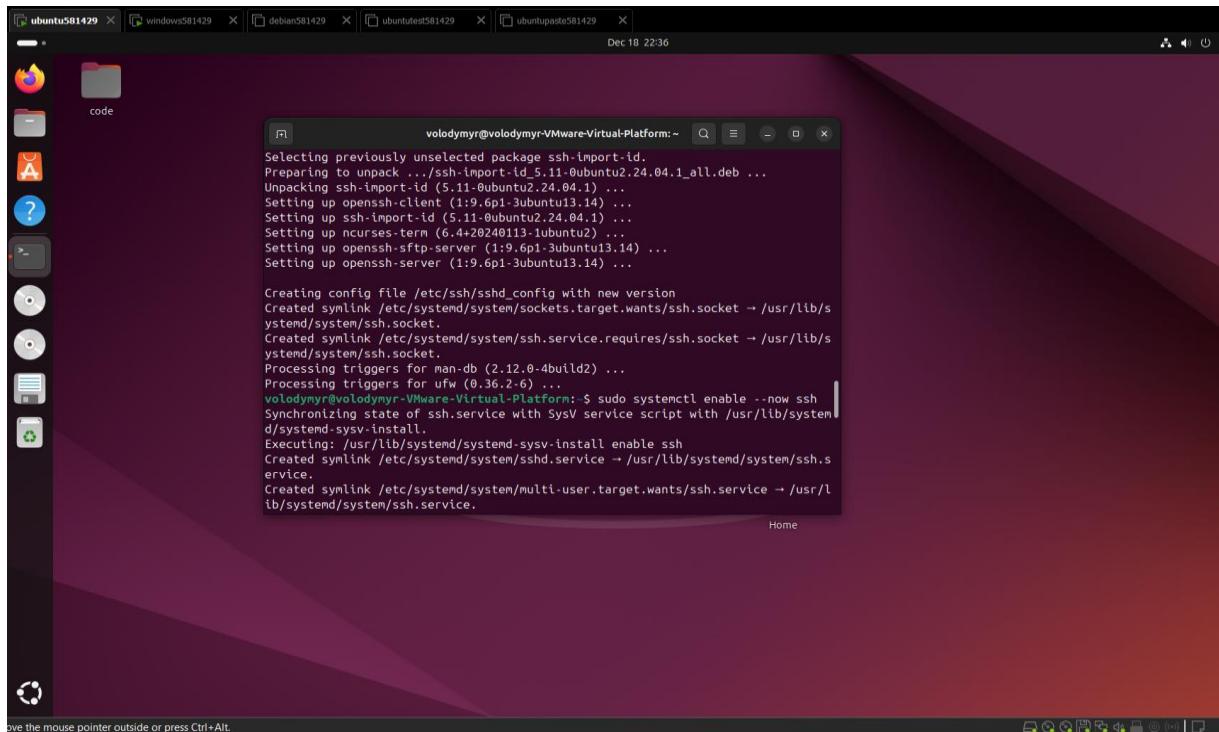


# Template Week 6 – Networking

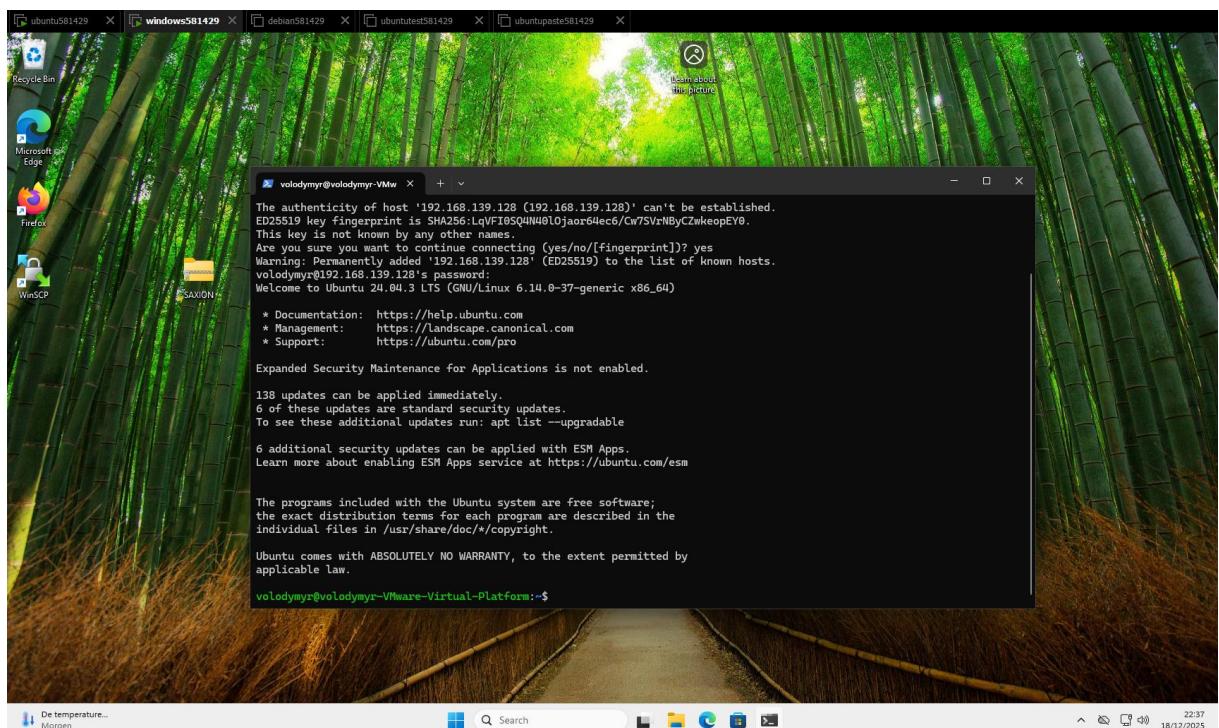
Student number: 581429

## Assignment 6.1: Working from home

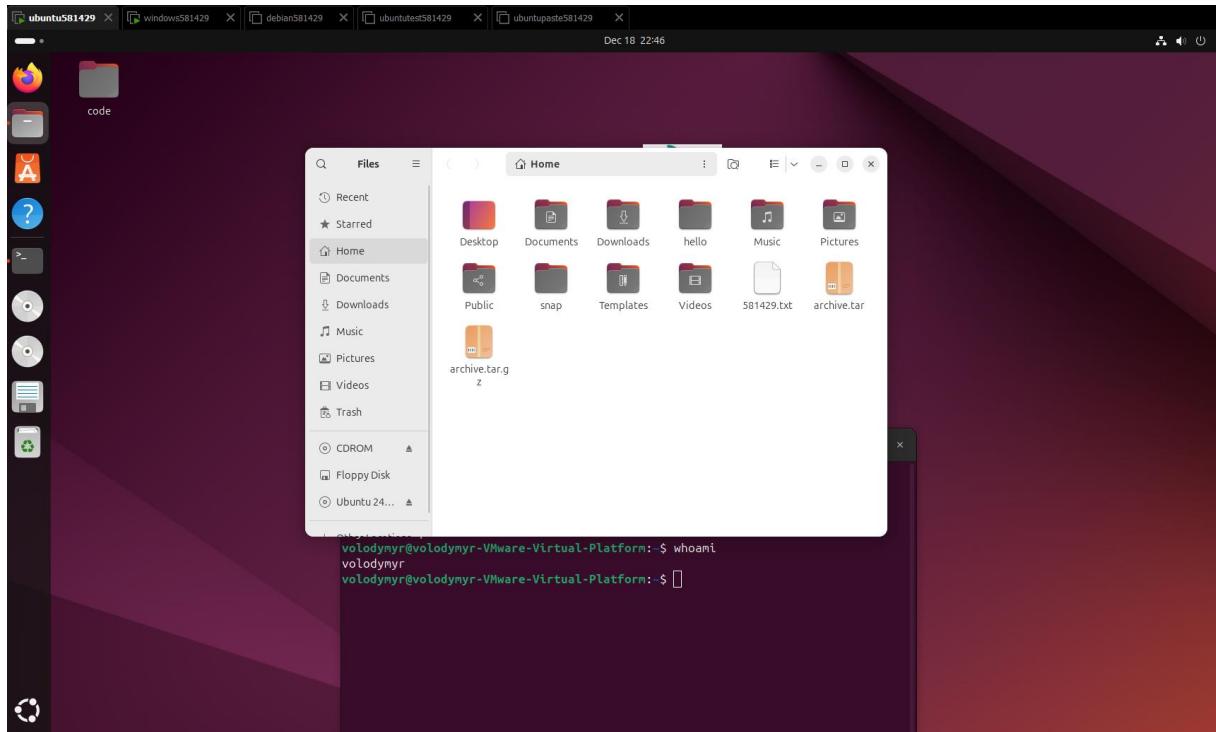
