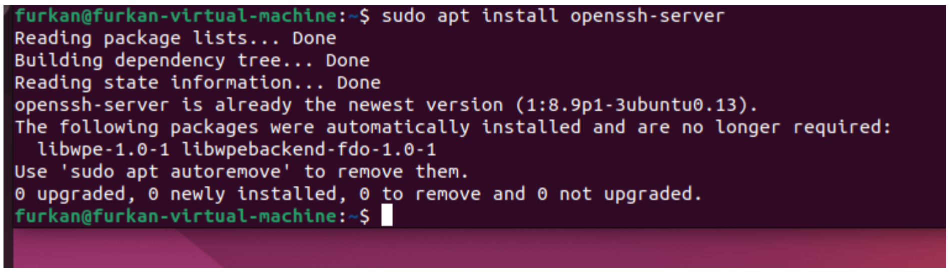
Template Week 6 – Networking

Student number: 593250 Furkan Yildirim

Assignment 6.1: Working from home

Screenshot installation openssh-server:

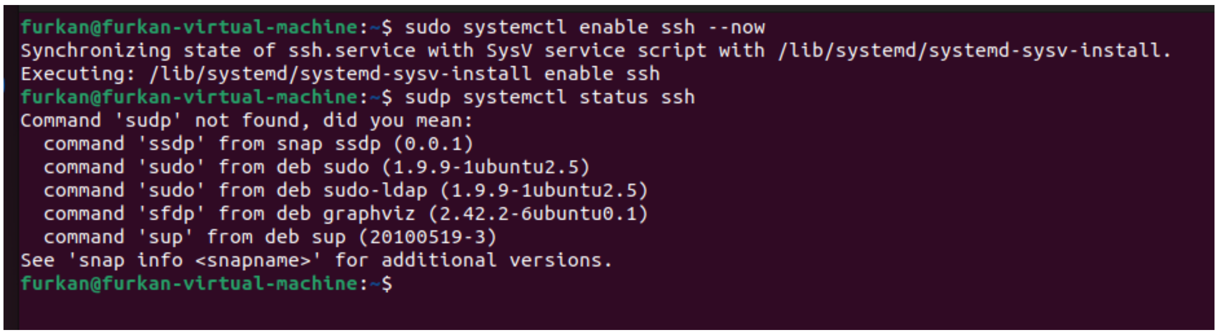
sudo apt install openssh-server



```
furkan@furkan-virtual-machine:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.13).
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
furkan@furkan-virtual-machine:~$ 
```

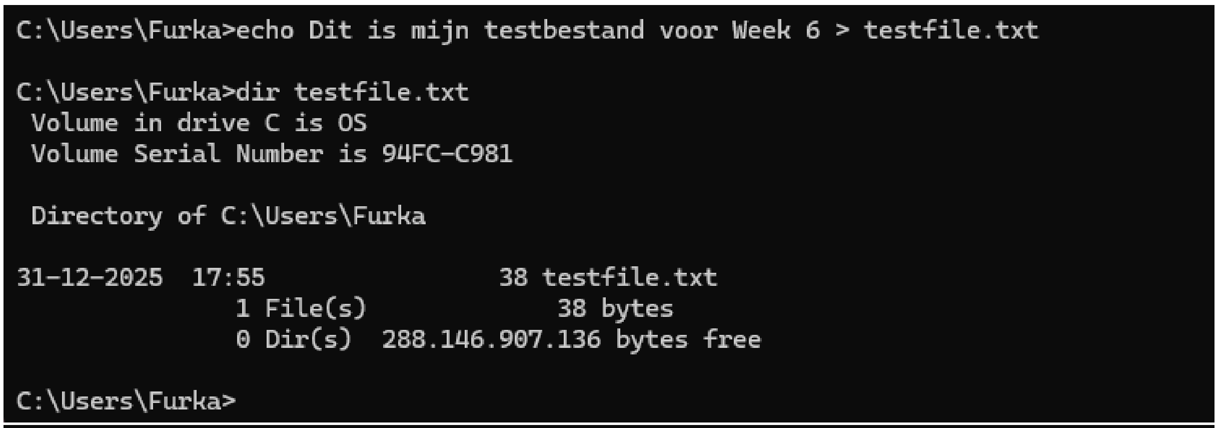
Screenshot successful SSH command execution:

sudo systemctl enable ssh –now
sudo systemctl status ssh

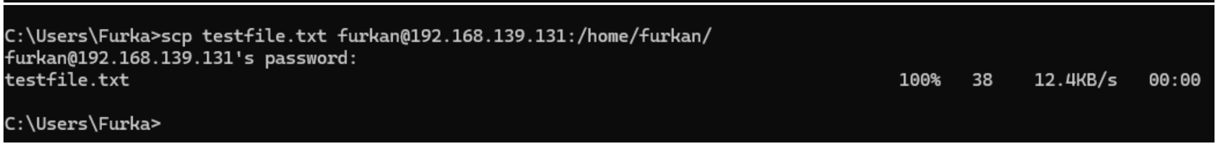


```
furkan@furkan-virtual-machine:~$ sudo systemctl enable ssh --now
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
furkan@furkan-virtual-machine:~$ sudp systemctl status ssh
Command 'sudp' not found, did you mean:
  command 'ssdp' from snap ssdp (0.0.1)
  command 'sudo' from deb sudo (1.9.9-1ubuntu2.5)
  command 'sudo' from deb sudo-ldap (1.9.9-1ubuntu2.5)
  command 'sfdp' from deb graphviz (2.42.2-6ubuntu0.1)
  command 'sup' from deb sup (20100519-3)
See 'snap info <snapname>' for additional versions.
furkan@furkan-virtual-machine:~$
```

Screenshot successful execution SCP command:



