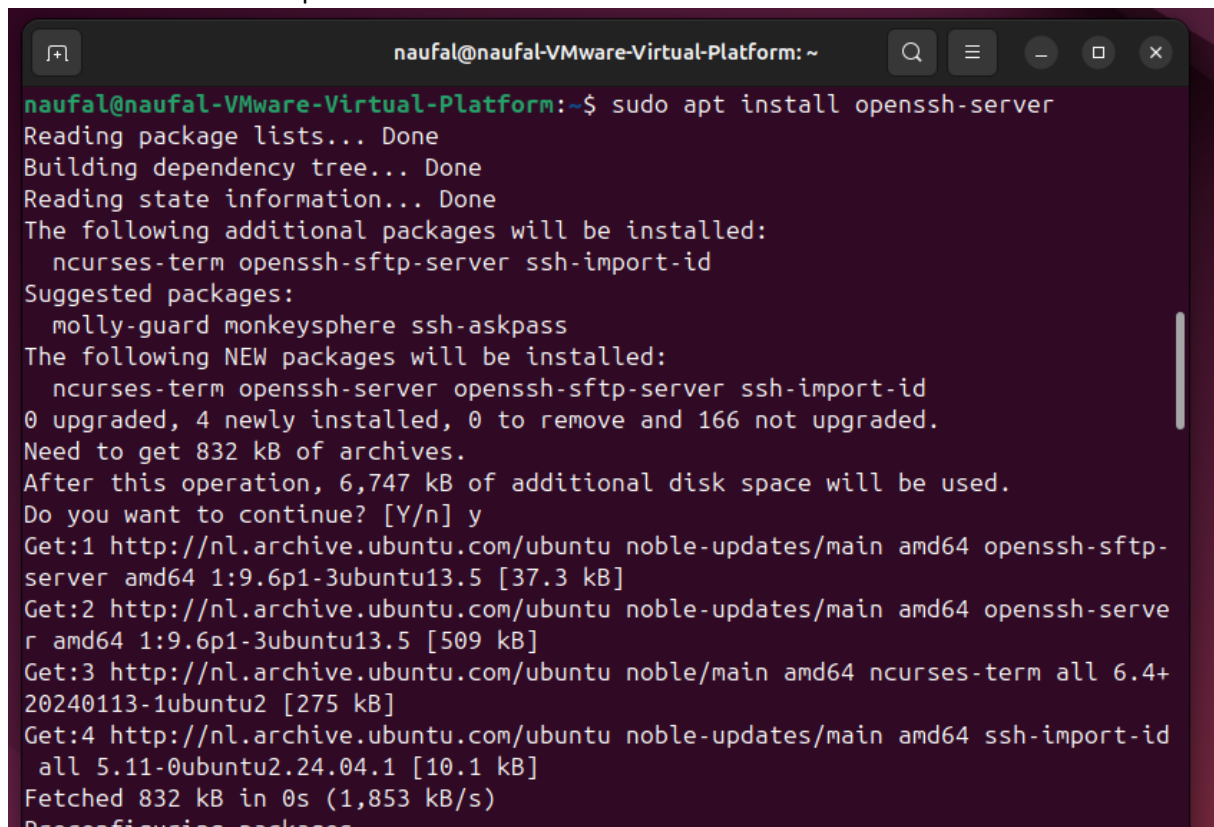


# Template Week 6 – Networking

Student number: 569508

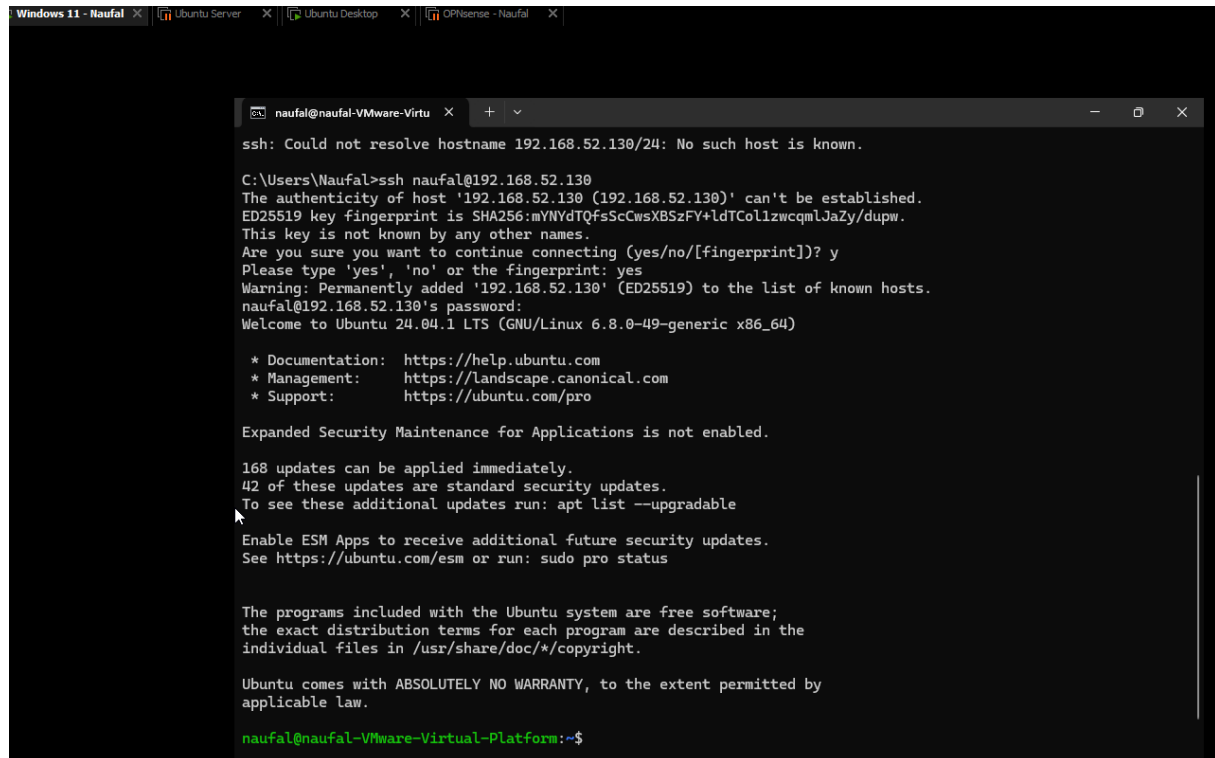
## Assignment 6.1: Working from home

Screenshot installation openssh-server:

A terminal window titled 'naufal@naufal-VMware-Virtual-Platform: ~' showing the command 'sudo apt install openssh-server' and its output. The output lists additional packages to be installed, suggested packages, and the disk space requirements. The user confirms the installation with 'y'. The terminal shows the progress of fetching packages from the Ubuntu repository.

```
naufal@naufal-VMware-Virtual-Platform:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 166 not upgraded.
Need to get 832 kB of archives.
After this operation, 6,747 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://nl.archive.ubuntu.com/ubuntu noble-updates/main amd64 openssh-sftp-
server amd64 1:9.6p1-3ubuntu13.5 [37.3 kB]
Get:2 http://nl.archive.ubuntu.com/ubuntu noble-updates/main amd64 openssh-serve
r amd64 1:9.6p1-3ubuntu13.5 [509 kB]
Get:3 http://nl.archive.ubuntu.com/ubuntu noble/main amd64 ncurses-term all 6.4+
20240113-1ubuntu2 [275 kB]
Get:4 http://nl.archive.ubuntu.com/ubuntu noble-updates/main amd64 ssh-import-id
all 5.11-0ubuntu2.24.04.1 [10.1 kB]
Fetched 832 kB in 0s (1,853 kB/s)
Preconfiguring packages...
```

## Screenshot successful SSH command execution:



```
Windows 11 - Naufal X Ubuntu Server X Ubuntu Desktop X OPNsense - Naufal X
naufal@naufal-VMware-Virtu X + - O X

ssh: Could not resolve hostname 192.168.52.130/24: No such host is known.

C:\Users\Naufal>ssh naufal@192.168.52.130
The authenticity of host '192.168.52.130 (192.168.52.130)' can't be established.
ED25519 key fingerprint is SHA256:mYNYdTQfsScCwsXBSzFY+LdTCollzwcqmlJaZy/dupw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.52.130' (ED25519) to the list of known hosts.
naufal@192.168.52.130's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-49-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

168 updates can be applied immediately.
42 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

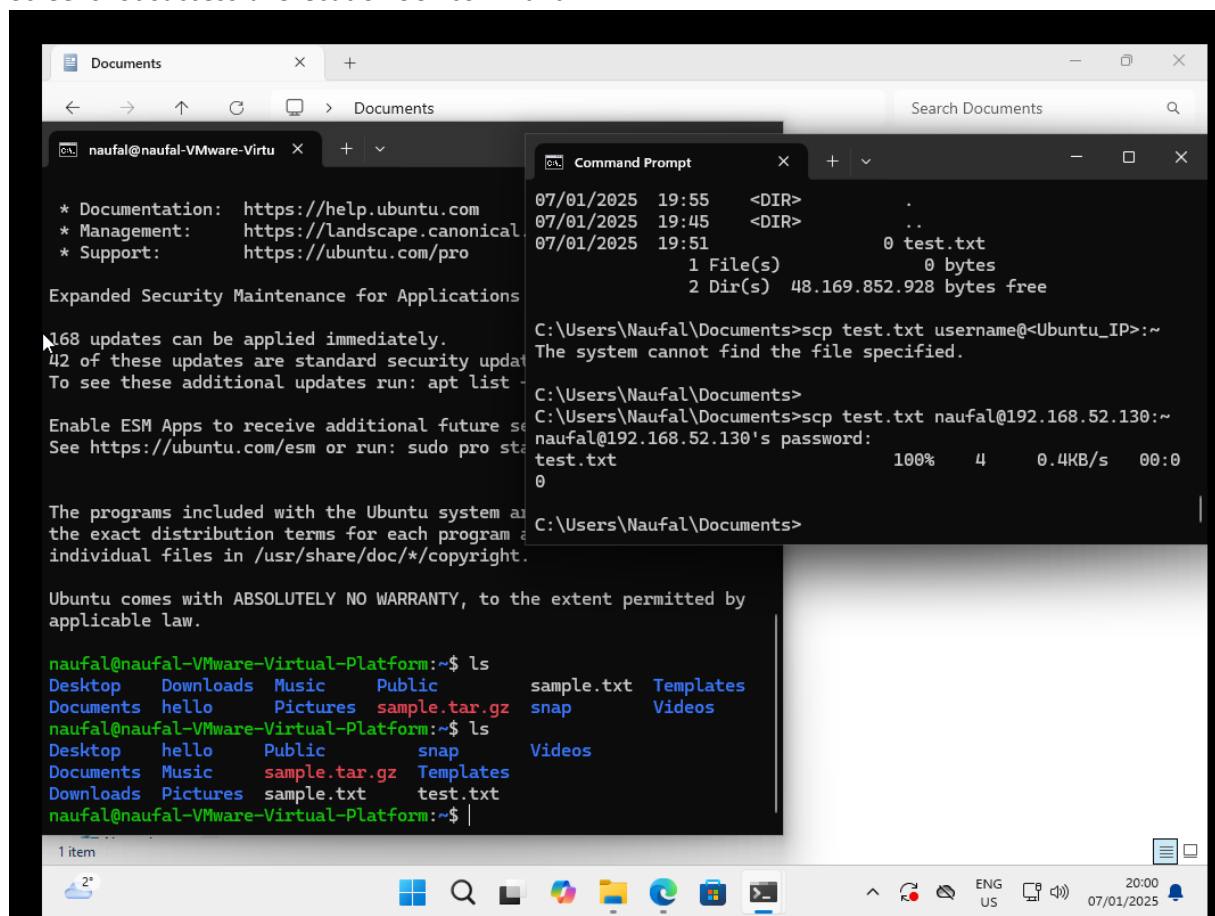
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

naufal@naufal-VMware-Virtual-Platform:~$
```