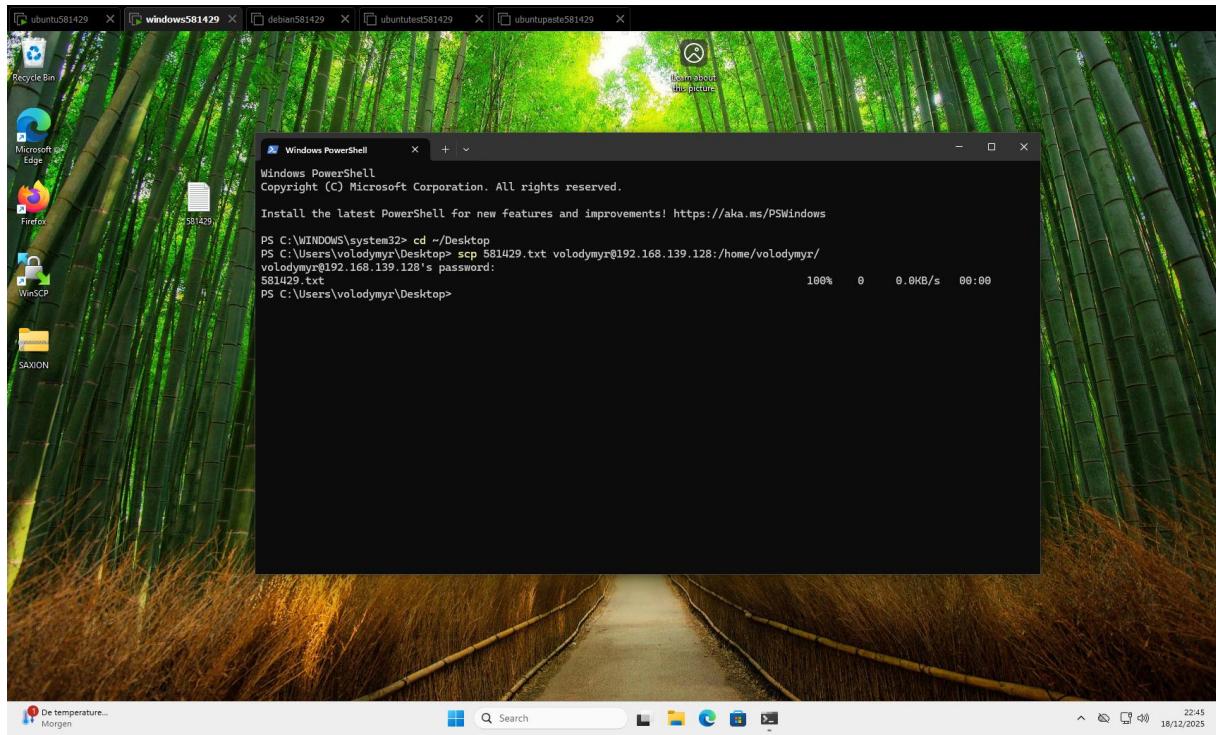
Screenshot installation openssh-server:



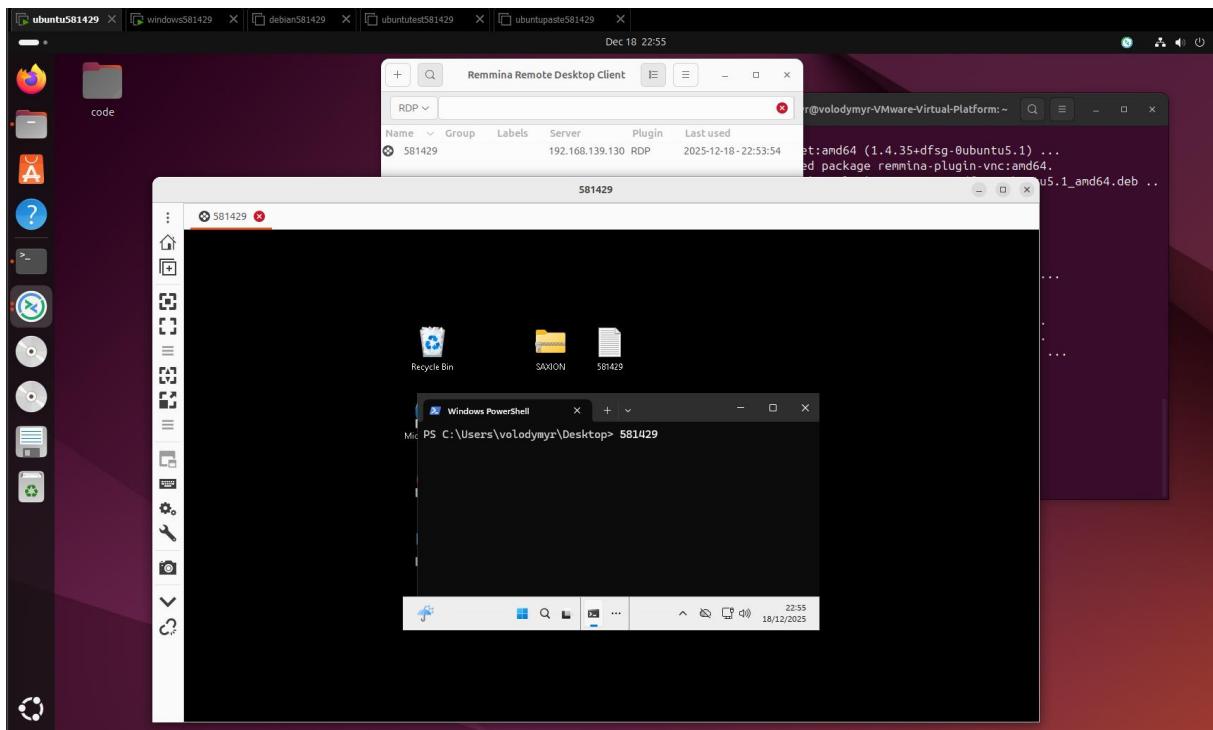
Screenshot successful SSH command execution:



