo chmod 750 /studentdata
admin@rellamariofrio:~$ su - faculty2
Password:
faculty2@rellamariofrio:~$ cd /facultydata
faculty2@rellamariofrio:/facultydata$ touch testfile.txt
faculty2@rellamariofrio:/facultydata$ ls /facultydata
testfile.txt
faculty2@rellamariofrio:/facultydata$ su - student4
```

```
faculty2@rellamariofrio:/facultydata$ su - student4
Password:
student4@rellamariofrio:~$ cd /studentdata
student4@rellamariofrio:/studentdata$ ls
student4@rellamariofrio:/studentdata$ touch testfile.txt
touch: cannot touch 'testfile.txt': Permission denied
student4@rellamariofrio:/studentdata$
```

Finally, we test the configuration by switching to each user. When logged in as faculty2, the user should be able to access /facultydata and create or edit files, but only read files in /studentdata. Conversely, when logged in as student4, the user should be able to access /studentdata in read-only mode, but will be denied if trying to create or modify files. These tests confirm that permissions are correctly enforced.



Bicol University Polangui  
Polangui, Albay  
Computer Studies Department

IT-123

# System Administration and Maintenance

Laboratory Week4 - File Systems and Storage Management

Submitted by:  
Jeremy O. Rellama  
Marc Elijah N. Riofrio  
**BSIT 4A Student**

Submitted to:  
Guillermo V. Red, Jr. DIT  
**PROFFESOR**

**Date Submitted:** 09/11/2025



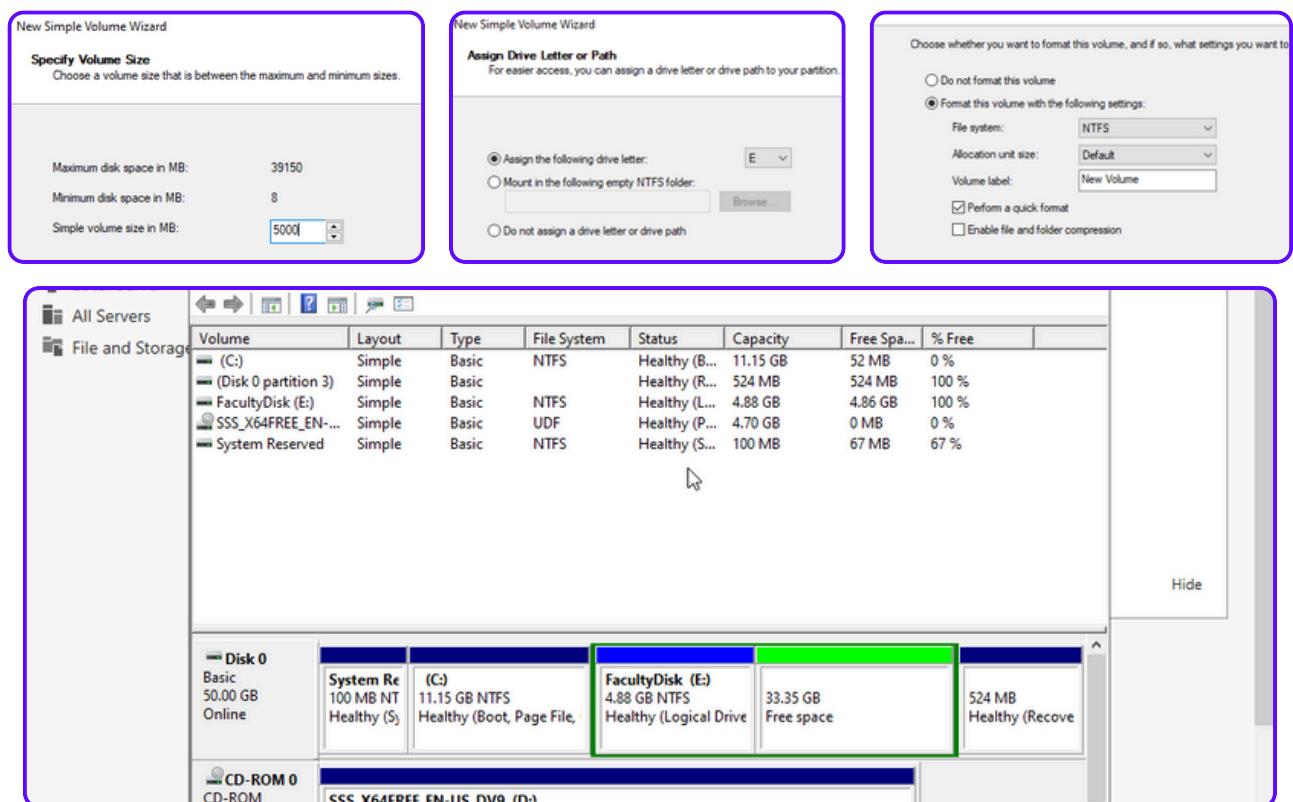
# Windows Server

# A. Partition Creation

Drive Letter: E:

File System: NTFS

Size: 5000 mb or 5gb

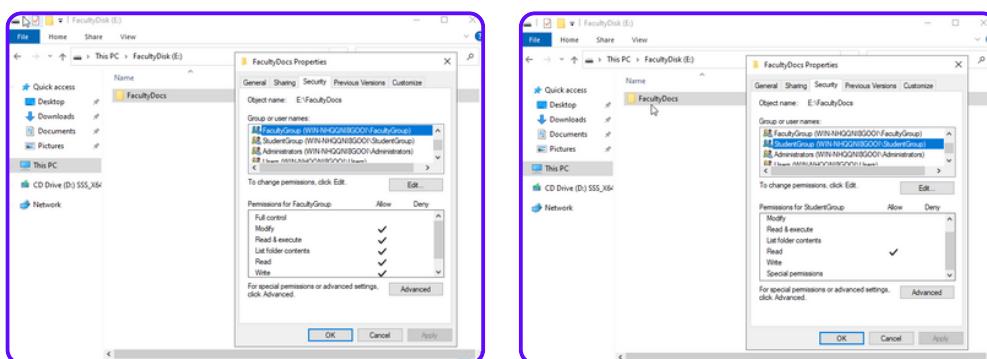


# B. Folder Setup & Permissions

Folder Name: E:\FacultyDocs

Group Assigned: FacultyGroup & StudentGroup

Permission Level: Modify only for faculty & Read only for student



**FacultyGroup**

**StudentGroup**

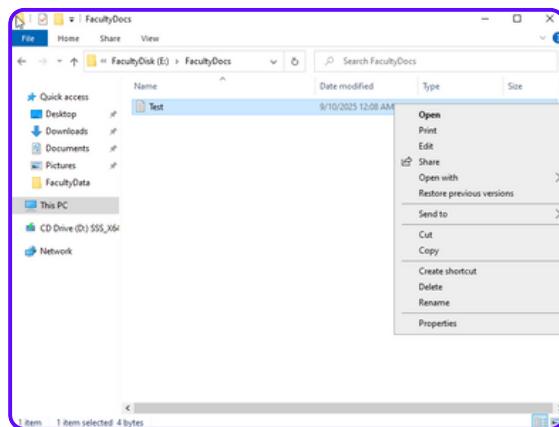
## C. Verification

Tested User Account: faculty1

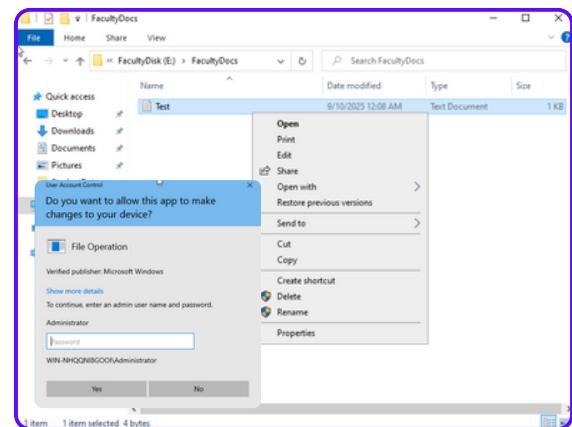
Access Allowed: [✓] Yes [ ] No

Tested User Account: student3

Access Denied: [ ] Yes [✓] No



FacultyGroup



StudentGroup

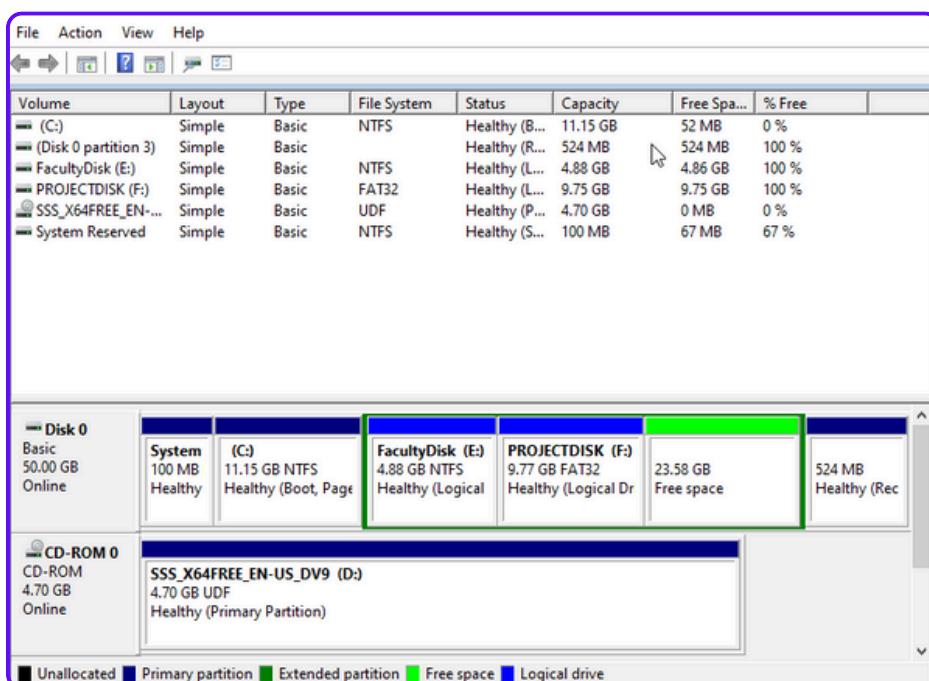
## Student Exercise

### A. Partition Creation

Drive Letter: F:

File System: FAT32

Size: 10000 mb or 10gb



## B. Folder Setup & Permissions

Folder Name: F:\SharedFaculty

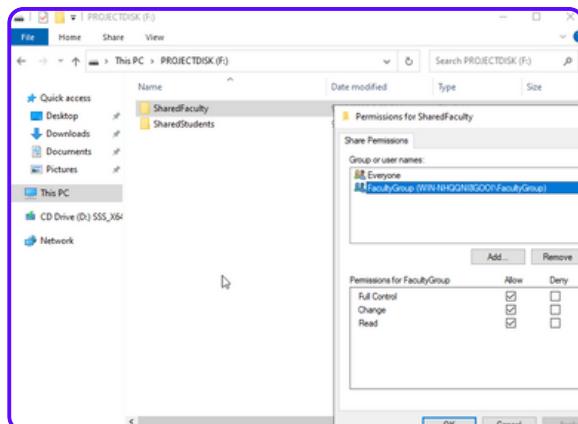
Group Assigned: FacultyGroup

Permission Level: Full Control

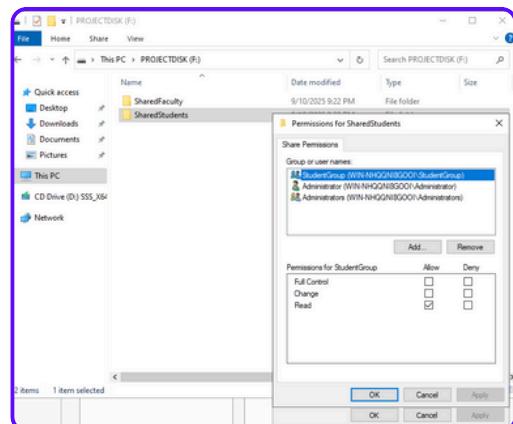
Folder Name: F:\SharedStudents

Group Assigned: StudentGroup

Permission Level: Read & Execute only



FacultyGroup



StudentGroup

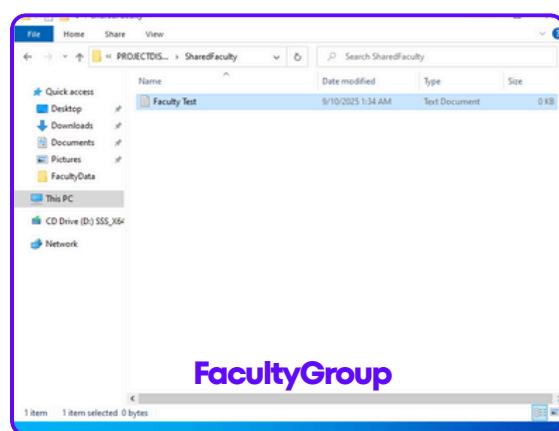
## C. Verification

Tested User Account: faculty1

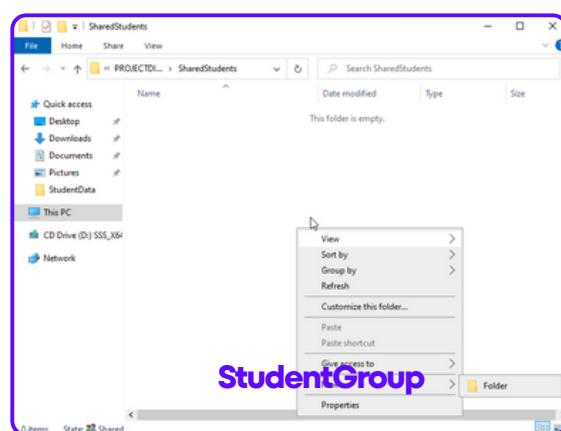
Access Allowed: [✓] Yes [ ] No

Tested User Account: student3

Access Denied: [ ] Yes [✓] No



FacultyGroup



StudentGroup



# **Ubuntu Server**

# A. Partition Creation

- Device Name: /dev/sdc1
- File System Type: ext4
- Mount Point: /projectdata