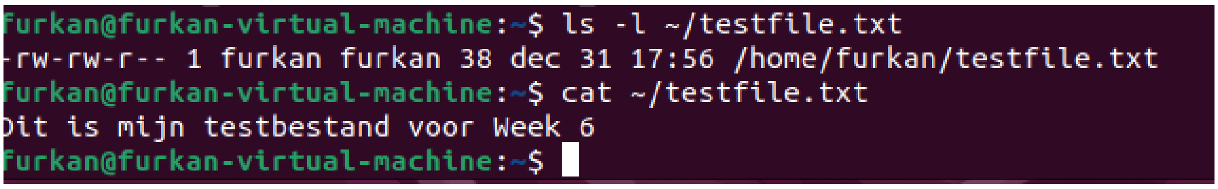
```
C:\Users\Furka>echo Dit is mijn testbestand voor Week 6 > testfile.txt

C:\Users\Furka>dir testfile.txt
 Volume in drive C is OS
 Volume Serial Number is 94FC-C981

 Directory of C:\Users\Furka

31-12-2025  17:55                38 testfile.txt
               1 File(s)             38 bytes
               0 Dir(s)  288.146.907.136 bytes free

C:\Users\Furka>
```



```
C:\Users\Furka>scp testfile.txt furkan@192.168.139.131:/home/furkan/
furkan@192.168.139.131's password:
testfile.txt                                                                          100%   38    12.4KB/s   00:00

C:\Users\Furka>
```

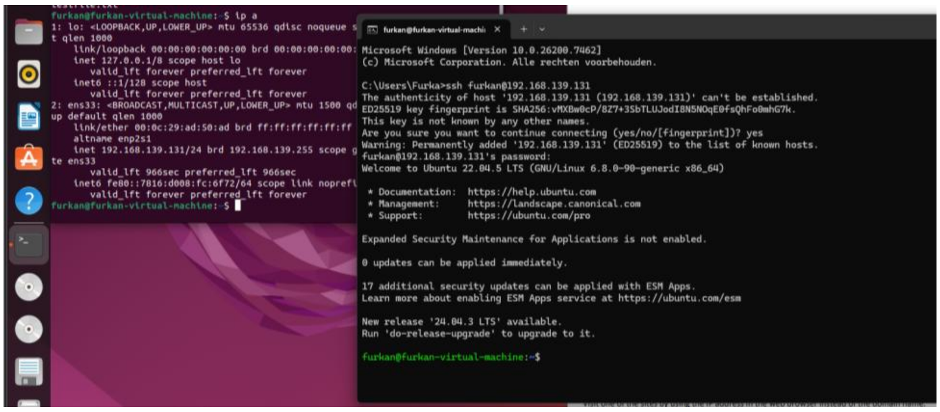
Hierboven met de commando **scp testfile.txt furkan@192.168.139.131:/home/furkan/**

Kopieert bestand van Windows naar Ubuntu.



```
furkan@furkan-virtual-machine:~$ ls -l ~/testfile.txt
-rw-rw-r-- 1 furkan furkan 38 dec 31 17:56 /home/furkan/testfile.txt
furkan@furkan-virtual-machine:~$ cat ~/testfile.txt
Dit is mijn testbestand voor Week 6
furkan@furkan-virtual-machine:~$
```

de inhoud van de tekstbestand.

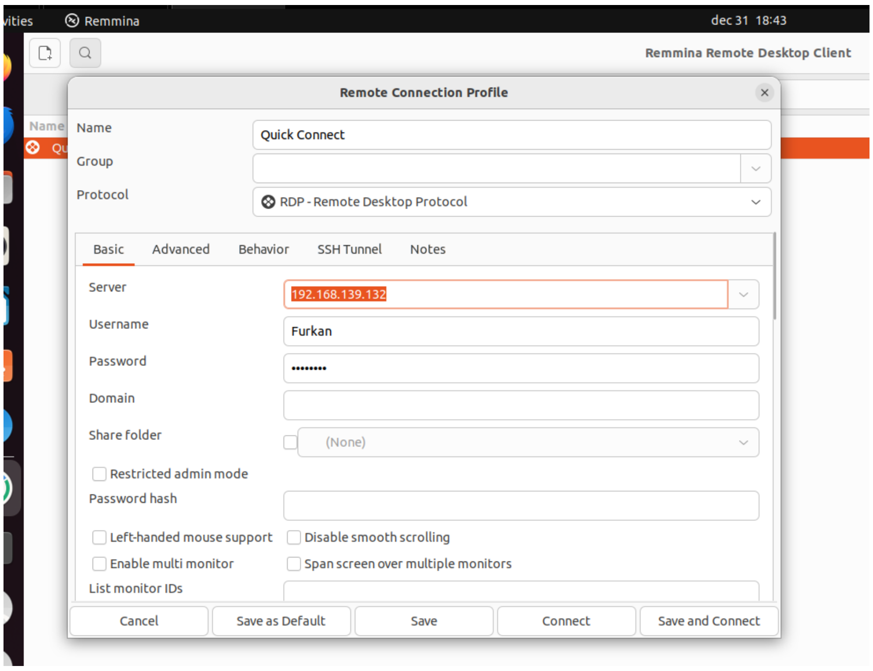


```
furkan@furkan-virtual-machine:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue s
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qd
up default qlen 1000
    link/ether 00:0c:29:ad:50:ad brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.139.131/24 brd 192.168.139.255 scope g
te ens33
       valid_lft 966sec preferred_lft 966sec
    inet6 fe80::7816:d008:fc:6f72/64 scope link nopref
       valid_lft forever preferred_lft forever
furkan@furkan-virtual-machine:~$
```

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

C:\Users\Furka>ssh furkan@192.168.139.131
The authenticity of host '192.168.139.131 (192.168.139.131)' can't be established.
ED25519 key fingerprint is SHA256:vMXBw0cP/8Z7+3SbTLUJodI8N5NOqE0fsQhFo0mhG7k.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.139.131' (ED25519) to the list of known hosts.
furkan@192.168.139.131's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-90-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.