Screenshot successful execution SCP command:

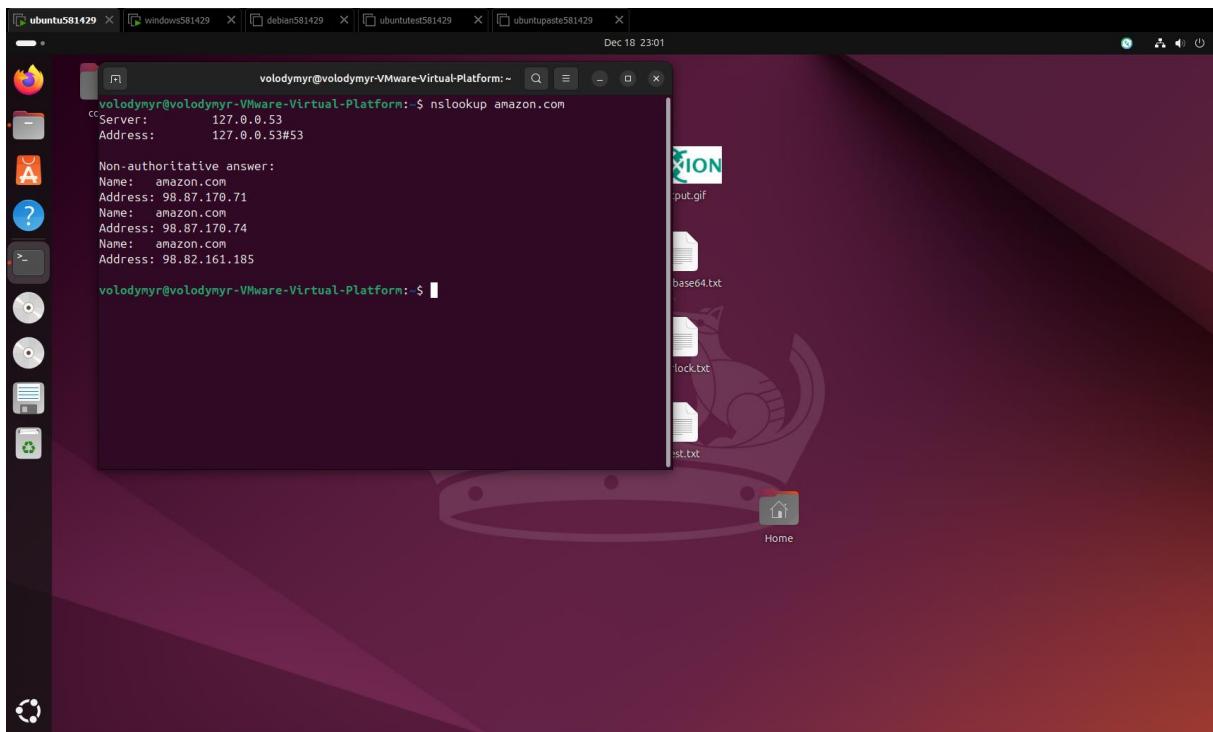


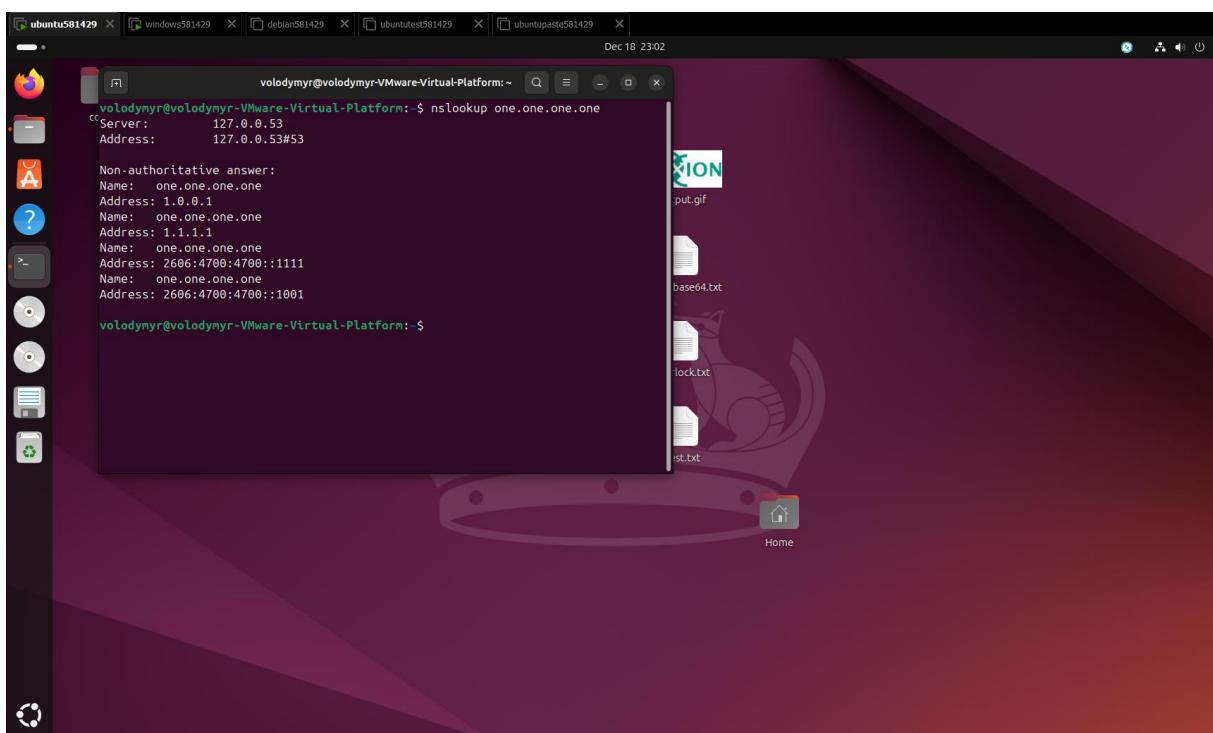