```
admin@rellamariofrio:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0   50G  0 disk
└─sda1   8:1    0     1M  0 part
└─sda2   8:2    0   50G  0 part /
sdb      8:16   0     5G  0 disk
sr0     11:0    1 1024M 0 rom
admin@rellamariofrio:~$
```

```
Command (m for help): n
Partition type
  p  primary (0 primary, 0 extended, 4 free)
  e  extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-10485759, default 10485759):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-10485759, default 10485759):
Created a new partition 1 of type 'Linux' and of size 5 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

```
admin@rellamariofrio:~$ sudo fdisk /dev/sdb
Welcome to fdisk (util-linux 2.39.3).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
```

```
Command (m for help):
```

```
admin@rellamariofrio:~$ sudo mkfs.ext4 /dev/sdb1
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 1310464 4k blocks and 327680 inodes
Filesystem UUID: 9a825db6-dc6f-4ec9-a6c3-b60f4f058354
Superblock backups stored on blocks:
          32768, 98304, 163840, 229376, 294912, 819200, 884736
Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

admin@rellamariofrio:~$ sudo mkdir /mnt/data
admin@rellamariofrio:~$ sudo mount /dev/sdb1 /mnt/data
```

```
admin@rellamariofrio:~$ sudo mkdir /mnt/data
admin@rellamariofrio:~$ sudo mount /dev/sdb1 /mnt/data
```

# B. Directory Setup & Permissions

- Directory Path: /mnt/data
- Group Assigned: facultygrp
- Permission Level: 770 (root + facultygrp full access, others none)

```
admin@rellamariofrio:~$ sudo chown root:facultygrp /mnt/data
admin@rellamariofrio:~$ sudo chmod 770 /mnt/data
admin@rellamariofrio:~$
```

# C. Verification

- Tested User Account: faculty2 and student4

```
admin@rellamariofrio:~$ su - faculty2
Password:
faculty2@rellamariofrio:~$ cd /projectdata
faculty2@rellamariofrio:/projectdata$ touch faculty_test.txt
faculty2@rellamariofrio:/projectdata$ ls
faculty_test.txt  lost+found
faculty2@rellamariofrio:/projectdata$ touch new_
faculty2@rellamariofrio:/projectdata$ ls
faculty_test.txt  lost+found  new_
faculty2@rellamariofrio:/projectdata$
```

Faculty User (**faculty2**) Can write inside /mnt/data

```
faculty2@rellamariofrio:/projectdata$ su - student4
Password:
student4@rellamariofrio:~$ cd /projectdata
student4@rellamariofrio:/projectdata$ ls
faculty_test.txt  lost+found  new_
student4@rellamariofrio:/projectdata$ cat faculty_test.txt
student4@rellamariofrio:/projectdata$ touch student_text.txt
touch: cannot touch 'student_text.txt': Permission denied
student4@rellamariofrio:/projectdata$
```

Student User (**student4**) denied write access

# Reflection

## **What challenges did you encounter in creating partitions and managing permissions?**

When we worked on Windows Server, one of the challenges we faced was making sure we selected the correct disk when creating a partition because choosing the wrong one could lead to data loss. We also found it tricky to assign the right permissions since it was easy to give either too much or too little access to a group. On Ubuntu Server, the challenge was using command-line tools like fdisk, mkfs.ext4, and lsblk, where a small mistake could affect the whole setup. We also needed to use ACLs along with chmod and chown to properly separate read/write access for faculty and read-only access for students.

## **How can these skills be applied in real-world system administration duties?**

We believe these skills are very important in real-world system administration because they help organize storage and protect files. On Windows Server, the same steps can be used to manage shared folders and set the right permissions for different groups. On Ubuntu Server, they are useful in handling storage and access control in enterprise environments where Linux is common. Overall, these activities showed us how system administrators secure data, manage resources, and keep systems stable.



Bicol University Polangui  
Polangui, Albay  
Computer Studies Department

IT-123

# System Administration and Maintenance

Laboratory Week6 - Network Configuration and Management

Submitted by:  
Jeremy O. Rellama  
Marc Elijah N. Riofrio  
**BSIT 4A Student**

Submitted to:  
Guillermo V. Red, Jr. DIT  
**PROFFESOR**



# Ubuntu Server

# Part 1. Checking Current Network Config

## Commands:

- `ip a`
- `ip route`
- `cat /etc/resolv.conf`

```
admin@rellamariofrio:~$ cat /etc/resolv.conf
This is /run/systemd/resolve/stub-resolv.conf managed by man:systemd-resolved(8)
Do not edit.

This file might be symlinked as /etc/resolv.conf. If you're looking at
/etc/resolv.conf and seeing this text, you have followed the symlink.

This is a dynamic resolv.conf file for connecting local clients to the
internal DNS stub resolver of systemd-resolved. This file lists all
configured search domains.

Run "resolvectl status" to see details about the uplink DNS servers
currently in use.

Third party programs should typically not access this file directly, but only
through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
different way, replace this symlink by a static file or a different symlink.

See man:systemd-resolved.service(8) for details about the supported modes of
operation for /etc/resolv.conf.

nameserver 127.0.0.53
options edns0 trust-ad
search .
admin@rellamariofrio:~$
```

```
admin@rellamariofrio:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:da:30:7c brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86247sec preferred_lft 86247sec
    inet6 fd17:625c:f037:2:a00:27ff:fed:307c/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 86250sec preferred_lft 14250sec
    inet6 fe80::a00:27ff:fed:307c/64 scope link
        valid_lft forever preferred_lft forever
admin@rellamariofrio:~$ ip route
default via 10.0.2.15 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
10.0.2.2 dev enp0s3 proto dhcp scope link src 10.0.2.15 metric 100
10.0.2.3 dev enp0s3 proto dhcp scope link src 10.0.2.15 metric 100
admin@rellamariofrio:~$
```

