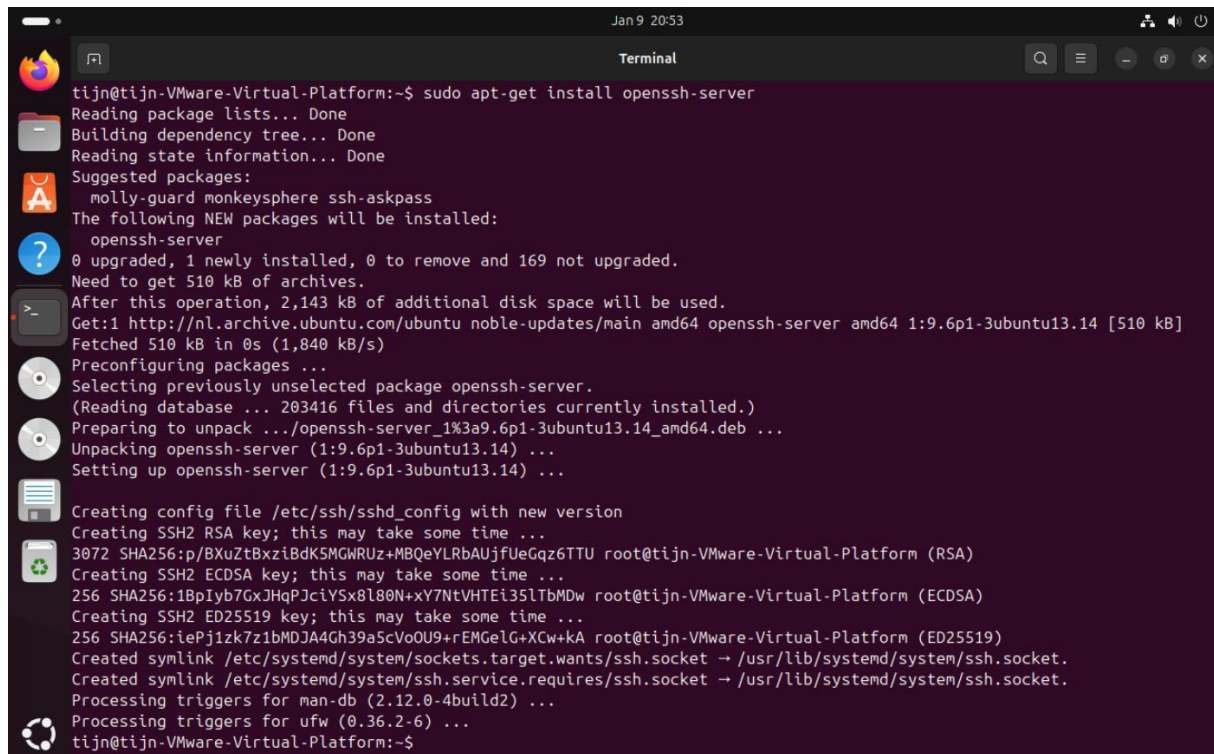


# Template Week 6 – Networking

Student number: 591007

## Assignment 6.1: Working from home

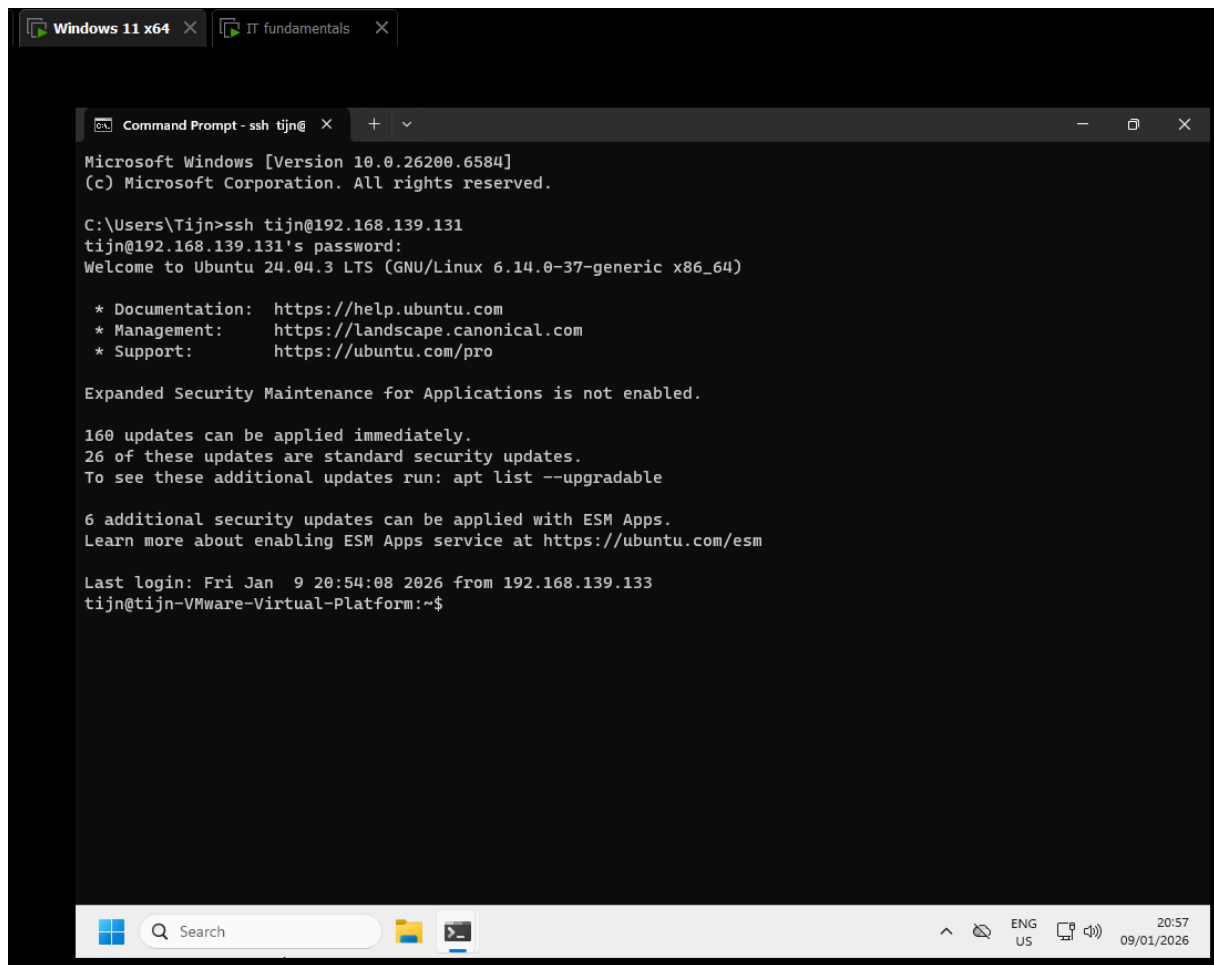
Screenshot installation openssh-server:



```
tiijn@tiijn-VMware-Virtual-Platform:~$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  openssh-server
0 upgraded, 1 newly installed, 0 to remove and 169 not upgraded.
Need to get 510 kB of archives.
After this operation, 2,143 kB of additional disk space will be used.
Get:1 http://nl.archive.ubuntu.com/ubuntu/noble-updates/main amd64 openssh-server amd64 1:9.6p1-3ubuntu13.14 [510 kB]
Fetched 510 kB in 0s (1,840 kB/s)
Preconfiguring packages ...
Selecting previously unselected package openssh-server.
(Reading database ... 203416 files and directories currently installed.)
Preparing to unpack .../openssh-server_1%3a9.6p1-3ubuntu13.14_amd64.deb ...
Unpacking openssh-server (1:9.6p1-3ubuntu13.14) ...
Setting up openssh-server (1:9.6p1-3ubuntu13.14) ...

Creating config file /etc/ssh/sshd_config with new version
Creating SSH2 RSA key; this may take some time ...
3072 SHA256:p/BXuZtBxzIBdK5MGWRUz+MBQeYLRbAUjUeGqz6TTU root@tiijn-VMware-Virtual-Platform (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:1BpIyb7GxJHqPJciYSx8l80N+xy7NtVHTEi35LTbMDw root@tiijn-VMware-Virtual-Platform (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:iePj1zk7z1bMDJA4Gh39a5cVoOU9+rEMGelG+XCw+kA root@tiijn-VMware-Virtual-Platform (ED25519)
Created symlink /etc/systemd/system/sockets.target.wants/ssh.socket → /usr/lib/systemd/system/ssh.socket.
Created symlink /etc/systemd/system/ssh.service.requires/ssh.socket → /usr/lib/systemd/system/ssh.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for ufw (0.36.2-6) ...
tiijn@tiijn-VMware-Virtual-Platform:~$
```