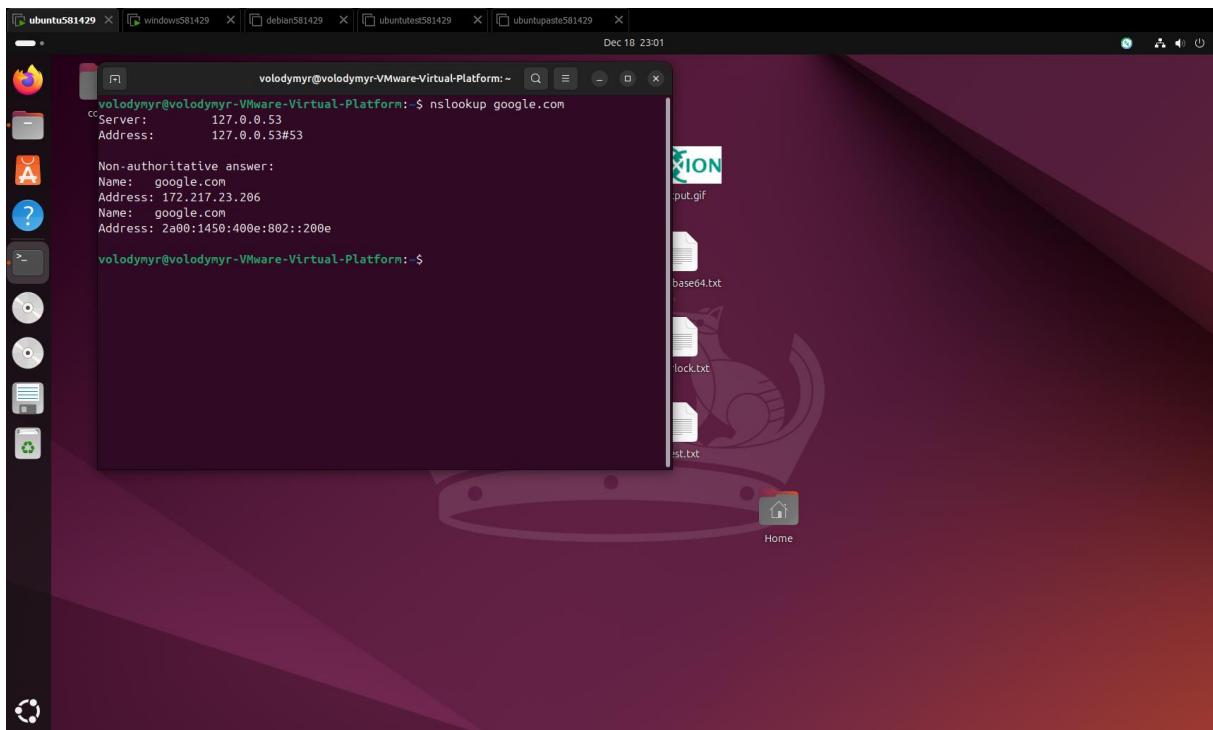
Screenshot remmina:

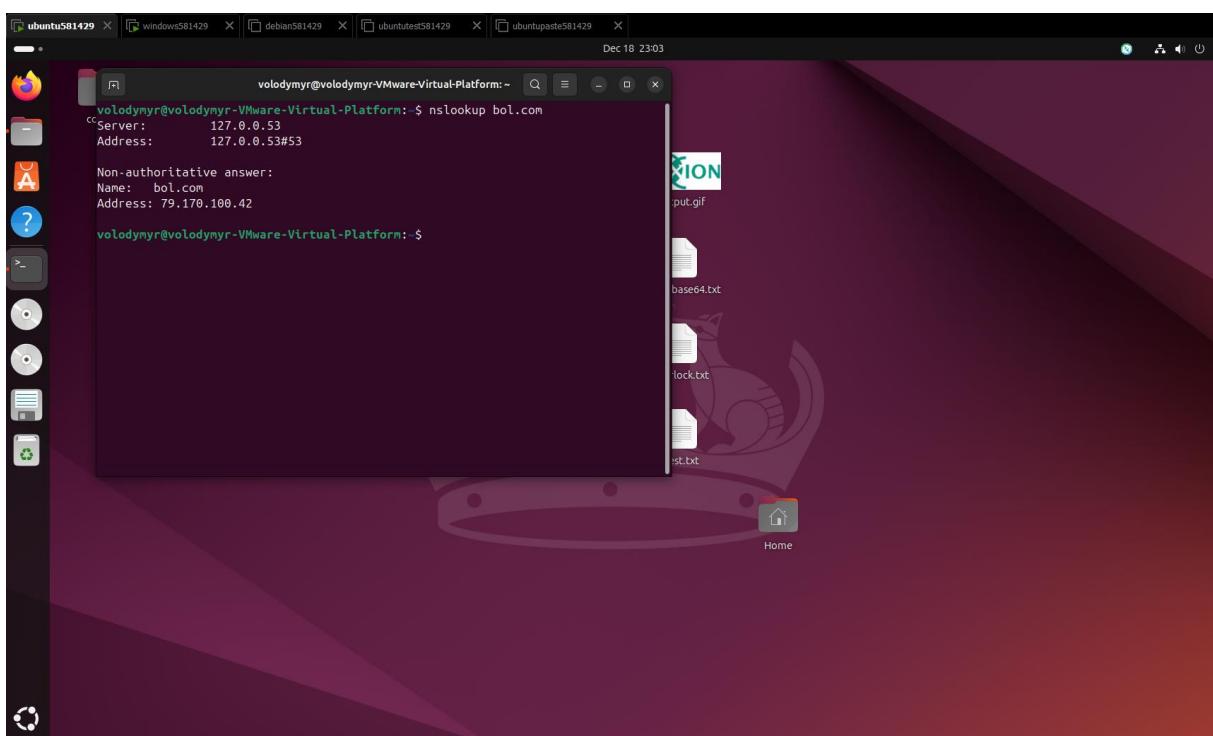
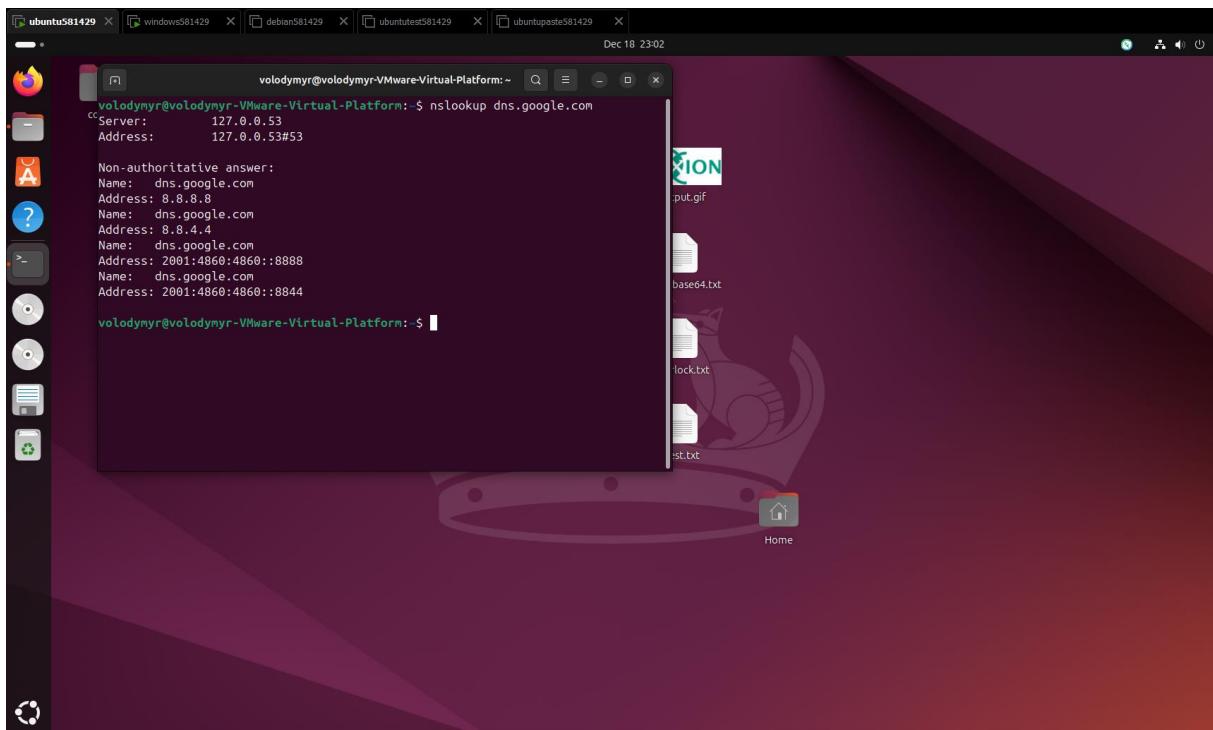