# Part 2 – Configuring a Static IP Address

## Commands:

- `sudo nano /etc/netplan/00-installer-config.yaml`

## yaml configuration:

```
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.1.100/24
      routes:
        - to: default
          via: 192.168.1.1
    nameservers:
      addresses:
        - 8.8.8.8
        - 8.8.4.4
```

**Then:** `sudo netplan apply`

```
admin@rellamariofrio:~$ sudo netplan apply
admin@rellamariofrio:~$
```

```
admin@rellamariofrio:~$ sudo netplan apply
admin@rellamariofrio:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:da:30:7c brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.100/24 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft forever preferred_lft forever
    inet 10.0.2.15/24 metric 100 brd ff:ff:ff:ff:ff:ff
        valid_lft 86366sec preferred_lft 86366sec
    inet6 fd17:625c:f037:2:a00:27ff:fed:307c/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 86369sec preferred_lft 14368sec
    inet6 fe80::a00:27ff:fed:307c/64 scope link
        valid_lft forever preferred_lft forever
admin@rellamariofrio:~$
```

```
admin@rellamariofrio:~$ ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=118 time=250 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=118 time=18.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=118 time=18.0 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=118 time=18.6 ms
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3989ms
rtt min/avg/max/mdev = 18.025/76.321/249.828/100.174 ms
```

`ping -c 4 8.8.8.8`

```
admin@rellamariofrio:~$ ping -c 4 google.com
PING google.com (2404:6800:4017:801::200e) 56 data bytes
64 bytes from mn107s02-in-xoe.1e100.net (2404:6800:4017:801::200e): icmp_seq=1 ttl=118 time=21.862 ms
64 bytes from mn107s02-in-xoe.1e100.net (2404:6800:4017:801::200e): icmp_seq=2 ttl=118 time=21.862 ms
64 bytes from mn107s02-in-xoe.1e100.net (2404:6800:4017:801::200e): icmp_seq=3 ttl=118 time=21.862 ms
64 bytes from mn107s02-in-xoe.1e100.net (2404:6800:4017:801::200e): icmp_seq=4 ttl=118 time=21.862 ms
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3011ms
rtt min/avg/max/mdev = 21.862/232.225/495.546/170.313 ms
```

`ping -c 4 google.com`

## Part 3 – Switching Back to DHCP

### Commands:

- sudo nano /etc/netplan/00-installer-config.yaml

### yaml configuration:

```
GNU nano 7.2
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: yes
```

```
admin@rellamariofrio:~$ sudo netplan apply
admin@rellamariofrio:~$ ip a show enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    link/ether 08:00:27:da:30:7c brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.7/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
            valid_lft 86399sec preferred_lft 86399sec
        inet6 2001:4453:7f3:5900:a00:27ff:feda:307c/64 scope global dynamic mngtmpaddr noprefixroute
            valid_lft 259198sec preferred_lft 172798sec
        inet6 fe80::a00:27ff:fed:a307c/64 scope link
            valid_lft forever preferred_lft forever
admin@rellamariofrio:~$ _
```

### Then:

- sudo netplan apply
- ip a / ip a show enp0s3

## Part 4 – Troubleshooting with CLI Tools

### 1. Ping(Gateway test)

- ping -c 4 192.168.1.1

```
admin@rellamariofrio:~$ ping -c 4 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=11.3 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=6.83 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=4.35 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=8.72 ms

--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 4324ms
rtt min/avg/max/mdev = 4.345/7.789/11.265/2.536 ms
```

### 2. Traceroute (Internet path)

- traceroute google.com

```
admin@rellamariofrio:~$ traceroute google.com
traceroute to google.com (142.251.220.174), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.1)  7.803 ms  3.455 ms  2.853 ms
 2  100.67.0.1 (100.67.0.1)  11.079 ms  10.727 ms  9.029 ms
 3  122.2.181.173.static.pldt.net (122.2.181.173)  10.382 ms  10.238 ms  9.889 ms
 4  210.213.134.20.static.pldt.net (210.213.134.20)  11.311 ms  210.213.134.40.static.pldt.net (210.213.134.40)  10.943 ms
 5  * *
 6  * *
 7  216.239.56.40 (216.239.56.40)  1081.057 ms  142.250.58.242 (142.250.58.242)  25.269 ms mn107s02-in-f14.1e100.net (142.250.58.242)
```

### 3. DNS Resolution (nslookup)

- nslookup google.com

```
admin@rellamariofrio:~$ nslookup google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.251.220.174
Name:   google.com
Address: 2404:6800:4017:802::200e
```

## 4. Active Connections (ss)

- `ss -tuln`

```
3 122.2.101.173.static.pldt.net (122.2.101.173) 10.302 ms 10.238 ms 3.003 ms
4 210.213.134.20.static.pldt.net (210.213.134.20) 11.311 ms 210.213.134.40.static.pldt.net (210.213.134.40) 10.943
5 * * *
6 * * *
7 216.239.56.40 (216.239.56.40) 1081.057 ms 142.250.58.242 (142.250.58.242) 25.269 ms mn107s02-in-f14.1e100.net (14
admin@rellamariofrio:~$ nslookup google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.251.220.174
Name:   google.com
Address: 2404:6800:4017:802::200e

admin@rellamariofrio:~$ ss -tuln
Netid      State      Recv-Q      Send-Q      Local Address:Port
udp        UNCONN      0          0          127.0.0.54:53
udp        UNCONN      0          0          127.0.0.53%lo:53
udp        UNCONN      0          0          192.168.1.7%enp0s3:68
udp        UNCONN      0          0          [fe80::a00:27ff:fed:a307c]:546
tcp        LISTEN      0          4096      127.0.0.54:53
tcp        LISTEN      0          4096      127.0.0.53%lo:53
```