## Screenshot successful execution SCP command:



```
Documents X + - O X
Documents Search Documents Q

naufal@naufal-VMware-Virtu X + - O X

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Applications

168 updates can be applied immediately.
42 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

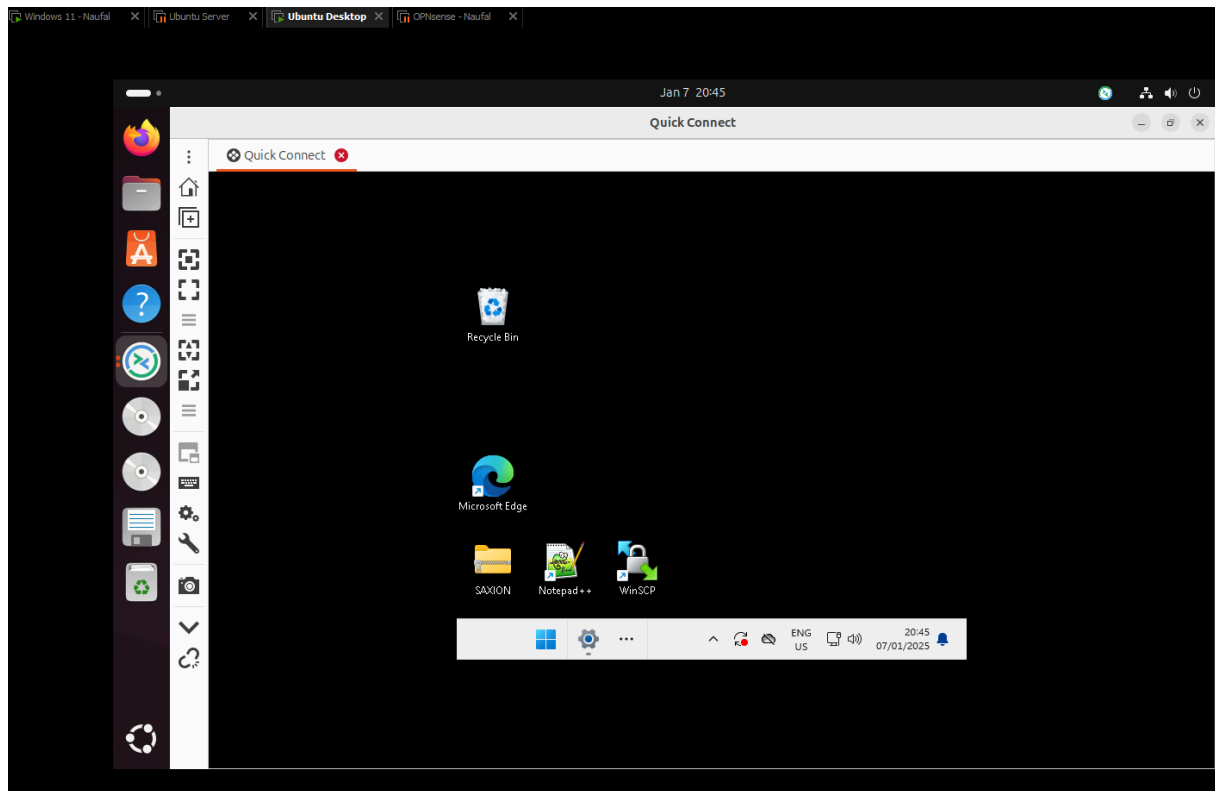
naufal@naufal-VMware-Virtual-Platform:~$ ls
Desktop  Downloads  Music  Public  sample.txt  Templates
Documents hello  Pictures  sample.tar.gz  snap  Videos
naufal@naufal-VMware-Virtual-Platform:~$ ls
Desktop  hello  Public  snap  Videos
Documents Music  sample.tar.gz  Templates
Downloads Pictures  sample.txt  test.txt
naufal@naufal-VMware-Virtual-Platform:~$ |

Command Prompt X + - O X
07/01/2025 19:55 <DIR> .
07/01/2025 19:45 <DIR> ..
07/01/2025 19:51 0 test.txt
1 File(s) 0 bytes
2 Dir(s) 48.169.852.928 bytes free

C:\Users\Naufal\Documents>scp test.txt username@<Ubuntu_IP>:~
The system cannot find the file specified.

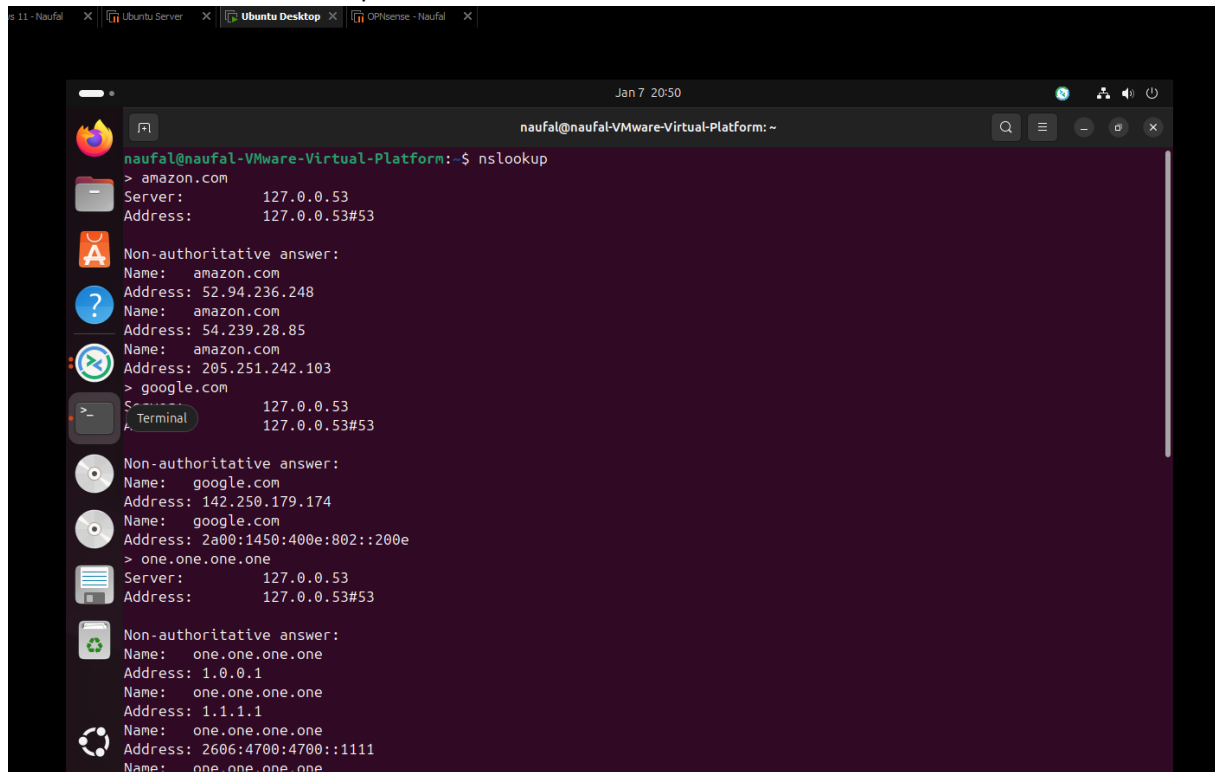
C:\Users\Naufal\Documents>
C:\Users\Naufal\Documents>scp test.txt naufal@192.168.52.130:~
naufal@192.168.52.130's password:
test.txt 100% 4 0.4KB/s 00:00
0
C:\Users\Naufal\Documents>
```