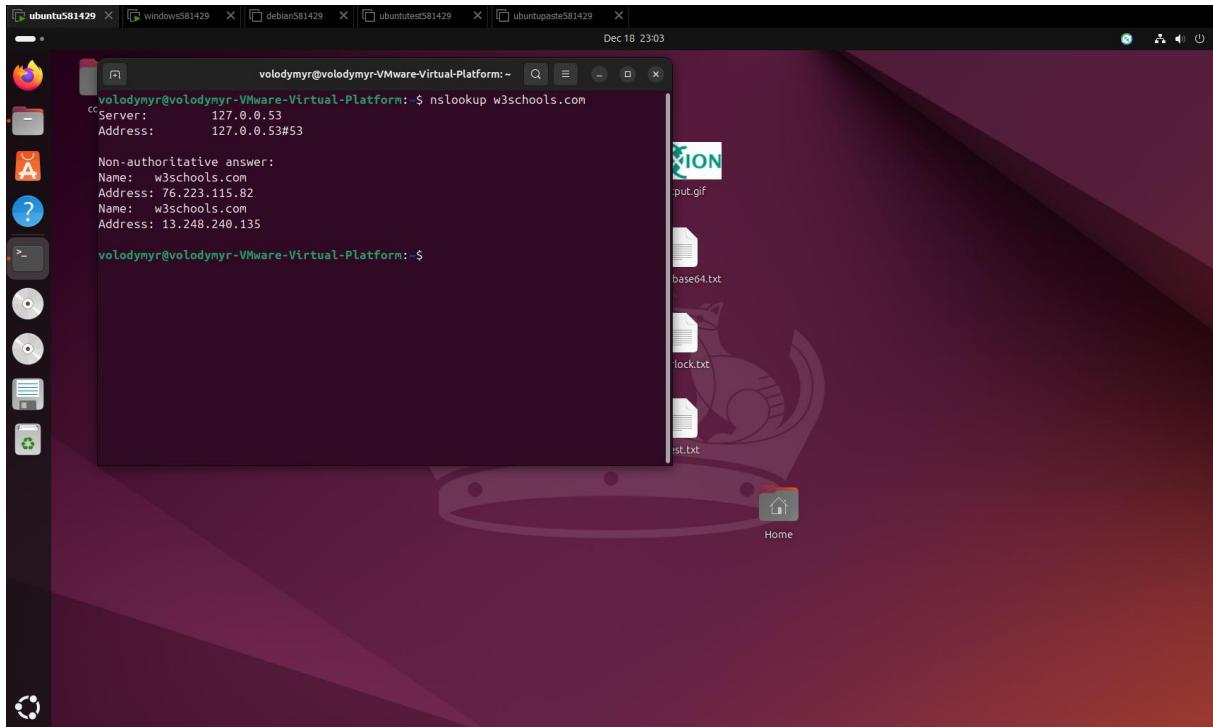
## Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