Screenshot successful SSH command execution:



Screenshot successful execution SCP command:

```
Command Prompt - ssh tijn X + -
Microsoft Windows [Version 10.0.26200.6584]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Tijn>echo Dit is een test bestand van Tijn! > testfile.txt

C:\Users\Tijn>scp testfile.txt tijn@192.168.139.131:/home/tijn/
tijn@192.168.139.131's password:
testfile.txt                                     100% 36    7.0KB/s  00:00

C:\Users\Tijn>ssh tijn@192.168.139.131
tijn@192.168.139.131's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-37-generic x86_64)

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

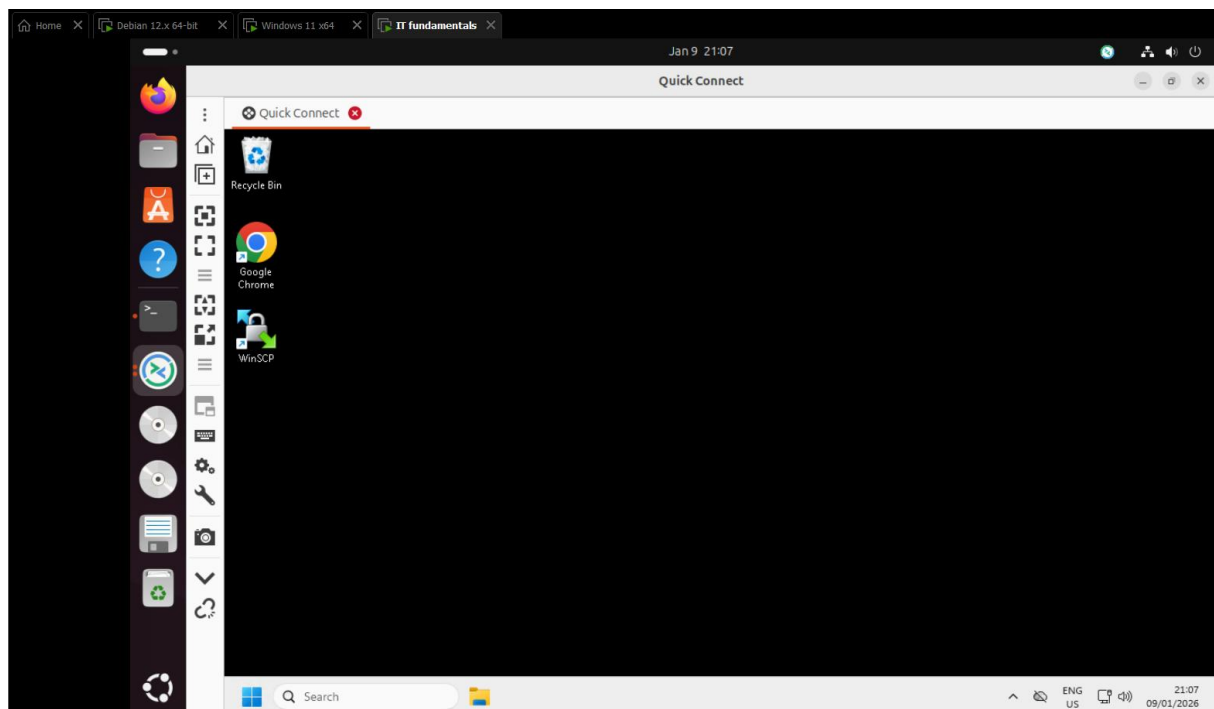
Expanded Security Maintenance for Applications is not enabled.

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

6 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

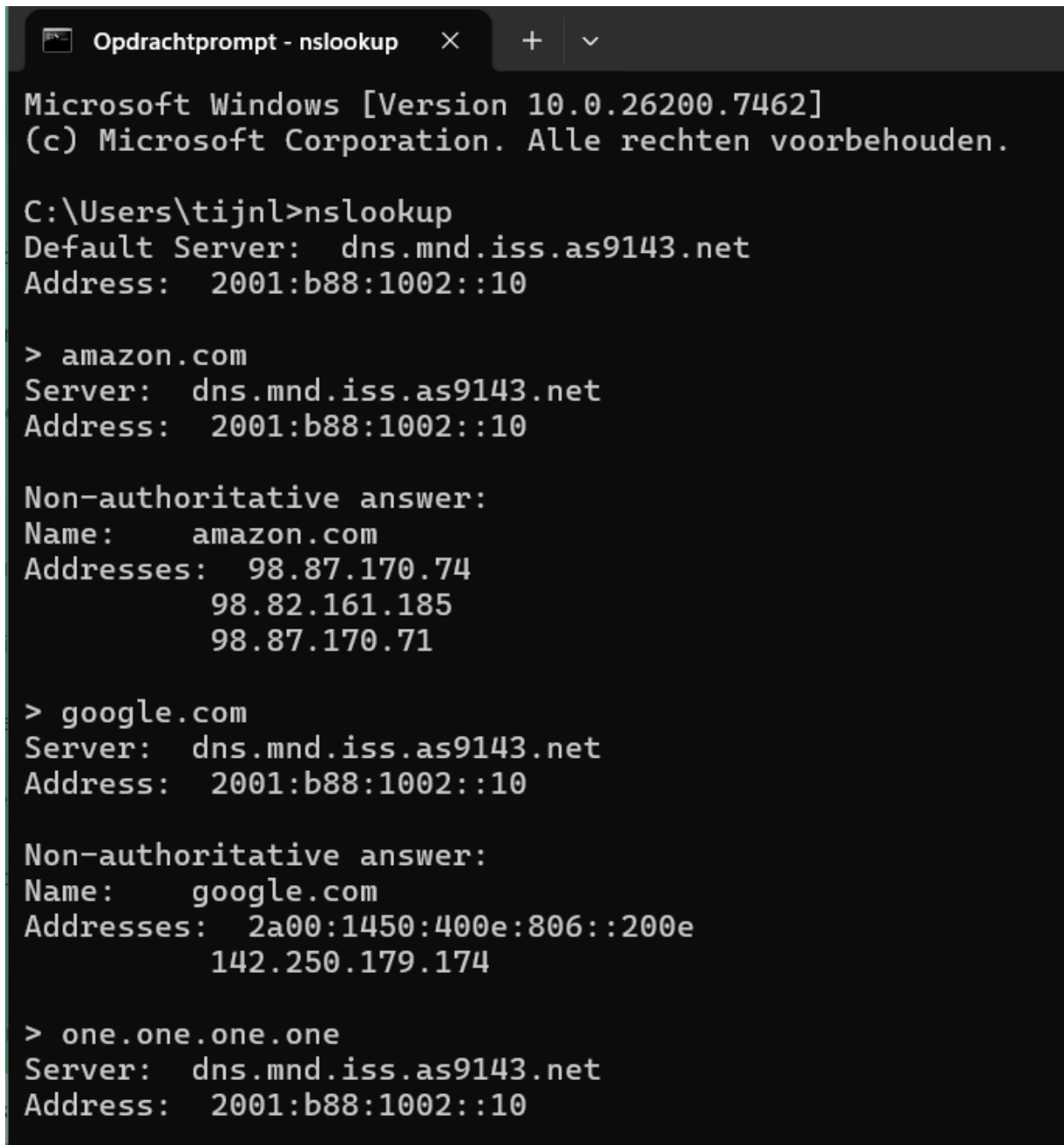
Last login: Fri Jan  9 20:57:37 2026 from 192.168.139.133
tijn@tijn-VMware-Virtual-Platform:~$ ls
Desktop Documents Downloads hello Music Pictures Public snap Templates testfile.txt Videos
tijn@tijn-VMware-Virtual-Platform:~$ cat testfile.txt
Dit is een test bestand van Tijn!
tijn@tijn-VMware-Virtual-Platform:~$ |
```