0 updates can be applied immediately.

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

New release '24.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

furkan@furkan-virtual-machine:~$
```

1. Show that you can log in to the Ubuntu VM via the command prompt via the command ssh

Screenshot remmina:



Remmina — dec 31 18:43 — Remmina Remote Desktop Client

Remote Connection Profile

Name	Quick Connect
Group	
Protocol	RDP - Remote Desktop Protocol

Basic | Advanced | Behavior | SSH Tunnel | Notes

Server	192.168.139.132
Username	Furkan
Password	••••••••
Domain	
Share folder	(None)

☐ Restricted admin mode

Password hash

☐ Left-handed mouse support ☐ Disable smooth scrolling

☐ Enable multi monitor ☐ Span screen over multiple monitors

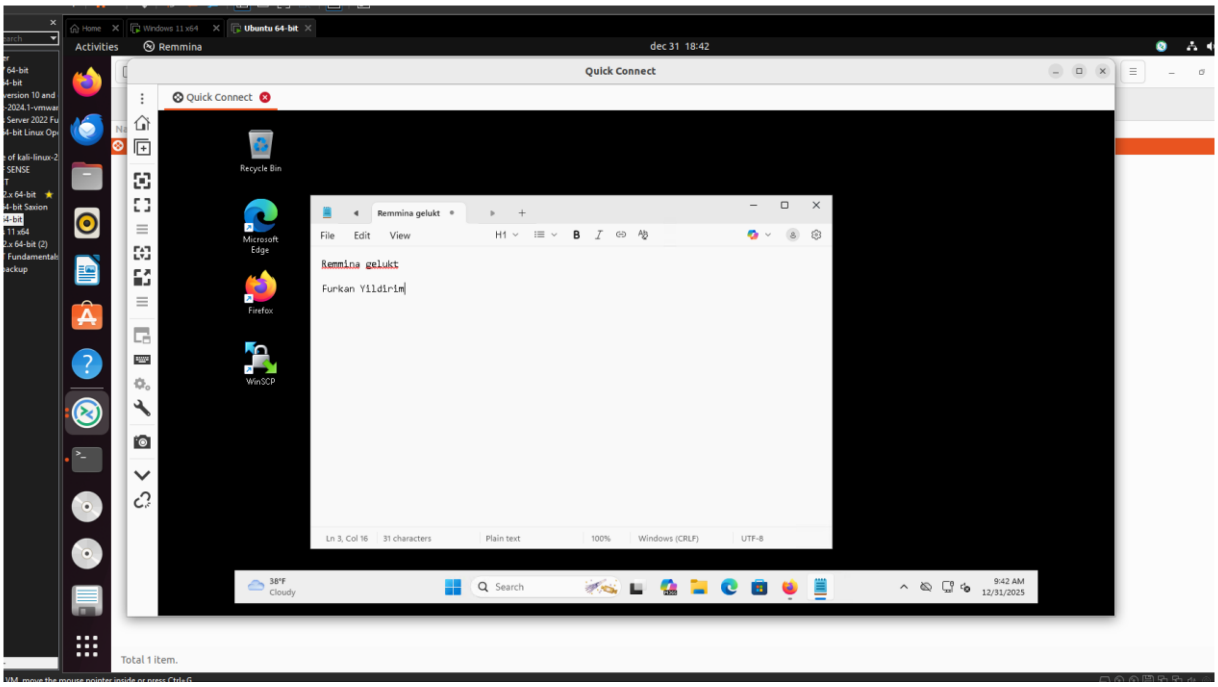
List monitor IDs

Remote Desktop verbinding opgezet met Remmina.

Configuratie: - Server: 192.168.139.132 (Windows VM) –

Protocol: RDP –

Username: Furkan