Screenshot remmina:

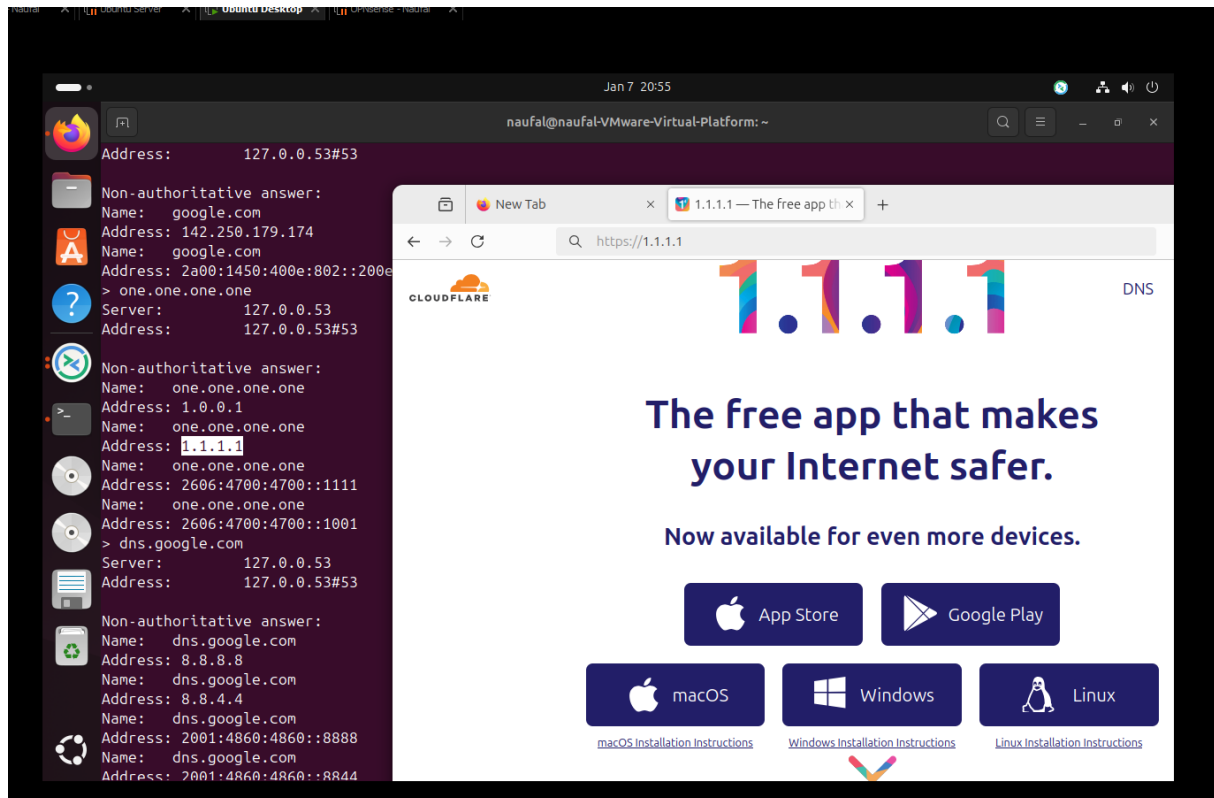


## Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:



Screenshot website visit via IP address:



### Assignment 6.3: subnetting

How many IP addresses are in this network configuration 192.168.110.128/25?