Screenshot remmina:



## Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:



```
Opdrachtprompt - nslookup

Microsoft Windows [Version 10.0.26200.7462]
(c) Microsoft Corporation. Alle rechten voorbehouden.

C:\Users\tijnl>nslookup
Default Server:  dns.mnd.iss.as9143.net
Address:  2001:b88:1002::10

> amazon.com
Server:  dns.mnd.iss.as9143.net
Address:  2001:b88:1002::10

Non-authoritative answer:
Name:    amazon.com
Addresses:  98.87.170.74
           98.82.161.185
           98.87.170.71

> google.com
Server:  dns.mnd.iss.as9143.net
Address:  2001:b88:1002::10

Non-authoritative answer:
Name:    google.com
Addresses:  2a00:1450:400e:806::200e
           142.250.179.174

> one.one.one.one
Server:  dns.mnd.iss.as9143.net
Address:  2001:b88:1002::10
```

Screenshot website visit via IP address:



### Assignment 6.3: subnetting

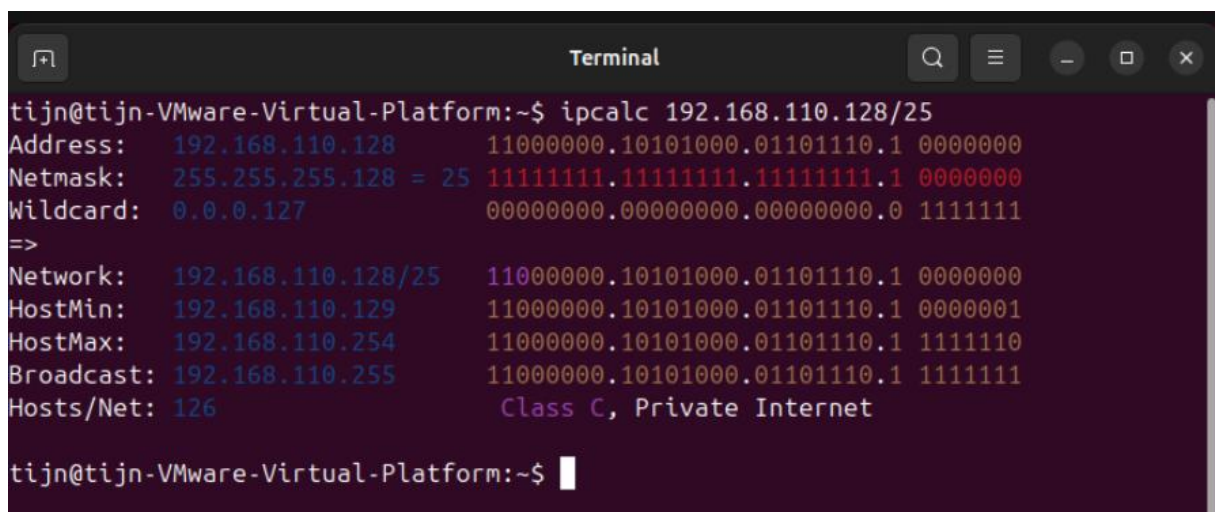
How many IP addresses are in this network configuration 192.168.110.128/25?

128 IP adressen want /25 betekent dat de eerste 25 bits voor het netwerk zijn. Dat laat dus maar 7 bits over voor unieke IP adressen en 2 tot de macht 7 is 128.

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

192.168.110.129 t/m 192.168.110.254 dit zijn er 2 minder. Dit komt doordat het eerste adres wordt gebruikt voor het netwerk en het laatste adres voor broadcasting.