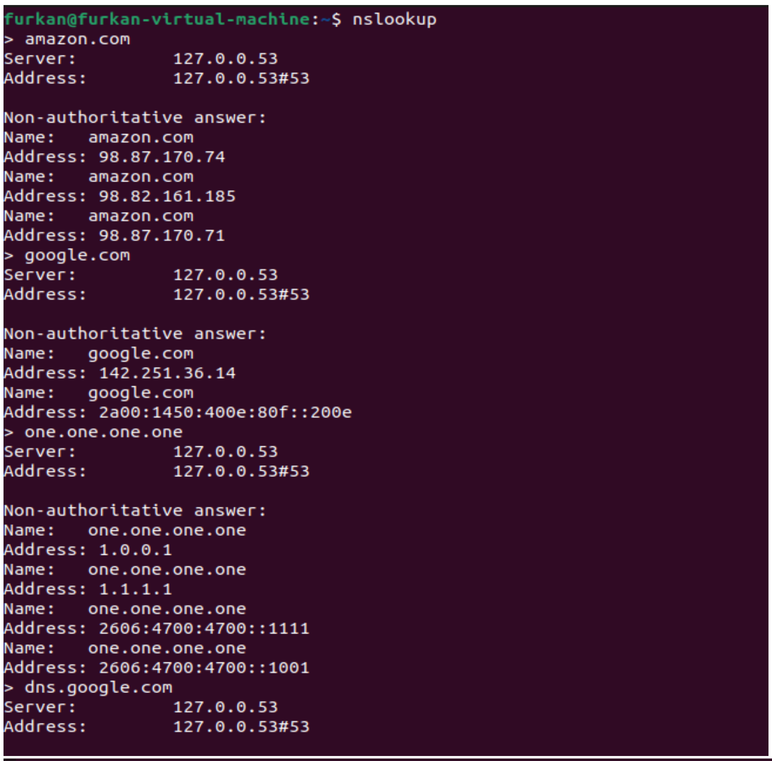
Remmina gelukt

Furkan Yildirim

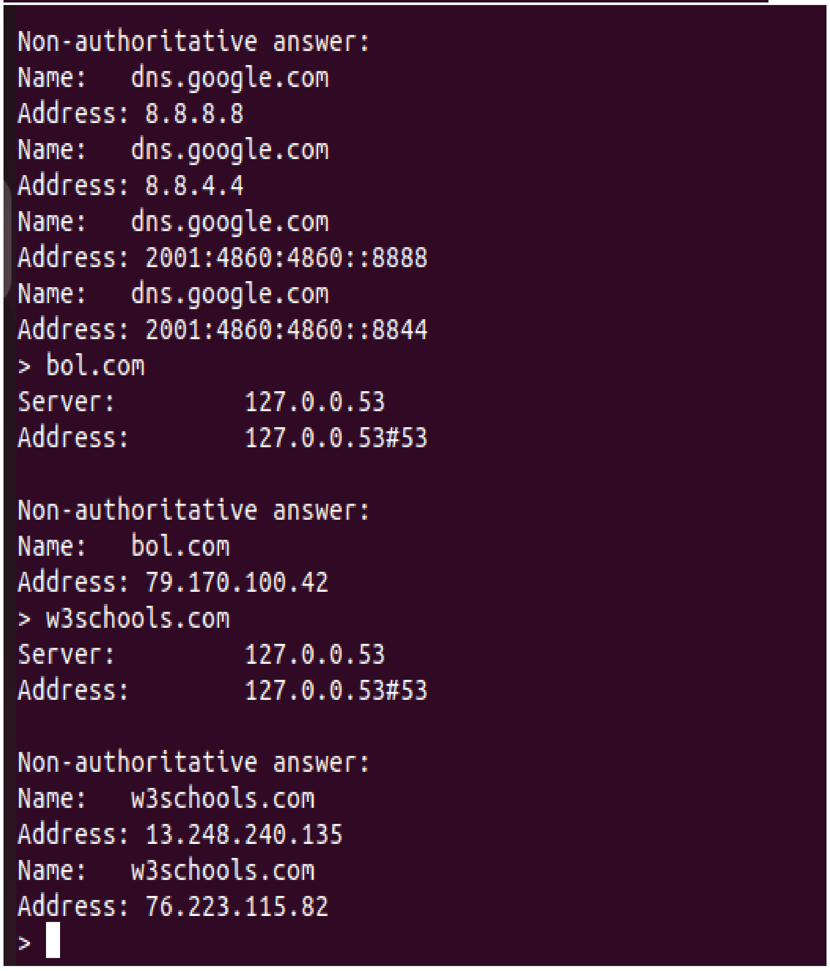
De screenshot toont het Windows bureaublad toegankelijk via Remmina vanuit Ubuntu VM

Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:



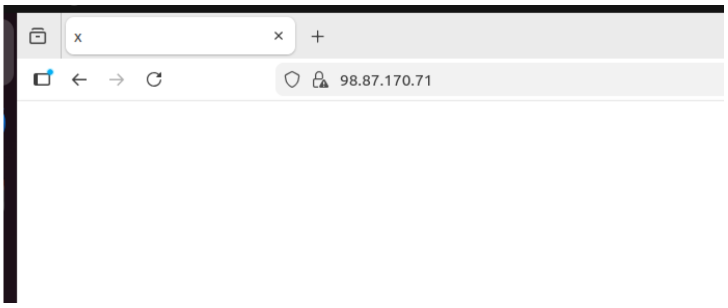
```
furkan@furkan-virtual-machine:~$ nslookup
> amazon.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   amazon.com
Address: 98.87.170.74
Name:   amazon.com
Address: 98.82.161.185
Name:   amazon.com
Address: 98.87.170.71
> google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.251.36.14
Name:   google.com
Address: 2a00:1450:400e:80f::200e
> one.one.one.one
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   one.one.one.one
Address: 1.0.0.1
Name:   one.one.one.one
Address: 1.1.1.1
Name:   one.one.one.one
Address: 2606:4700:4700::1111
Name:   one.one.one.one
Address: 2606:4700:4700::1001
> dns.google.com
Server:         127.0.0.53
Address:        127.0.0.53#53
```