128 addresses

This calculator returns a variety of information regarding Internet Protocol version 4 (IPv4) and IPv6 subnets including possible network addresses, usable host ranges, subnet mask, and IP class, among others.

## IPv4 Subnet Calculator

### Result

IP Address:	192.168.110.128
Network Address:	192.168.110.128
Usable Host IP Range:	192.168.110.129 - 192.168.110.254
Broadcast Address:	192.168.110.255
Total Number of Hosts:	128
Number of Usable Hosts:	126
Subnet Mask:	255.255.255.128
Wildcard Mask:	0.0.0.127
Binary Subnet Mask:	11111111.11111111.11111111.10000000
IP Class:	C
CIDR Notation:	/25
IP Type:	Private
Short:	192.168.110.128 /25
Binary ID:	11000000101010000110111010000000
Integer ID:	3232263808
Hex ID:	0xc0a86e80
in-addr.arpa:	128.110.168.192.in-addr.arpa
IPv4 Mapped Address:	::ffff:c0a8.6e80
6to4 Prefix:	2002:c0a8.6e80::/48

### All 2 of the Possible /25 Networks for 192.168.110.\*

Network Address	Usable Host Range	Broadcast Address:
192.168.110.0	192.168.110.1 - 192.168.110.126	192.168.110.127
192.168.110.128	192.168.110.129 - 192.168.110.254	192.168.110.255

What is the usable IP range to hand out to the connected computers?

Usable range is 192.168.110.129 to 192.168.110.254

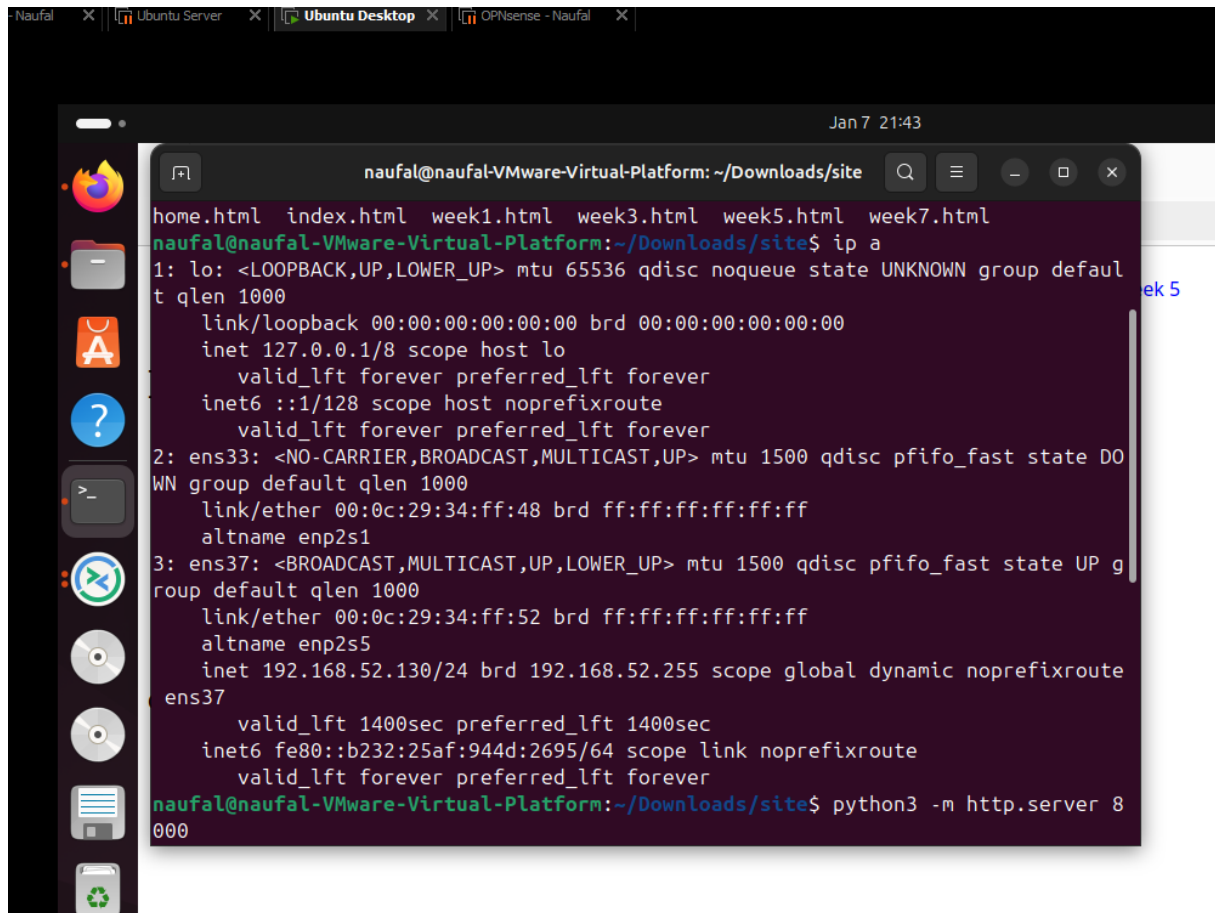
Check your two previous answers with this calculator:

<https://www.calculator.net/ip-subnet-calculator.html>

The network 192.168.110.128 includes 128 addresses in total. However, two of these addresses are reserved one for the network itself 192.168.110.128 and one for broadcasting messages to all devices 192.168.110.255. This means you can use 126 addresses for devices, range from 192.168.110.129 to 192.168.110.254