## Part 6 – Student Exercise

- Server A → Static IP: 192.168.1.150/24
- Server B → DHCP (Host machine at 192.168.1.3)

```
Connection-specific DNS Suffix  . : 
IPv6 Address . . . . . : [REDACTED]:45c1
Temporary IPv6 Address . . . . . : [REDACTED]:7c60-73d9-1122-2870
Link-local IPv6 Address . . . . . : [REDACTED]6523:471c%13
IPv4 Address . . . . . : 192.168.1.3
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : fe80::1%13
                           192.168.1.1
```

### Ping the host PC(Server B) from the Ubuntu Serve(VM)

```
admin@rellamariofrio:~$ ping -c 4 192.168.1.3
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.
64 bytes from 192.168.1.3: icmp_seq=1 ttl=128 time=0.703 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=128 time=0.566 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=128 time=0.563 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=128 time=0.515 ms

--- 192.168.1.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3027ms
rtt min/avg/max/mdev = 0.515/0.586/0.703/0.070 ms
```

## Ping Server A (Ubuntu Server from host pC)

```
PS C:\Users\Mark> ping 192.168.1.150  
  
Pinging 192.168.1.150 with 32 bytes of data:  
Reply from 192.168.1.150: bytes=32 time=1ms TTL=64  
  
Ping statistics for 192.168.1.150:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

## Part 6.1 – 8.8.8.8 and google.com ping tests

### From Ubuntu Server

```
admin@rellamariofrio:~$ ping -c 4 8.8.8.8  
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=118 time=122 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=118 time=410 ms  
64 bytes from 8.8.8.8: icmp_seq=3 ttl=118 time=18.2 ms  
64 bytes from 8.8.8.8: icmp_seq=4 ttl=118 time=18.4 ms  
  
--- 8.8.8.8 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3012ms  
rtt min/avg/max/mdev = 18.237/142.074/409.747/160.221 ms
```

```
admin@rellamariofrio:~$ ping -c 4 8.8.8.8  
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=118 time=122 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=118 time=410 ms  
64 bytes from 8.8.8.8: icmp_seq=3 ttl=118 time=18.2 ms  
64 bytes from 8.8.8.8: icmp_seq=4 ttl=118 time=18.4 ms  
  
--- 8.8.8.8 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3012ms  
rtt min/avg/max/mdev = 18.237/142.074/409.747/160.221 ms
```

### From Windows 11(HOST)

```
PS C:\Users\Mark> ping 8.8.8.8  
  
Pinging 8.8.8.8 with 32 bytes of data:  
Reply from 8.8.8.8: bytes=32 time=336ms TTL=118  
Reply from 8.8.8.8: bytes=32 time=17ms TTL=118  
Reply from 8.8.8.8: bytes=32 time=17ms TTL=118  
Reply from 8.8.8.8: bytes=32 time=16ms TTL=118  
  
Ping statistics for 8.8.8.8:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 16ms, Maximum = 336ms, Average = 96ms  
PS C:\Users\Mark> ping google.com
```

```
PS C:\Users\Mark> ping 8.8.8.8  
  
Pinging 8.8.8.8 with 32 bytes of data:  
Reply from 8.8.8.8: bytes=32 time=336ms TTL=118  
Reply from 8.8.8.8: bytes=32 time=17ms TTL=118  
Reply from 8.8.8.8: bytes=32 time=17ms TTL=118  
Reply from 8.8.8.8: bytes=32 time=16ms TTL=118  
  
Ping statistics for 8.8.8.8:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 16ms, Maximum = 336ms, Average = 96ms  
PS C:\Users\Mark> ping google.com
```



Bicol University Polangui  
Polangui, Albay  
Computer Studies Department

IT-123

# System Administration and Maintenance

Laboratory Week7 - Configuring and Securing Network Services

Submitted by:  
Jeremy O. Rellama  
Marc Elijah N. Riofrio  
**BSIT 4A Student**

Submitted to:  
Guillermo V. Red, Jr. DIT  
**PROFFESOR**

# Part 1. DNS Configuration

## Commands:

- sudo apt update
- sudo apt install bind9 bind9utils -y
- sudo nano /etc/bind/named.conf.local

```
et:39 http://ph.archive.ubuntu.com/ubuntu noble-backports/main amd64
et:40 http://ph.archive.ubuntu.com/ubuntu noble-backports/main arm64
et:41 http://ph.archive.ubuntu.com/ubuntu noble-backports/restricted amd64
et:42 http://ph.archive.ubuntu.com/ubuntu noble-backports/universe amd64
et:43 http://ph.archive.ubuntu.com/ubuntu noble-backports/universe arm64
et:44 http://ph.archive.ubuntu.com/ubuntu noble-backports/universe i386
et:45 http://ph.archive.ubuntu.com/ubuntu noble-backports/universe all
et:46 http://ph.archive.ubuntu.com/ubuntu noble-backports/multiverse i386
etched 12.3 MB in 3min 6s (66.1 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them
admin2@ubuntuserver:~$
```

```
zone company.local {
    type master;
    file "/etc/bind/db.company.local";
};
```

- sudo cp /etc/bind/db.local /etc/bind/db.company.local
- sudo nano /etc/bind/db.company.local
- sudo systemctl restart bind9
- nslookup www.company.local 127.0.0.1

```
86400      ; Retry
2419200    ; Expire
604800 )   ; Negative Cache TTL
            ;

;           IN      NS      ns1.example.local.
ns1        IN      A       192.168.1.150
www        IN      A       192.168.1.150
```

```
admin2@ubuntuserver:~$ nslookup www.example.local 127.0.0.1
Server:      127.0.0.1
Address:     127.0.0.1#53

Name:        www.example.local
Address:    192.168.1.150
```

# Part 2. DHCP Configuration

## Commands:

- sudo apt install isc-dhcp-server -y
- sudo nano /etc/default/isc-dhcp-server

Change to enp0s8

```
admin2@ubuntuserver:~$ sudo apt install isc-dhcp-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  isc-dhcp-common
Suggested packages:
  isc-dhcp-server-ldap policycoreutils
The following NEW packages will be installed:
  isc-dhcp-server
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
```

```
[INTERFACESv4="enp0s8"
[INTERFACESv6=""]
```

- sudo nano /etc/dhcp/dhcpd.conf
- sudo systemctl restart isc-dhcp-server
- sudo systemctl status isc-dhcp-server

```
ubnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.160 192.168.1.170;
    option routers 192.168.1.1;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