```
Non-authoritative answer:
Name:   dns.google.com
Address: 8.8.8.8
Name:   dns.google.com
Address: 8.8.4.4
Name:   dns.google.com
Address: 2001:4860:4860::8888
Name:   dns.google.com
Address: 2001:4860:4860::8844
> bol.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   bol.com
Address: 79.170.100.42
> w3schools.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   w3schools.com
Address: 13.248.240.135
Name:   w3schools.com
Address: 76.223.115.82
>
```

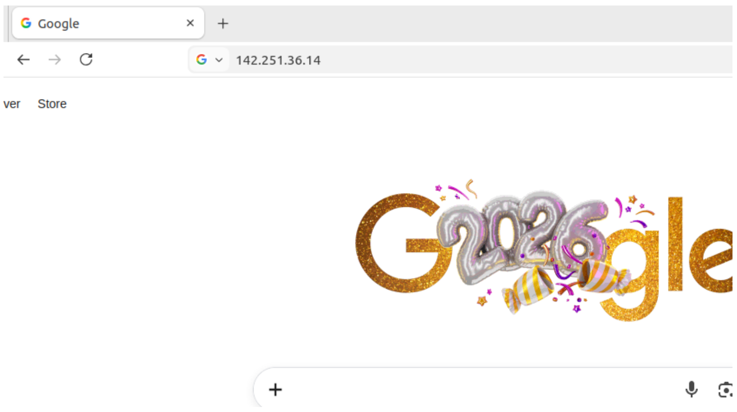
Screenshot website visit via IP address:

Amazon



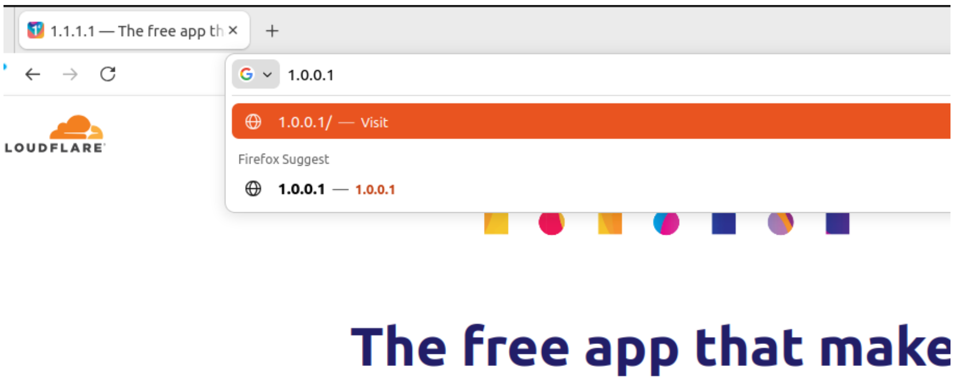
Alle 3 ip address geprobeerd maar geen succes. Geeft de melding dat de website onveilig is.

Google.com



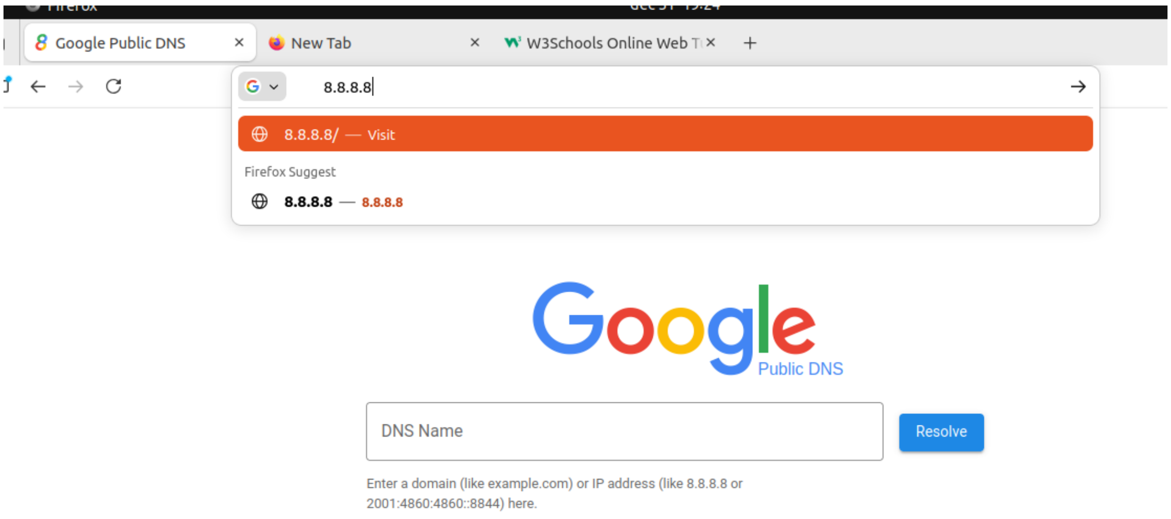
Tested IP: 142.251.36.14 (Google)

One.one.one.one



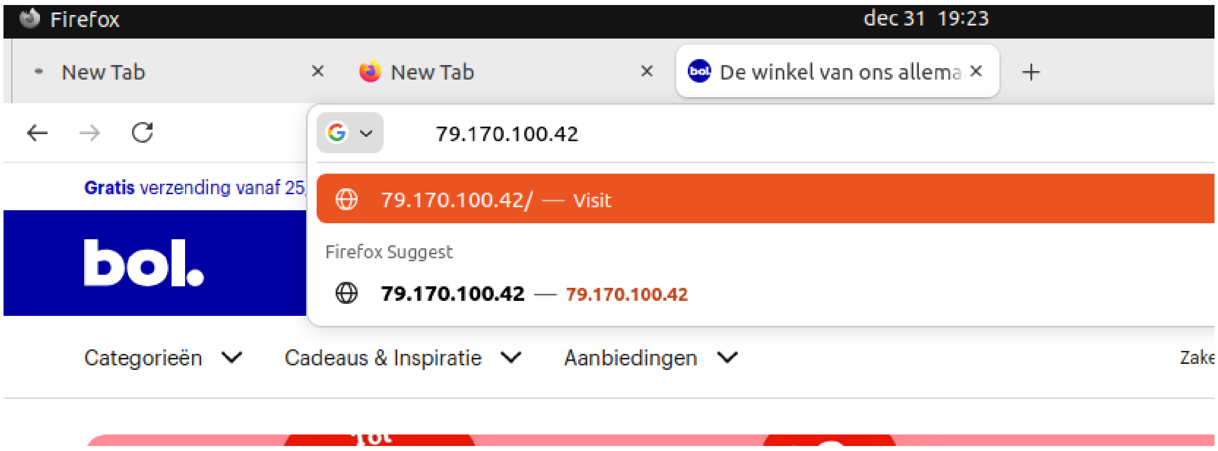
Tested ip:
1.0.0.1, 1.1.1.1

Dns.google.com



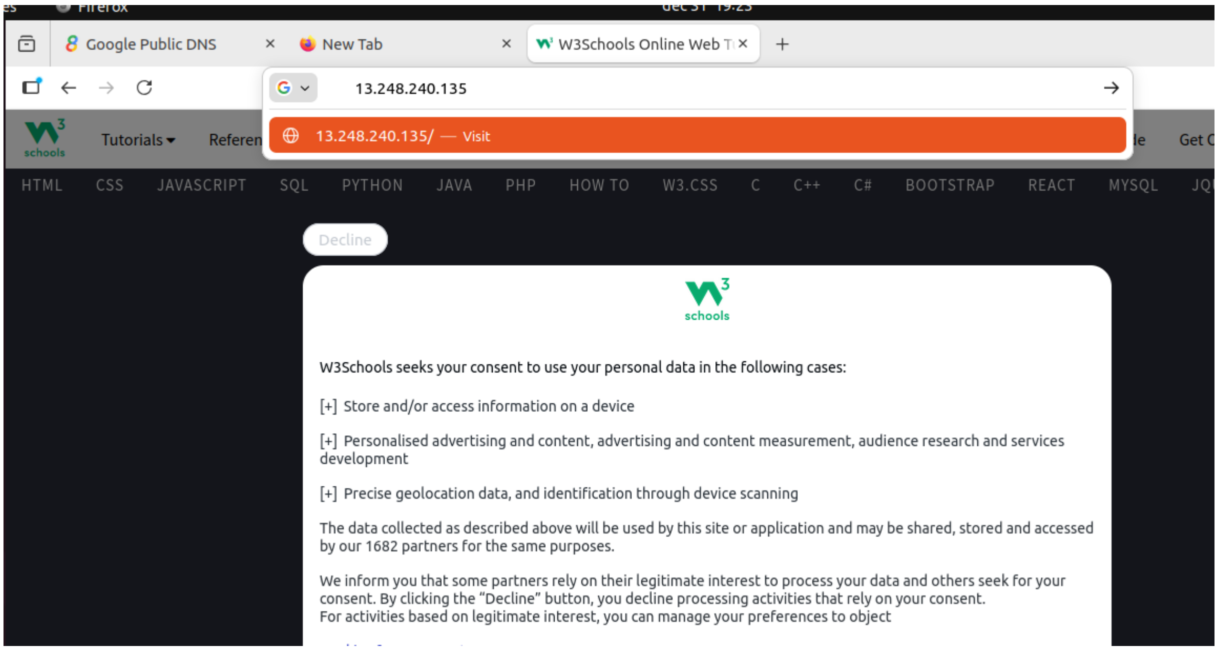
Tested ip: 8.8.8.8, 8.8.4.4

Bol.com



Tested ip:
79.170.100.42

W3schools.com



W3Schools seeks your consent to use your personal data in the following cases:

[+] Store and/or access information on a device

[+] Personalised advertising and content, advertising and content measurement, audience research and services development