## Assignment 6.4: HTML

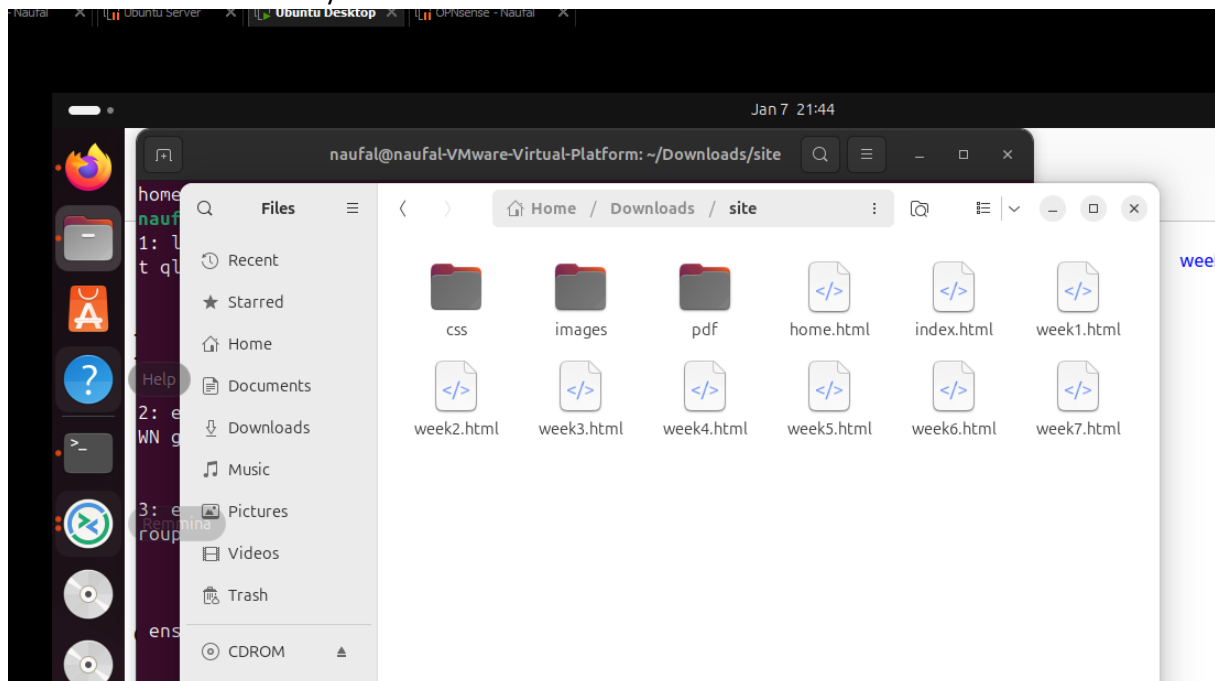
Screenshot IP address Ubuntu VM:



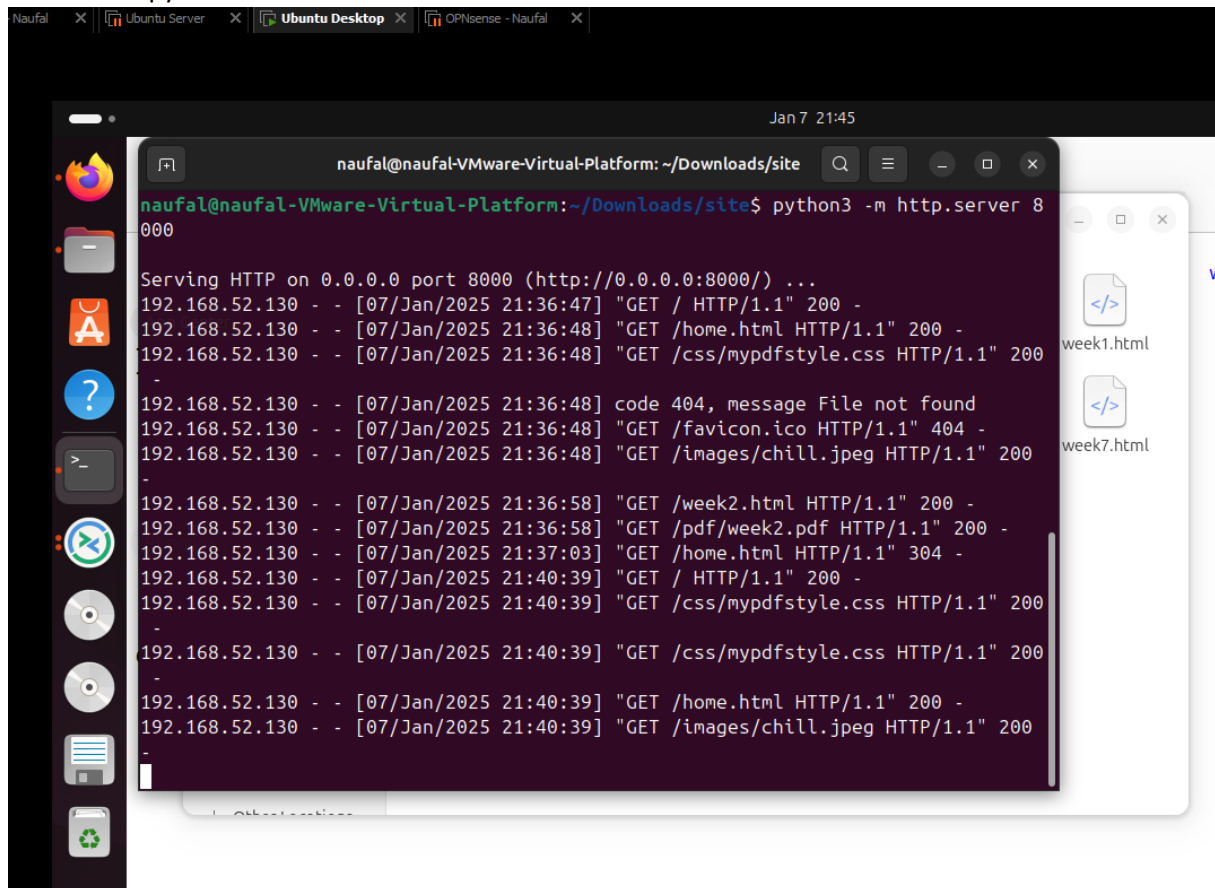
The screenshot shows a terminal window titled "naufal@naufal-VMware-Virtual-Platform: ~/Downloads/site". The terminal output displays the IP configuration for the system. It shows the loopback interface 'lo' with IP 127.0.0.1 and the ethernet interface 'ens37' with IP 192.168.52.130. The command 'ip a' was executed. Below the IP configuration, the command 'python3 -m http.server 8000' was entered, which starts a simple web server on port 8000.

```
naufal@naufal-VMware-Virtual-Platform: ~/Downloads/site
home.html index.html week1.html week3.html week5.html week7.html
naufal@naufal-VMware-Virtual-Platform:~/Downloads/site$ 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: ens33: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 00:0c:29:34:ff:48 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
3: ens37: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:34:ff:52 brd ff:ff:ff:ff:ff:ff
    altname enp2s5
    inet 192.168.52.130/24 brd 192.168.52.255 scope global dynamic noprefixroute ens37
        valid_lft 1400sec preferred_lft 1400sec
    inet6 fe80::b232:25af:944d:2695/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
naufal@naufal-VMware-Virtual-Platform:~/Downloads/site$ python3 -m http.server 8000
```