DHCP Running

```
admin2@ubuntuserver:~$ sudo systemctl restart isc-dhcp-server
admin2@ubuntuserver:~$ sudo systemctl status isc-dhcp-server
  isc-dhcp-server.service - ISC DHCP IPv4 server
    Loaded: loaded (/usr/lib/systemd/system/isc-dhcp-server.service)
    Active: active (running) since Wed 2025-10-08 13:44:12 UTC; 1s ago
      Docs: man:isc-dhcp-server(8)
```

## Part 3. HTTP Service (Apache)

### Commands:

- sudo apt install apache2 -y
- sudo systemctl restart apache2
- curl http://192.168.1.150

```
dmin2@ubuntuserver:~$  
dmin2@ubuntuserver:~$ sudo apt install apache2 -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
apache2 is already the newest version (2.4.58-1ubuntu8.8).
```

```
dmin2@ubuntuserver:~$ sudo systemctl status apache2  
apache2.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/apache2.service;  
             Active: active (running) since Wed 2025-10-08 12:26:22 U  
   Main PID: 1695 (apache2)  
     Tasks: 55 (limit: 5286)  
    Memory: 5.4M (peak: 5.7M)  
      CPU: 86ms  
     CpuTime: 1.000us
```

```
<div class="section_header">  
  <div id="docroot"></div>  
  Document Roots  
</div>  
  
<div class="content_section_text">  
  <p>By default, Ubuntu does not allow access through the web browser to<br/>any files outside of those located in <a href="/var/www/html">/var/www/html</a>.<br/>This is to prevent users from accidentally exposing sensitive or private<br/>directories (when enabled) and <a href="https://www.iana.org/assignments/txt-parameters/txt-parameters.xhtml#public_html" rel="nofollow">public_html</a> (for web<br/>applications). If your site is using a web document root<br/>located elsewhere (such as in <a href="/vhosts">/vhosts</a>) you may need to whitelist your<br/>document root directory in <a href="https://www.iana.org/assignments/txt-parameters/txt-parameters.xhtml#public_html" rel="nofollow">/etc/apache2/sites-available</a>.</p>  
  <p>The default Ubuntu document root is <a href="/var/www/html">/var/www/html</a>. You<br/>can make your own virtual hosts under <a href="/vhosts">/vhosts</a>.  
</p>
```

```
<div class="section_header">  
  <div id="bugs"></div>  
  Reporting Problems  
</div>  
  
<div class="content_section_text">  
  <p>Please use the <a href="https://ubuntu-bug.com/">ubuntu-bug</a> tool to report bugs in the<br/>Apache2 package with Ubuntu. However, check <a href="https://bugs.launchpad.net/ubuntu/+source/apache2" rel="nofollow">existing bug reports</a> before reporting a new bug.</p>  
  <p>Please report bugs specific to modules (such as PHP and others)<br/>to their respective packages, not to the web server itself.</p>  
</div>
```

## Part 4. FTP Service

### Commands:

- sudo apt install vsftpd -y
- sudo nano /etc/vsftpd.conf

```
dmin2@ubuntuserver:~$ sudo apt install vsftpd -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done
```

Uncomment this to enable any form of FTP write command.  
write\_enable=YES

Default umask for local users is 077. You may wish to change  
it if your users expect that 022 is used by most other ftppd's

- sudo systemctl restart vsftpd
- ftp 192.168.1.150

```
admin2@ubuntuserver:~$ sudo systemctl restart vsftpd  
admin2@ubuntuserver:~$ ftp 192.168.1.150  
Connected to 192.168.1.150.  
220 (vsFTPd 3.0.5)  
Name (192.168.1.150:admin2):
```

# Part 5. Securing Services (Firewall)

## Commands:

- sudo ufw allow 22/tcp
- sudo ufw allow 53/tcp
- sudo ufw allow 80/tcp
- sudo ufw allow 21/tcp
- sudo ufw enable
- sudo ufw status

```
# Uncomment this to enable any +  
write_enable=YES  
listen_address=192.168.1.150
```

```
admin2@ubuntuserver:~$ sudo ufw status  
Status: active  
  
To                         Action      From  
---                        ----  
22/tcp                      ALLOW       Anywhere  
53/tcp                      ALLOW       Anywhere  
80/tcp                      ALLOW       Anywhere  
21/tcp                      ALLOW       Anywhere  
22/tcp (v6)                 ALLOW       Anywhere (v6)  
53/tcp (v6)                 ALLOW       Anywhere (v6)  
80/tcp (v6)                 ALLOW       Anywhere (v6)  
21/tcp (v6)                 ALLOW       Anywhere (v6)
```

```
admin2@ubuntuserver:~$ sudo ufw allow 22/tcp  
Rules updated  
Rules updated (v6)  
admin2@ubuntuserver:~$ sudo ufw allow 53/tcp  
Rules updated  
Rules updated (v6)  
admin2@ubuntuserver:~$ sudo ufw allow 80/tcp  
Rules updated  
Rules updated (v6)  
admin2@ubuntuserver:~$ sudo ufw allow 21/tcp  
Rules updated  
Rules updated (v6)  
admin2@ubuntuserver:~$ sudo ufw enable  
Firewall is active and enabled on system startup  
admin2@ubuntuserver:~$ _
```

# Part 6 – Student Exercise (Ubuntu CLI)

## Commands:

- sudo apt update
- sudo apt install bind9 bind9utils -y
- sudo nano /etc/bind/named.conf.local
- sudo cp /etc/bind/db.local /etc/bind/db.company.local
- sudo nano /etc/bind/db.company.local
- sudo systemctl restart bind9
- nslookup www.company.local 127.0.0.1

```
zone "company.local" {  
    type master;  
    file "/etc/bind/db.company.local";  
};
```

```
admin2@ubuntuserver:~$ sudo systemctl restart bind9  
admin2@ubuntuserver:~$ nslookup www.company.local 127.0.0.1  
Server: 127.0.0.1  
Address: 127.0.0.1#53  
  
Name: www.company.local  
Address: 192.168.1.150
```

```
; BIND data file for local loopback interface  
;  
$TTL    604800  
@      IN      SOA     ns1.company.local. admin.company.local (  
                          2           ; Serial  
                          604800    ; Refresh  
                          86400     ; Retry  
                          2419200   ; Expire  
                          604800 )  ; Negative Cache TTL  
;  
@      IN      NS      ns1.company.local  
ns1   IN      A       192.168.1.150  
www  IN      A       192.168.1.150
```