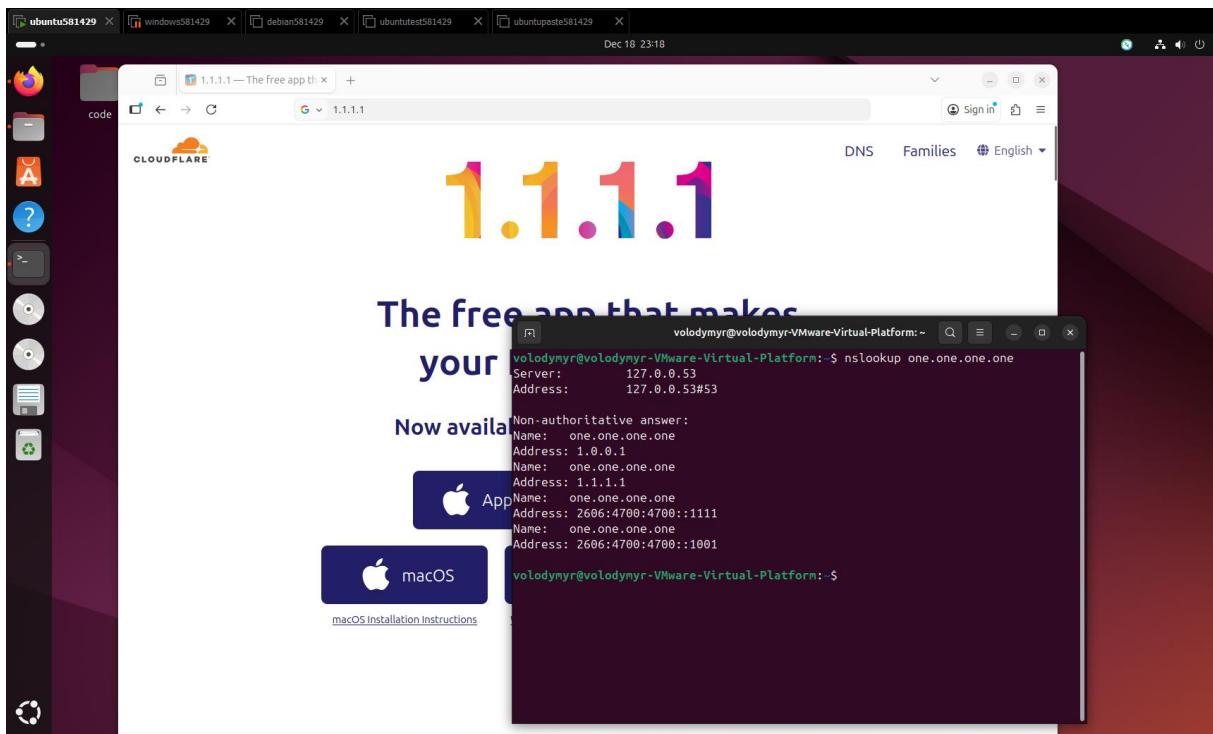








Screenshot website visit via IP address:



### Assignment 6.3: subnetting

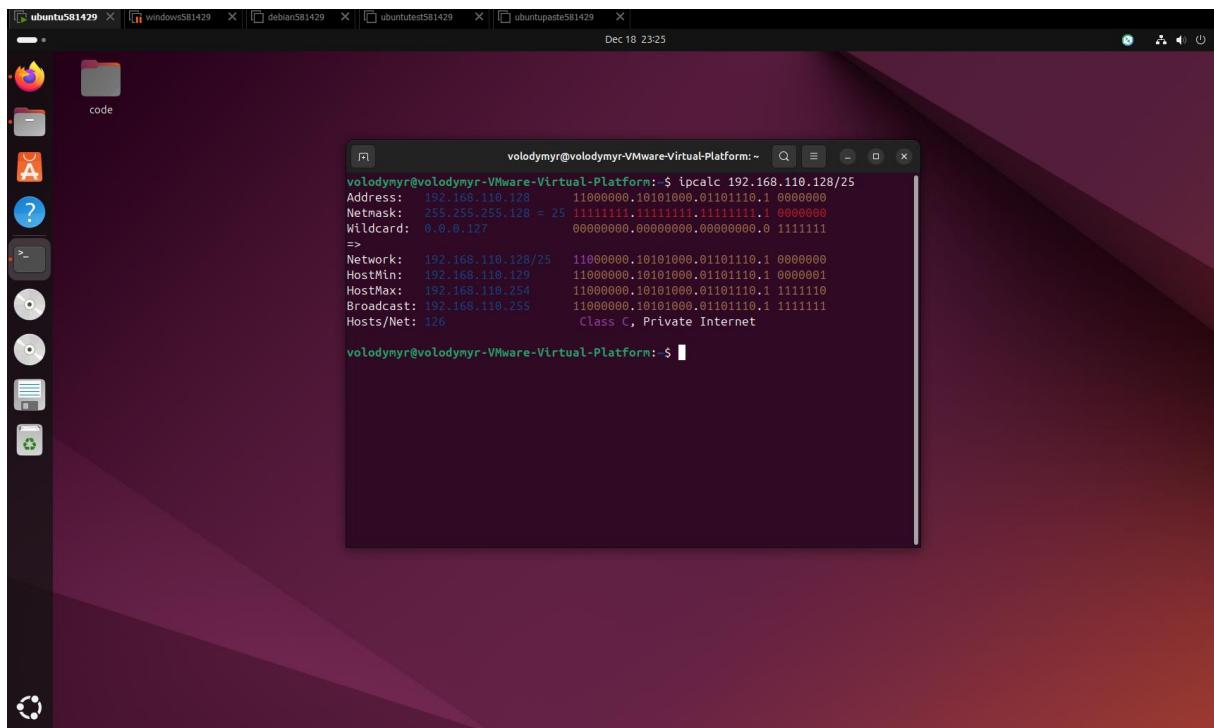
How many IP addresses are in this network configuration 192.168.110.128/25?

128

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

**192.168.110.129 – 192.168.110.254**

Check your two previous answers with this Linux command: `ipcalc 192.168.110.128/25`



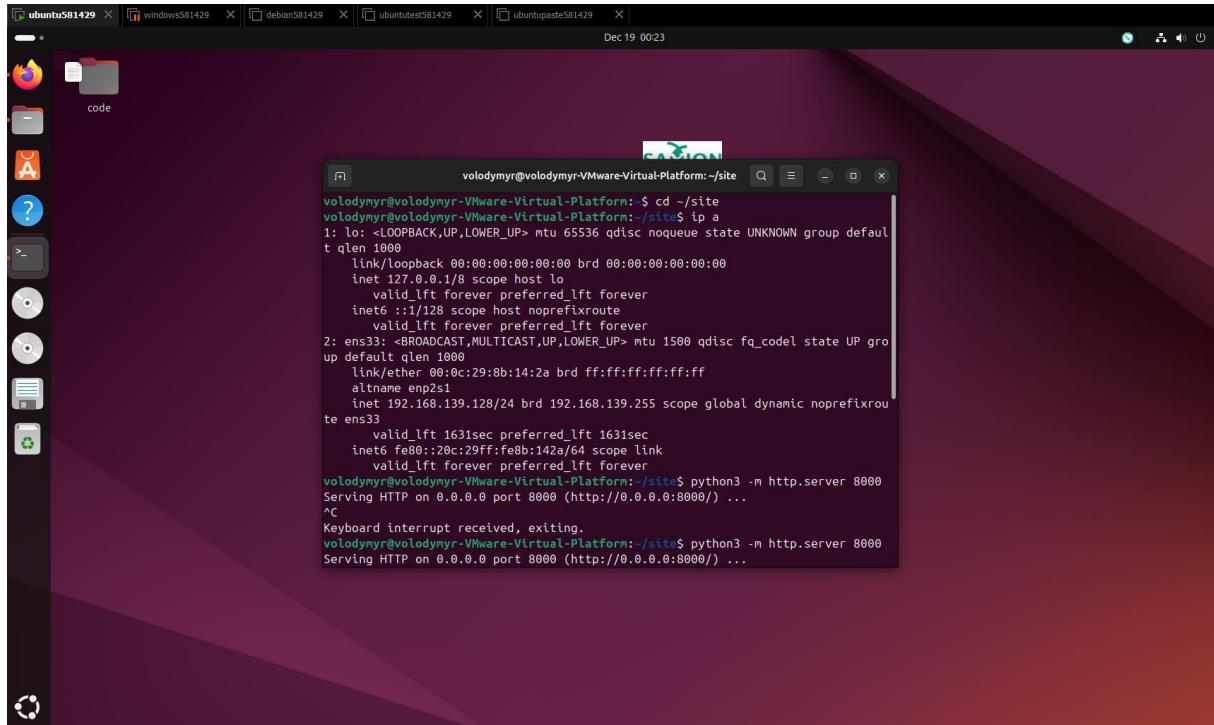
```
volodymyr@volodymyr-VMware-Virtual-Platform: ~ ipcalc 192.168.110.128/25
Address: 192.168.110.128      11000000.10101000.01101110.1 00000000
Netmask: 255.255.255.128 = 25 11111111.11111111.11111111.0 00000000
Wildcard: 0.0.0.127          00000000.00000000.00000000.0 11111111
=>
Network: 192.168.110.128    11000000.10101000.01101110.1 00000000
HostMin: 192.168.110.129    11000000.10101000.01101110.1 00000001
HostMax: 192.168.110.254    11000000.10101000.01101110.1 11111110
Broadcast: 192.168.110.255  11000000.10101000.01101110.1 11111111
Hosts/Net: 126              Class C, Private Internet
```