[+] Precise geolocation data, and identification through device scanning

The data collected as described above will be used by this site or application and may be shared, stored and accessed by our 1682 partners for the same purposes.

We inform you that some partners rely on their legitimate interest to process your data and others seek for your consent. By clicking the "Decline" button, you decline processing activities that rely on your consent.
For activities based on legitimate interest, you can manage your preferences to object

Tested ip 13.248.240.135,

Assignment 6.3: subnetting

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

/25 = 25 bits voor netwerk, dus 32-25 = 7 bits voor hosts

2^7 = 128 totale IP-adressen

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

Bruikbare IP-range: 192.168.110.129 tot 192.168.110.254 Totaal bruikbare adressen: 126

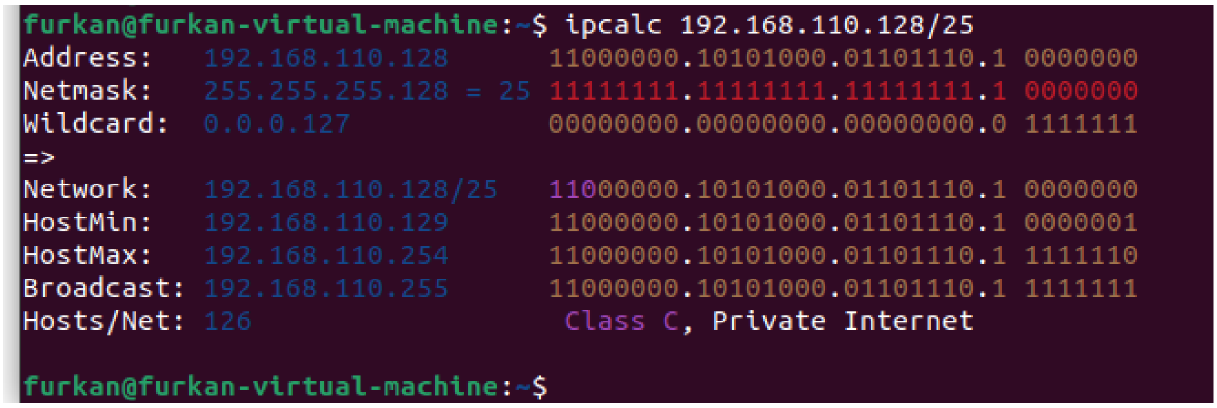
Waarom?

Eerste adres (192.168.110.128) = Network adres (gereserveerd)

Laatste adres (192.168.110.255) = Broadcast adres (gereserveerd)

126 bruikbare adressen voor computers

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