## Commands:

- sudo nano /etc/dhcp/dhcpd.conf
- sudo systemctl restart isc-dhcp-server
- sudo systemctl status isc-dhcp-server

```
#subnet 192.168.1.0 netmask 255.255.255.0 {  
    range 192.168.1.160 192.168.1.170;  
    option routers 192.168.1.1;  
    option domain-name-servers 192.168.1.150, 8.8.8.8;  
    option domain-name "company.local";  
}
```

```
dmin2@ubuntuserver:~$ sudo systemctl restart isc-dhcp-server  
dmin2@ubuntuserver:~$ sudo systemctl status isc-dhcp-server  
isc-dhcp-server.service - ISC DHCP IPv4 server  
  Loaded: loaded (/usr/lib/systemd/system/isc-dhcp-server.  
  Active: active (running) since Wed 2025-10-08 13:44:12 U  
    Docs: man:dhcpd(8)  
   Main PID: 1655 (dhcnd)
```

## Commands:

- sudo systemctl restart apache2
- curl http://192.168.1.150

```
admin2@ubuntuserver:~$ sudo apt install apache2  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
apache2 is already the newest version (2.4.58-1  
+upgraded, 0 newly installed, 0 to remove and  
admin2@ubuntuserver:~$ sudo systemctl restart a  
admin2@ubuntuserver:~$ curl http://192.168.1.15
```

```
</div>  
<div class="validator">  
<h1> Welcome to Company Intranet <  
<p>Server IP: 192.168.1.150 </p>  
</div>  
</body>
```

## Commands:

- sudo nano /etc/vsftpd.conf
- sudo systemctl restart vsftpd
- ftp 192.168.1.150

```
#  
# Uncomment this to enable any file write  
write_enable=YES  
listen_address=192.168.1.150  
#
```

```
admin2@ubuntuserver:~$ sudo systemctl r  
admin2@ubuntuserver:~$ ftp 192.168.1.15  
Connected to 192.168.1.150.  
220 (vsFTPd 3.0.5)  
Name (192.168.1.150:admin2):
```

## Commands:

- sudo ufw allow 22/tcp
- sudo ufw allow 53/tcp
- sudo ufw allow 80/tcp
- sudo ufw allow 21/tcp
- sudo ufw enable
- sudo ufw status

```
admin2@ubuntuserver:~$ sudo ufw status  
Status: active  
  
 To          Action    From  
 --          ----  
 22/tcp      ALLOW     Anywhere  
 53/tcp      ALLOW     Anywhere  
 80/tcp      ALLOW     Anywhere  
 21/tcp      ALLOW     Anywhere  
 22/tcp (v6)  ALLOW     Anywhere (v6)  
 53/tcp (v6)  ALLOW     Anywhere (v6)  
 80/tcp (v6)  ALLOW     Anywhere (v6)  
 21/tcp (v6)  ALLOW     Anywhere (v6)
```

```
admin2@ubuntuserver:~$ sudo ufw allow 22/tcp  
Rules updated  
Rules updated (v6)  
admin2@ubuntuserver:~$ sudo ufw allow 53/tcp  
Rules updated  
Rules updated (v6)  
admin2@ubuntuserver:~$ sudo ufw allow 80/tcp  
Rules updated  
Rules updated (v6)  
admin2@ubuntuserver:~$ sudo ufw allow 21/tcp  
Rules updated  
Rules updated (v6)  
admin2@ubuntuserver:~$ sudo ufw enable  
Firewall is active and enabled on system startup  
admin2@ubuntuserver:~$ _
```



Bicol University Polangui  
Polangui, Albay  
Computer Studies Department

IT-123

# System Administration and Maintenance

Laboratory Week10 - Firewall and VPN Configuration

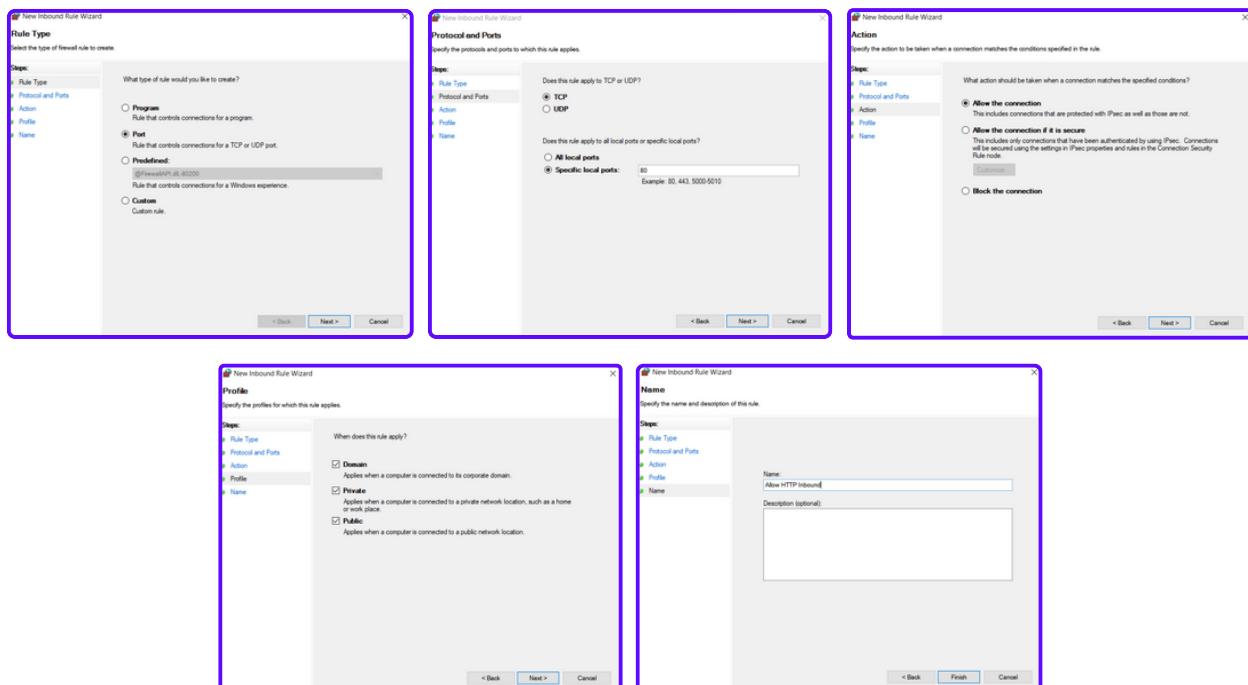
Submitted by:  
Jeremy O. Rellama  
Marc Elijah N. Riofrio  
**BSIT 4A Student**