Explain the above calculation in your own words.

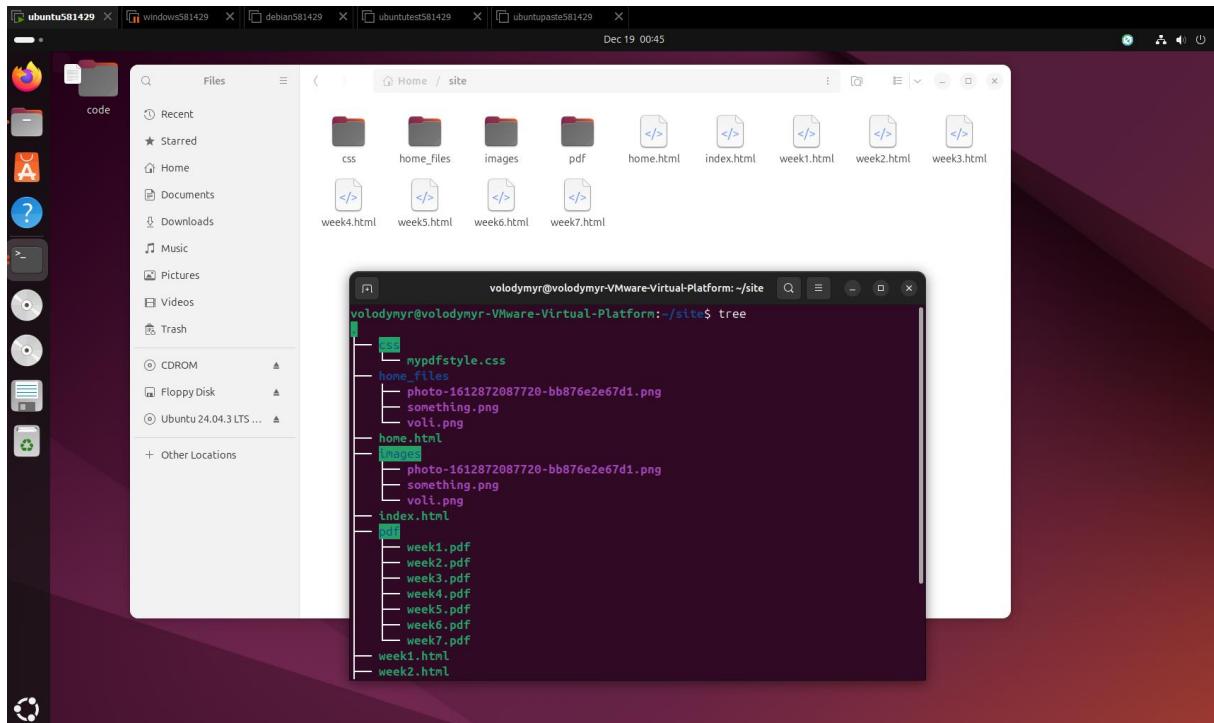
The subnet 192.168.110.128/25 means that 25 bits are used for the network and 7 bits are left for hosts. Since 7 bits are available, the total number of addresses is  $2^7 = 128$ . Out of these 128 addresses, one is reserved as the network ID (192.168.110.128) and one is reserved as the broadcast address (192.168.110.255). That leaves 126 usable addresses for devices. The usable range starts at 192.168.110.129 and ends at 192.168.110.254

## Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:



Screenshot of Site directory contents:



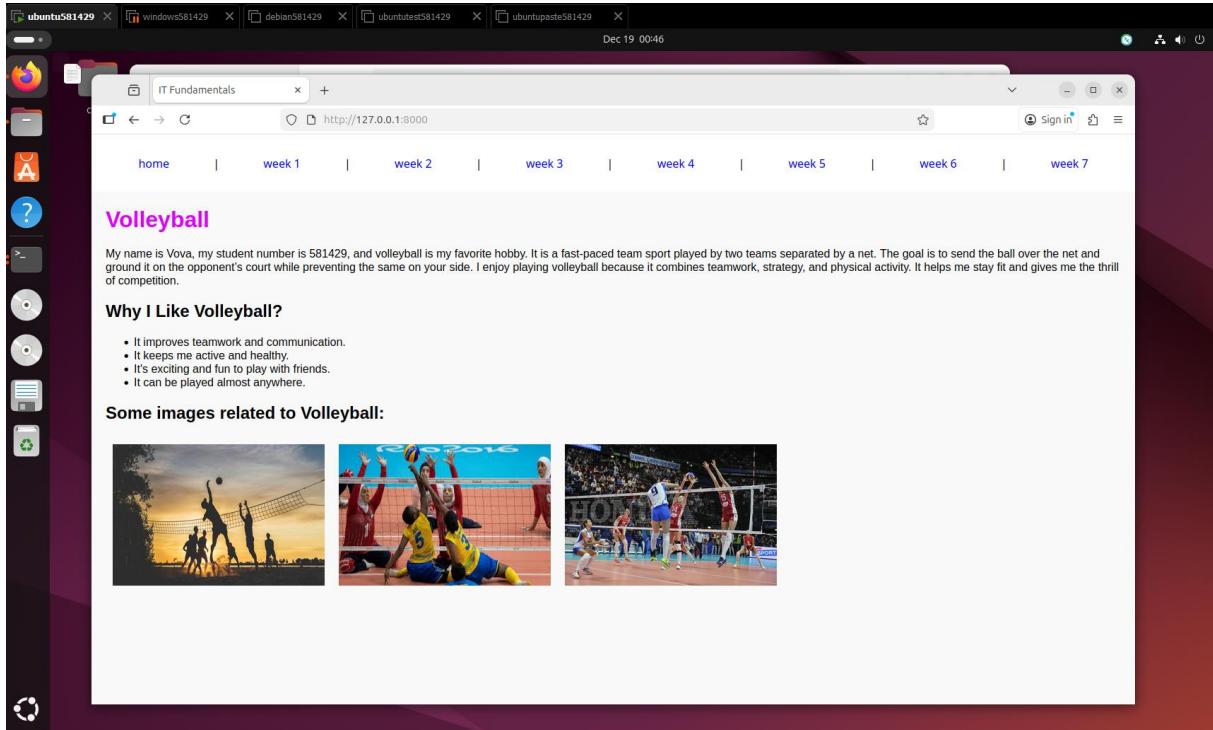
Screenshot python3 webserver command:

```

Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
^C
Keyboard interrupt received, exiting.
volodymyr@volodymyr-VMware-Virtual-Platform:~/site$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [19/Dec/2025 00:17:44] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2025 00:17:44] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2025 00:17:44] "GET /home.html HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2025 00:17:44] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [19/Dec/2025 00:19:07] "GET /week1.html HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2025 00:19:07] "GET /pdf/week1.pdf HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2025 00:19:08] "GET /week2.html HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2025 00:19:08] "GET /pdf/week2.pdf HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2025 00:19:09] "GET /week3.html HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2025 00:19:09] "GET /pdf/week3.pdf HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2025 00:19:53] "GET / HTTP/1.1" 304 -
127.0.0.1 - - [19/Dec/2025 00:19:53] "GET /home.html HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2025 00:19:53] "GET /home_files/photo-1612872087720-bb876e2e67d1.png HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2025 00:19:53] "GET /home_files/voli.png HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2025 00:19:53] "GET /home_files/something.png HTTP/1.1" 200 -
0 -