Screenshot of Site directory contents:



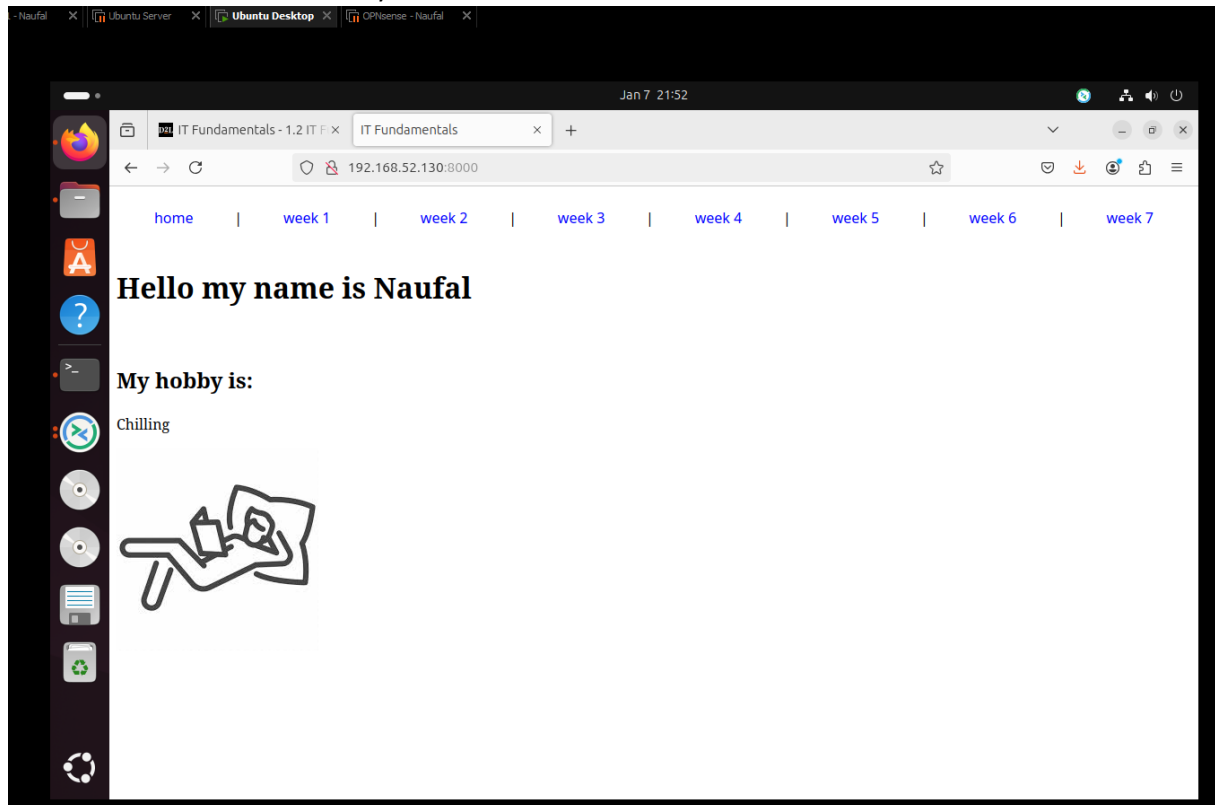
Screenshot python3 webserver command:



The screenshot shows a desktop environment with a terminal window open. The terminal window title is "naufal@naufal-VMware-Virtual-Platform: ~/Downloads/site". The command entered is `python3 -m http.server 8000`. The output shows the server starting on port 8000 and handling several requests from 192.168.52.130. The requests include `GET / HTTP/1.1`, `GET /home.html HTTP/1.1`, `GET /css/mypdfstyle.css HTTP/1.1`, `code 404, message File not found`, `GET /favicon.ico HTTP/1.1`, `GET /images/chill.jpeg HTTP/1.1`, `GET /week2.html HTTP/1.1`, `GET /pdf/week2.pdf HTTP/1.1`, `GET /home.html HTTP/1.1`, `GET / HTTP/1.1`, `GET /css/mypdfstyle.css HTTP/1.1`, `GET /home.html HTTP/1.1`, and `GET /images/chill.jpeg HTTP/1.1`. The status codes are 200 for successful requests and 404 for the file not found error. A file explorer window is open in the background, showing files like `week1.html` and `week7.html`.

```
naufal@naufal-VMware-Virtual-Platform: ~/Downloads/site$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.52.130 - - [07/Jan/2025 21:36:47] "GET / HTTP/1.1" 200 -
192.168.52.130 - - [07/Jan/2025 21:36:48] "GET /home.html HTTP/1.1" 200 -
192.168.52.130 - - [07/Jan/2025 21:36:48] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
192.168.52.130 - - [07/Jan/2025 21:36:48] code 404, message File not found
192.168.52.130 - - [07/Jan/2025 21:36:48] "GET /favicon.ico HTTP/1.1" 404 -
192.168.52.130 - - [07/Jan/2025 21:36:48] "GET /images/chill.jpeg HTTP/1.1" 200 -
192.168.52.130 - - [07/Jan/2025 21:36:58] "GET /week2.html HTTP/1.1" 200 -
192.168.52.130 - - [07/Jan/2025 21:36:58] "GET /pdf/week2.pdf HTTP/1.1" 200 -
192.168.52.130 - - [07/Jan/2025 21:37:03] "GET /home.html HTTP/1.1" 304 -
192.168.52.130 - - [07/Jan/2025 21:40:39] "GET / HTTP/1.1" 200 -
192.168.52.130 - - [07/Jan/2025 21:40:39] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
192.168.52.130 - - [07/Jan/2025 21:40:39] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
192.168.52.130 - - [07/Jan/2025 21:40:39] "GET /home.html HTTP/1.1" 200 -
192.168.52.130 - - [07/Jan/2025 21:40:39] "GET /images/chill.jpeg HTTP/1.1" 200 -
```

## Screenshot web browser visits your site



### Bonus point assignment – week 6

Remember that bitwise java application you've made in week 2? Expand that application so that you can also calculate a network segment as explained in the PowerPoint slides of week 6. Use the bitwise & AND operator. You need to be able to input two Strings. An IP address and a subnet.

IP: 192.168.1.100 and subnet: 255.255.255.224 for /27

Example: 192.168.1.100/27

Calculate the network segment

IP Address: 11000000.10101000.00000001.01100100

Subnet Mask: 11111111.11111111.11111111.11100000

-----  
Network Addr: 11000000.10101000.00000001.01100000

This gives 192.168.1.96 in decimal as the network address.

For a /27 subnet, each segment (or subnet) has 32 IP addresses ( $2^5$ ).

The range of this network segment is from 192.168.1.96 to 192.168.1.127.

Paste source code here, with a screenshot of a working application.



```

Choose an option:
1. Is the number odd?
2. Is the number a power of 2?
3. Two's complement of the number?
4. Calculate network address and range
5. Exit
4
Enter an IP address and subnet mask (e.g. 192.168.1.100
255.255.255.224
Network Address: 192.168.1.96
Range: 192.168.1.97 to 192.168.1.128

```

```
import nl.saxion.app.SaxionApp;
```

```
public class Application implements Runnable {
```

```
    public static void main(String[] args) {
        SaxionApp.start(new Application(), 350, 500);
    }
```

```
    public void run() {
        while (true) {
            SaxionApp.println("Choose an option:");
            SaxionApp.println("1. Is the number odd?");
            SaxionApp.println("2. Is the number a power of 2?");
            SaxionApp.println("3. Two's complement of the number?");
            SaxionApp.println("4. Calculate network address and range");
            SaxionApp.println("5. Exit");
```

```
            int choice = SaxionApp.readInt();
```

```
            if (choice == 5) {
                SaxionApp.println("DOE!");
                break;
            }
```

```
            int number = 0;
            if (choice != 4) {
                number = SaxionApp.readInt("Enter a number: ");
            }
```

```
            switch (choice) {
                case 1 -> SaxionApp.println("Is number odd? " + isOdd(number));
            }
```

```

        case 2 -> SaxionApp.println("Is number a power of 2? " + isPowerOfTwo(number));
        case 3 -> SaxionApp.println("Two's complement: " + twosComplement(number));
        case 4 -> calculateNetworkSegment();
        default -> SaxionApp.println("Invalid choice. Try again!!");
    }
}

private boolean isOdd(int number) {
    return (number & 1) == 1;
}

private boolean isPowerOfTwo(int number) {
    return number > 0 && (number & (number - 1)) == 0;
}

private int twosComplement(int number) {
    return ~number + 1;
}

private void calculateNetworkSegment() {
    SaxionApp.println("Enter an IP address and subnet mask: ");

    String ip = SaxionApp.readString();
    String subnet = SaxionApp.readString();

    String networkAddress = calculateNetworkAddress(ip, subnet);
    SaxionApp.println("Network Address: " + networkAddress);

    String[] networkParts = networkAddress.split("\\.");
    int networkStart = Integer.parseInt(networkParts[3]) + 1;
    int networkEnd = networkStart + 31;

    SaxionApp.println("Range: " + networkParts[0] + "." + networkParts[1] + "." + networkParts[2]
+ "." + networkStart
    + " to " + networkParts[0] + "." + networkParts[1] + "." + networkParts[2] + "." +
networkEnd);
}

private String calculateNetworkAddress(String ip, String subnet) {
    String[] ipParts = ip.split("\\.");
    String[] subnetParts = subnet.split("\\.");

    int[] networkParts = new int[4];
    for (int i = 0; i < 4; i++) {
        networkParts[i] = Integer.parseInt(ipParts[i]) & Integer.parseInt(subnetParts[i]);
    }
}

```

```
        return networkParts[0] + "." + networkParts[1] + "." + networkParts[2] + "." + networkParts[3];  
    }  
}
```

Ready? Save this file and export it as a pdf file with the name: [week6.pdf](#)