Submitted to:  
Guillermo V. Red, Jr. DIT  
**PROFFESOR**

# TASK 1: CONFIGURING WINDOWS FIREWALL RULES

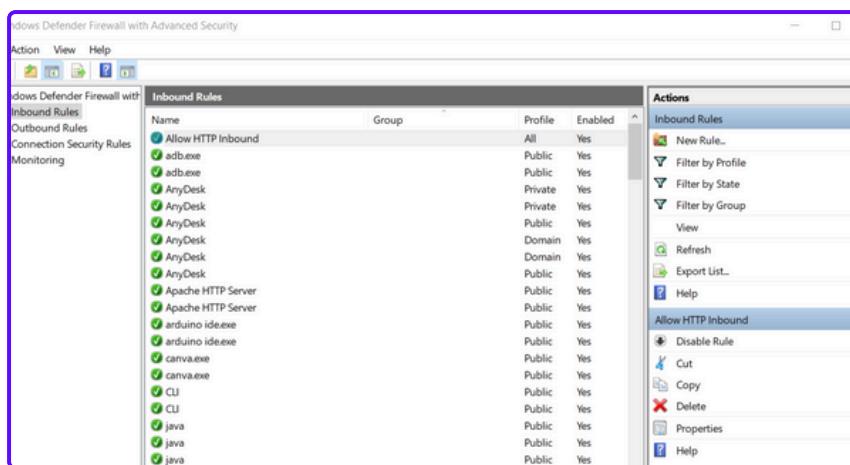
## 1. Configure Windows Firewall Rules (Inbound & Outbound)

- In the left pane select **Inbound Rules**.
- Click **New Rule...** in the right pane. Choose **Port** → **Next**.
- Select **TCP** (or **UDP** as needed) and enter the Port number (e.g., 80). Click **Next**.
- Choose **Allow the connection** → **Next**. Select profiles (**Domain/Private/Public**) appropriate to your test environment → **Next**.
- Give the rule a name, e.g., **Allow HTTP**, and click **Finish**.



## 2. Create an Outbound Port Rule

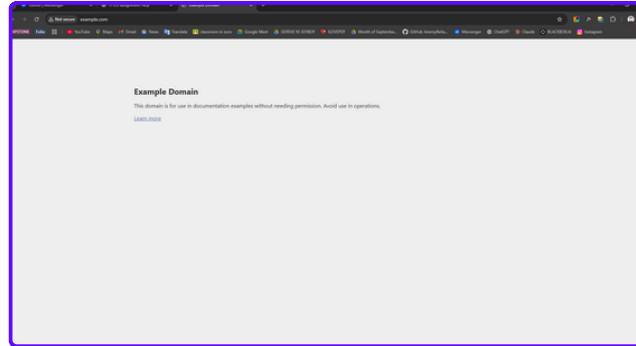
Repeat the inbound rule steps but start with **Outbound Rules** → **New Rule...**; configure the same port/protocol and name it (e.g., **Allow Outbound HTTP**).



### 3. Test Service Accessibility (Rule Enabled)

- If you opened port 80, run a simple web service (or point a client to a known web server) and use a browser on another machine/VM to connect to the server's IP on that port.
- Confirm the connection succeeds while the **firewall rule is enabled**.

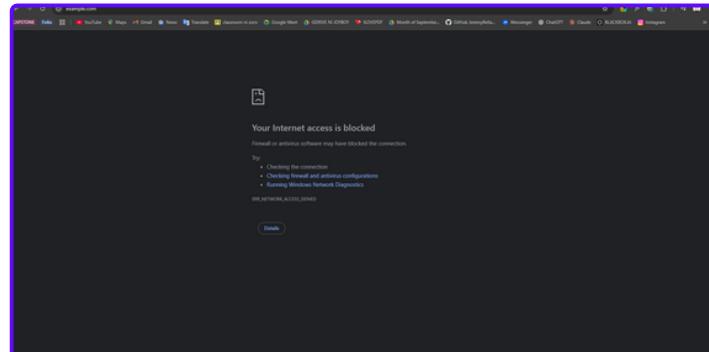
The screenshot shows two windows of the Windows Defender Firewall with Advanced Security interface. The left window displays the 'Outbound Rules' section, which includes a list of rules such as 'Allow HTTP Outbound', 'Allow Firewall API DLL', and several 'Allown Router' entries. The right window displays the 'Inbound Rules' section, also listing similar rules like 'Allow HTTP Inbound', 'Allow Firewall API DLL', and 'Allown Router'. Both sections have an 'Actions' pane on the right with options like 'New Rule...', 'Filter by Profile', 'Filter by State', and 'Filter by Group'.



### 4. Disable Rule & Confirm Blocking

- In the firewall console, **disable your inbound rule** (right-click → Disable Rule).
- Retry the same connection — it should fail.

The screenshot shows the Windows Defender Firewall with Advanced Security interface. The 'Inbound Rules' section is displayed, showing a list of rules. The 'Allow HTTP Inbound' rule, which was previously enabled, now has its 'Enabled' column set to 'No'. The 'Actions' pane on the right side of the interface is visible, showing options like 'New Rule...', 'Filter by Profile', 'Filter by State', and 'Filter by Group'.



# TASK 2: SETTING UP A VPN IN WINDOWS

## 1. Set Up a VPN (Windows)

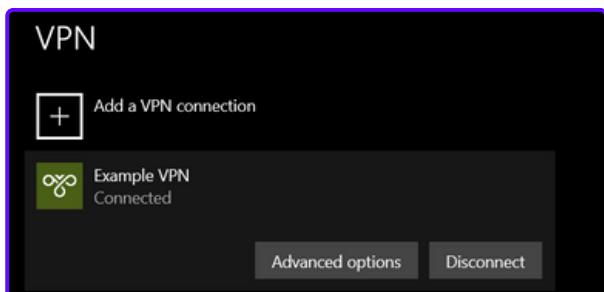
Open VPN Settings then, **Create the VPN Profile**  
In the Add VPN dialog:

- VPN provider: Windows (built-in)
- Connection name: Example VPN
- Server name or address: enter VPN server IP/domain
- VPN type: L2TP/IPsec with pre-shared key
- Type of sign-in info: Username and password
- Enter Username and Password fields (or leave blank to prompt at connect) → Save.



## 1. Testing

a. Connect to VPN



b. Verify VPN Tunnel using **WhatIsMyIP.com**