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



```
tijn@tijn-VMware-Virtual-Platform:~$ ipcalc 192.168.110.128/25
Address:    192.168.110.128      11000000.10101000.01101110.1 0000000
Netmask:    255.255.255.128 = 25 11111111.11111111.11111111.1 0000000
Wildcard:   0.0.0.127           00000000.00000000.00000000.0 1111111
=>
Network:    192.168.110.128/25  11000000.10101000.01101110.1 0000000
HostMin:    192.168.110.129     11000000.10101000.01101110.1 0000001
HostMax:    192.168.110.254     11000000.10101000.01101110.1 1111110
Broadcast:  192.168.110.255     11000000.10101000.01101110.1 1111111
Hosts/Net:  126                  Class C, Private Internet

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

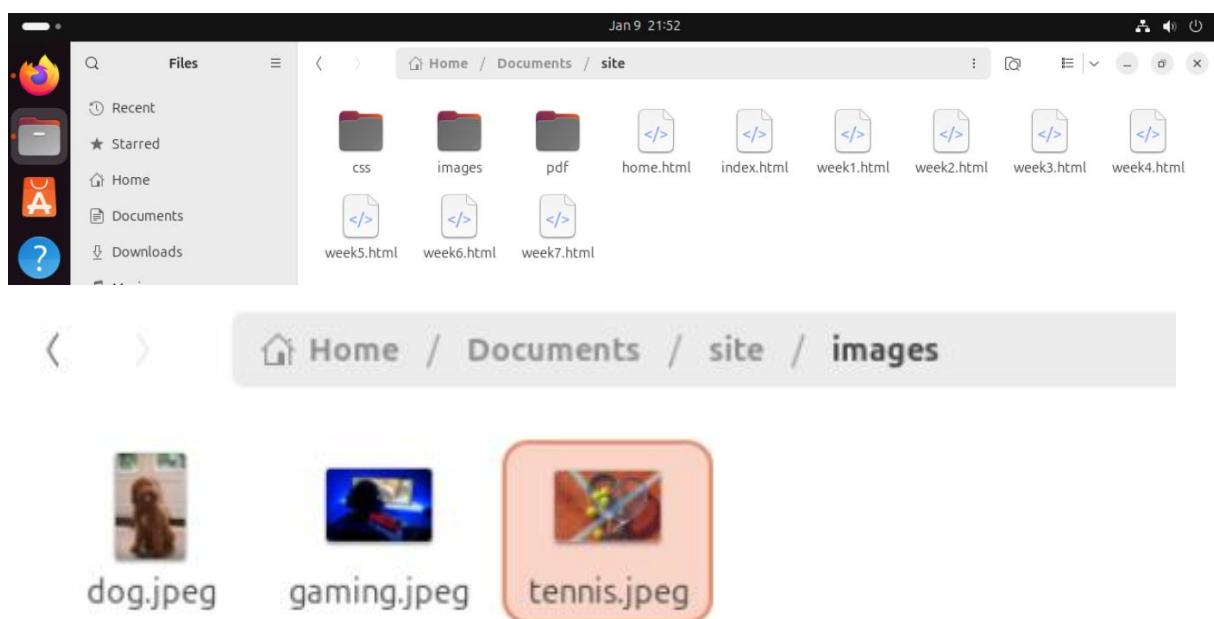
Explain the above calculation in your own words.

### Assignment 6.4: HTML

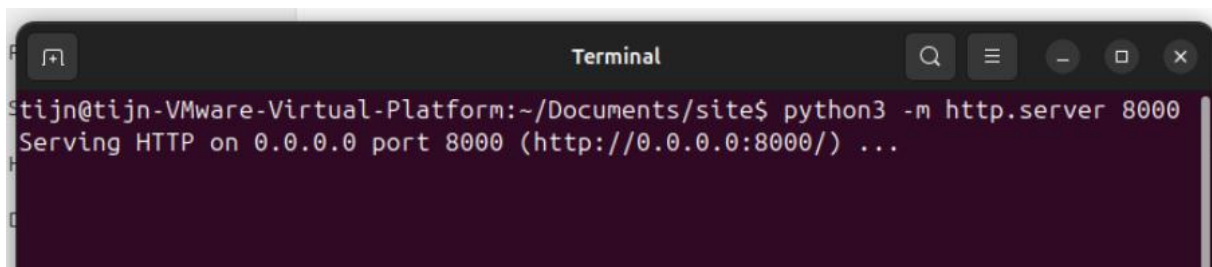
Screenshot IP address Ubuntu VM:

```
Terminal
tijn@tijn-VMware-Virtual-Platform:~/Documents/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: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:51:82:ba brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.139.131/24 brd 192.168.139.255 scope global dynamic noprefixroute ens33
        valid_lft 1737sec preferred_lft 1737sec
    inet6 fe80::20c:29ff:fe51:82ba/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
tijn@tijn-VMware-Virtual-Platform:~/Documents/site$
```

Screenshot of Site directory contents:

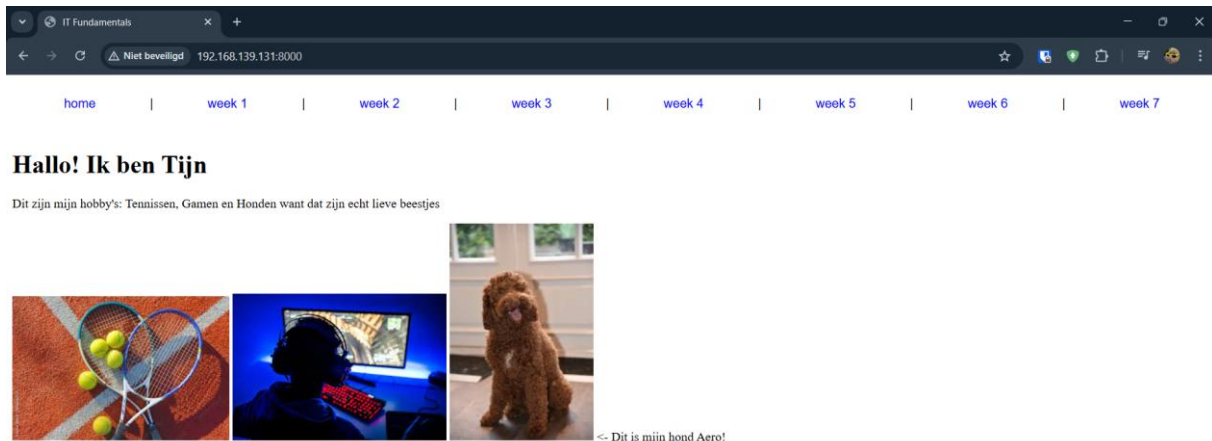


Screenshot python3 webserver command:



```
stijn@tijn-VMware-Virtual-Platform:~/Documents/site$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

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 java.util.Scanner;

public class Main {
    public static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        while (true) {
            printMainMenu();

            int input = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (input) {
                case 1:
                    printOddNumber();
                    break;
                case 2:
                    printPowerOf2();
                    break;
                case 3:
                    printTwosComplement();
                    break;
                case 4:
                    calculateNetworkSegment();
                    break;
                case 0:
                    return;
            }
        }
    }

    public static void printMainMenu() {
        System.out.println("Choose one of these options:");
        System.out.println("1. Is number odd?");
        System.out.println("2. Is number a power of 2?");
        System.out.println("3. Two's complement of number?");
        System.out.println("4. Calculate network segment");
        System.out.println("0. Exit program");
    }

    public static void printOddNumber() {
        System.out.println("Fill in the number you want to check: ");

        int number = scanner.nextInt();

        if ((number & 1) == 1) {
            System.out.println(number + " is odd");
        } else {

```



```

        System.out.println(number + " is even");
    }
}

public static void printPowerOf2() {
    System.out.println("Fill in the number you want to check: ");

    int number = scanner.nextInt();

    if ((number & (number - 1)) == 0) {
        System.out.println(number + " is a power of 2");
    } else {
        System.out.println(number + " isn't a power of 2");
    }
}

public static void printTwosComplement() {
    System.out.println("Fill in the number you want to convert: ");

    int number = scanner.nextInt();

    int twosComplement = ~number + 1;

    System.out.println("Two's complement of " + number + " is " + twosComplement);
}

public static void calculateNetworkSegment() {
    System.out.println("Enter IP address (e.g., 192.168.1.100): ");
    String ipAddress = scanner.nextLine();