```
furkan@furkan-virtual-machine:~$ 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

furkan@furkan-virtual-machine:~$
```

Explain the above calculation in your own words.

Het /25 getal betekent dat het eerste deel van het IP-adres vastligt (192.168.110), en alleen het laatste stukje kan veranderen.

Met /25 heb je 7 "vrije" plekken die kunnen veranderen.

Dat geeft 2×2×2×2×2×2×2 = 128 mogelijkheden.

Van die 128 adressen:

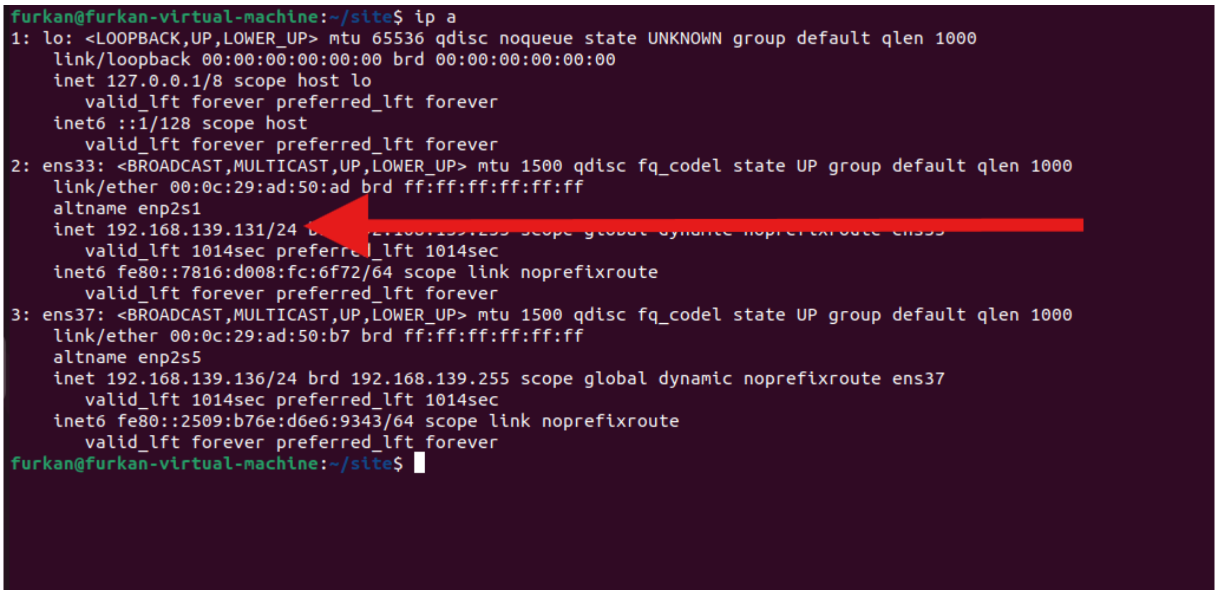
- 192.168.110.128 = Het netwerk zelf (mag niet gebruiken)
- 192.168.110.129 tot 254 = Voor computers (mag wel gebruiken!)
- 192.168.110.255 = Voor berichten naar iedereen (mag niet gebruiken)

Dus: 128 totaal - 2 gereserveerd = 126 bruikbare adressen.

Dit subnetting helpt om grote netwerken op te delen in kleinere stukjes, zodat het overzichtelijker en veiliger is.

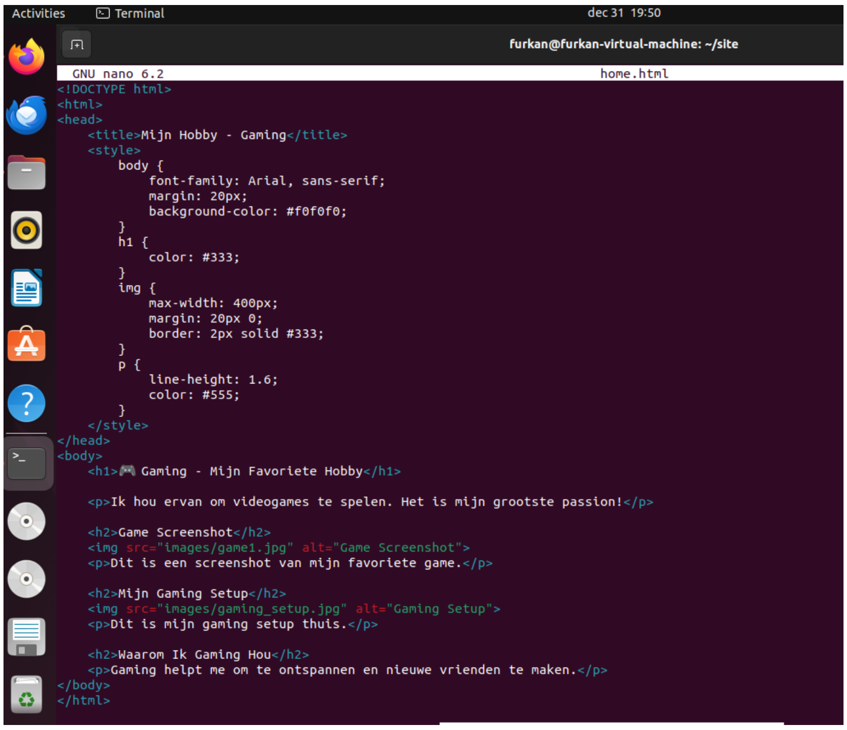
Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:



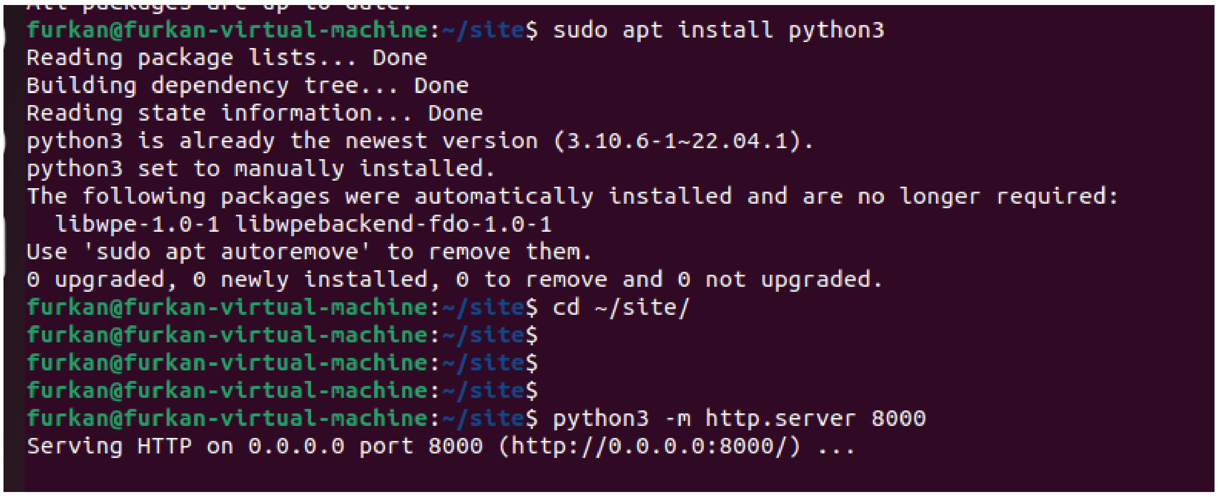
```
furkan@furkan-virtual-machine:~/site$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       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:ad:50:ad 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 1014sec preferred_lft 1014sec
    inet6 fe80::7816:d008:fc:6f72/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: ens37: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:ad:50:b7 brd ff:ff:ff:ff:ff:ff
    altname enp2s5
    inet 192.168.139.136/24 brd 192.168.139.255 scope global dynamic noprefixroute ens37
       valid_lft 1014sec preferred_lft 1014sec
    inet6 fe80::2509:b76e:d6e6:9343/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
furkan@furkan-virtual-machine:~/site$ 