```

## Screenshot web browser visits your site



## Assignment 6.5: Network segment

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.

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

public class Application implements Runnable {

    public static void main(String[] args) {
        SaxionApp.start(new Application(), 800, 800);
    }

    public void run() {

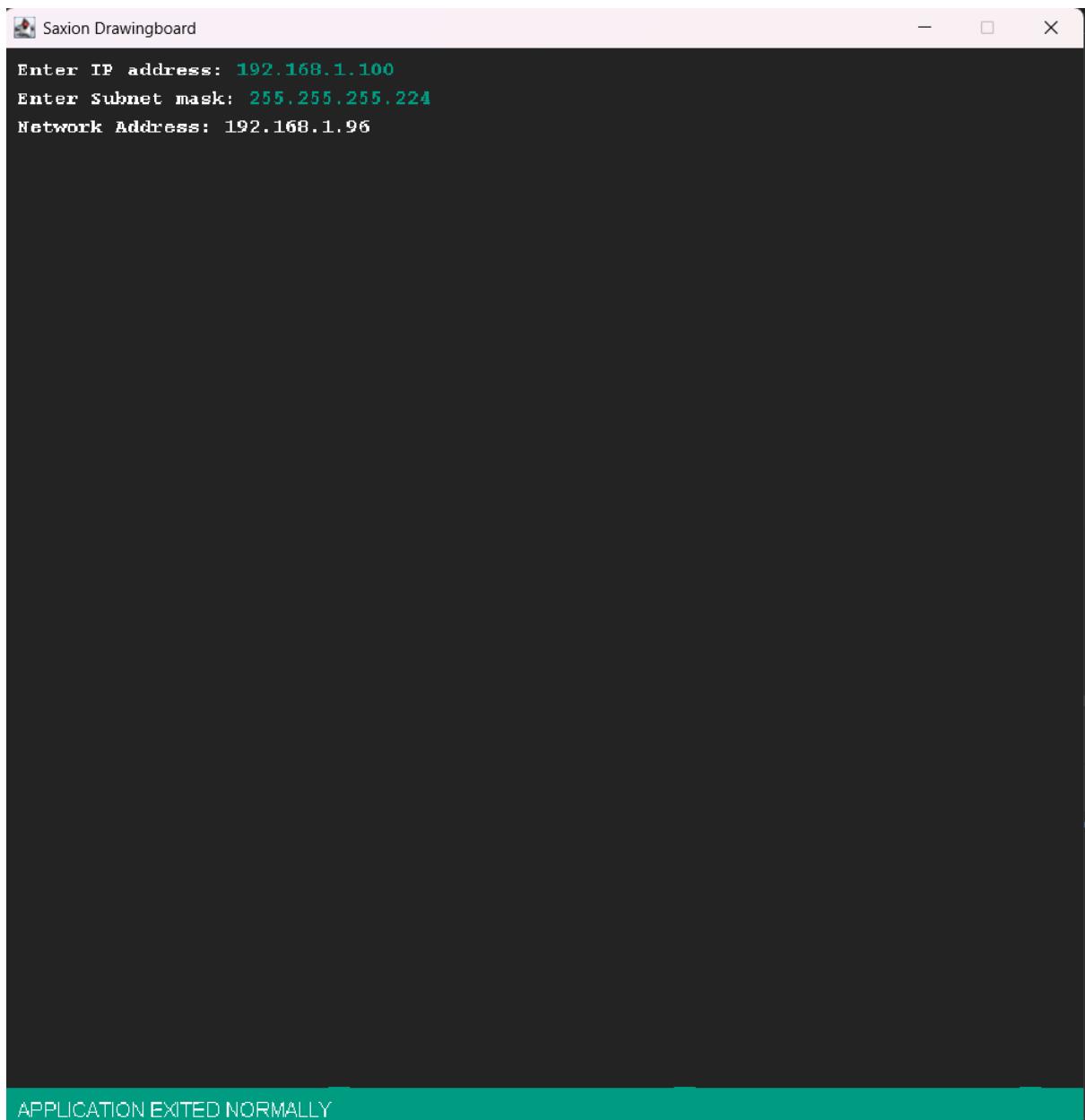
        // Input IP and subnet mask using SaxionApp
        SaxionApp.print("Enter IP address: ");
        String ipStr = SaxionApp.readString();
        SaxionApp.print("Enter Subnet mask: ");
        String subnetStr = SaxionApp.readString();

        // Split into octets
        String[] ipParts = ipStr.split("\\.");
        String[] subnetParts = subnetStr.split("\\.");

        int[] ip = new int[4];
        int[] subnet = new int[4];
        int[] network = new int[4];

        // Calculate network address
        for (int i = 0; i < 4; i++) {
            ip[i] = Integer.parseInt(ipParts[i]);
            subnet[i] = Integer.parseInt(subnetParts[i]);
            network[i] = ip[i] & subnet[i];
        }
    }
}
```

```
        subnet[i] = Integer.parseInt(subnetParts[i]);
        network[i] = ip[i] & subnet[i]; // bitwise AND
    }
    // Print network address
    SaxionApp.println("Network Address: " + network[0] + "." + network[1] + "." + network[2] + "."
+ network[3]);
}
}
```



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