    System.out.println("Enter subnet mask (e.g., 255.255.255.224): ");
    String subnetMask = scanner.nextLine();

    // Convert IP address string to array of integers
    String[] ipParts = ipAddress.split("\\.");
    int[] ip = new int[4];
    for (int i = 0; i < 4; i++) {
        ip[i] = Integer.parseInt(ipParts[i]);
    }

    // Convert subnet mask string to array of integers
    String[] maskParts = subnetMask.split("\\.");
    int[] mask = new int[4];
    for (int i = 0; i < 4; i++) {
        mask[i] = Integer.parseInt(maskParts[i]);
    }

    // Calculate network address using bitwise AND

```

```

int[] networkAddress = new int[4];
for (int i = 0; i < 4; i++) {
    networkAddress[i] = ip[i] & mask[i];
}

// Print results in binary format
System.out.println("\nCalculation:");
System.out.println("IP Address: " + toBinaryString(ip));
System.out.println("Subnet Mask: " + toBinaryString(mask));
System.out.println("-----");
System.out.println("Network Addr: " + toBinaryString(networkAddress));

// Print network address in decimal
System.out.println("\nNetwork Address: " + networkAddress[0] + "." +
    networkAddress[1] + "." + networkAddress[2] + "." +
    networkAddress[3]);

// Calculate CIDR notation (count 1's in subnet mask)
int cidr = countOnes(mask);
System.out.println("CIDR Notation: /" + cidr);

// Calculate number of addresses in subnet
int hostBits = 32 - cidr;
int totalAddresses = (int) Math.pow(2, hostBits);
System.out.println("Total addresses in subnet: " + totalAddresses);

// Calculate broadcast address
int[] broadcastAddress = new int[4];
for (int i = 0; i < 4; i++) {
    broadcastAddress[i] = networkAddress[i] | (~mask[i] & 0xFF);
}

System.out.println("Broadcast Address: " + broadcastAddress[0] + "." +
    broadcastAddress[1] + "." + broadcastAddress[2] + "." +
    broadcastAddress[3]);

// Calculate usable range
int[] firstUsable = networkAddress.clone();
firstUsable[3] += 1;

int[] lastUsable = broadcastAddress.clone();
lastUsable[3] -= 1;

System.out.println("Usable IP Range: " + firstUsable[0] + "." +
    firstUsable[1] + "." + firstUsable[2] + "." +
    firstUsable[3] + " to " + lastUsable[0] + "." +
    lastUsable[1] + "." + lastUsable[2] + "." +
    lastUsable[3]);

```

```

        System.out.println();
    }

    // Convert IP address octets to binary string format
    public static String toBinaryString(int[] octets) {
        StringBuilder result = new StringBuilder();
        for (int i = 0; i < octets.length; i++) {
            String binary = String.format("%8s", Integer.toBinaryString(octets[i]))
                .replace(' ', '0');
            result.append(binary);
            if (i < octets.length - 1) {
                result.append(".");
            }
        }
        return result.toString();
    }

    // Count number of 1's in subnet mask (for CIDR notation)
    public static int countOnes(int[] mask) {
        int count = 0;
        for (int octet : mask) {
            count += Integer.bitCount(octet);
        }
        return count;
    }
}

```

The screenshot shows an IDE window titled "Bitwise Calculations" with a "Run" tab active. The console output displays the following:

```

C:\Users\tijn\jdk\ms-21.0.8\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=65212" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8
Choose one of these options:
1. Is number odd?
2. Is number a power of 2?
3. Two's complement of number?
4. Calculate network segment
0. Exit program
4
Enter IP address (e.g., 192.168.1.100):
192.168.1.100
Enter subnet mask (e.g., 255.255.255.224):
255.255.255.224

Calculation:
IP Address:  11000000.10101000.00000001.01100100
Subnet Mask: 11111111.11111111.11111111.11100000
-----
Network Addr: 11000000.10101000.00000001.01100000

Network Address: 192.168.1.96
CIDR Notation: /27
Total addresses in subnet: 32
Broadcast Address: 192.168.1.127
Usable IP Range: 192.168.1.97 to 192.168.1.126

```

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