```

Screenshot of Site directory contents:



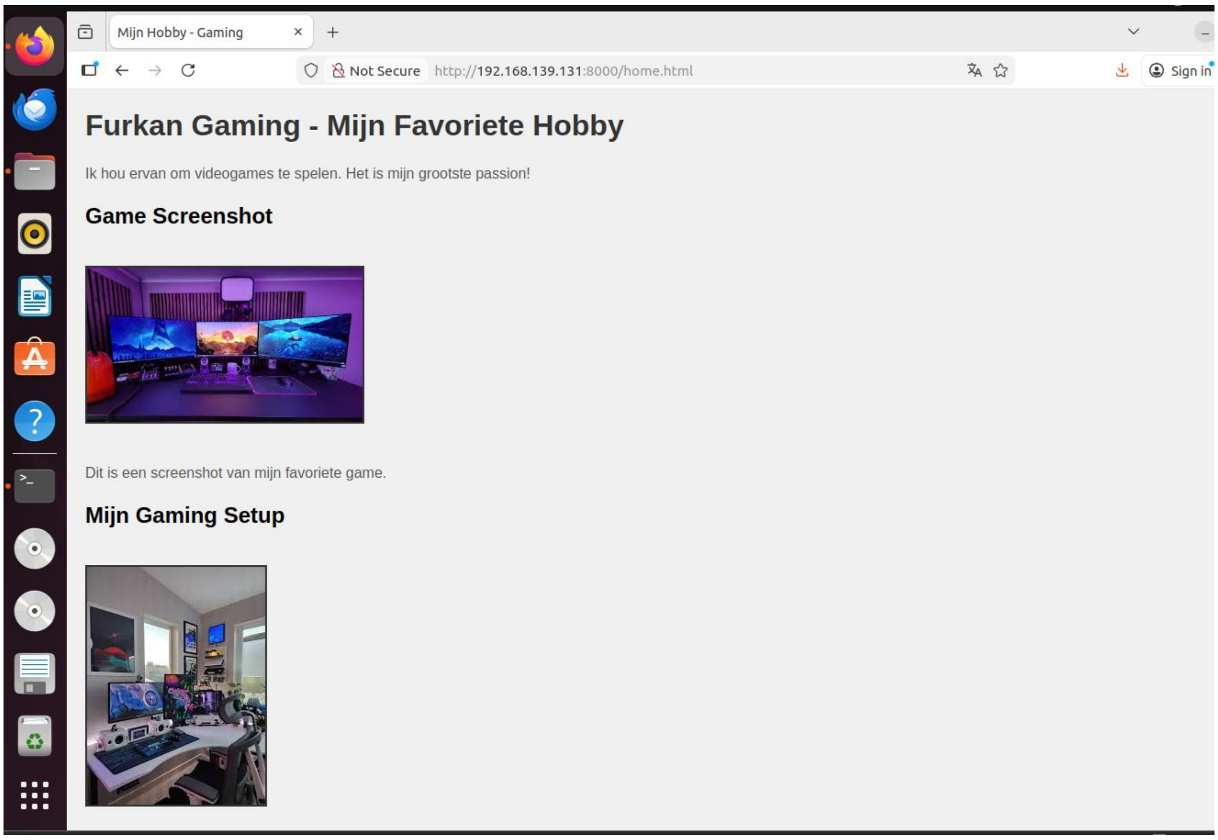
```
<!DOCTYPE html>
<html>
<head>
    <title>Mijn Hobby - Gaming</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
            background-color: #f0f0f0;
        }
        h1 {
            color: #333;
        }
        img {
            max-width: 400px;
            margin: 20px 0;
            border: 2px solid #333;
        }
        p {
            line-height: 1.6;
            color: #555;
        }
    </style>
</head>
<body>
    <h1>🎮 Gaming - Mijn Favoriete Hobby</h1>

    <p>Ik hou ervan om videogames te spelen. Het is mijn grootste passion!</p>

    <h2>Game Screenshot</h2>
    <img src="images/game1.jpg" alt="Game Screenshot">
    <p>Dit is een screenshot van mijn favoriete game.</p>

    <h2>Mijn Gaming Setup</h2>
    <img src="images/gaming_setup.jpg" alt="Gaming Setup">
    <p>Dit is mijn gaming setup thuis.</p>

    <h2>Waarom Ik Gaming Hou</h2>
    <p>Gaming helpt me om te ontspannen en nieuwe vrienden te maken.</p>
</body>
</html>
```

Screenshot python3 webserver command:



```
furkan@furkan-virtual-machine:~/site$ sudo apt install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04.1).
python3 set to manually installed.
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
furkan@furkan-virtual-machine:~/site$ cd ~/site/
furkan@furkan-virtual-machine:~/site$
furkan@furkan-virtual-machine:~/site$
furkan@furkan-virtual-machine:~/site$
furkan@furkan-virtual-machine:~/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



Furkan Gaming - Mijn Favoriete Hobby

Ik hou ervan om videogames te spelen. Het is mijn grootste passion!

Game Screenshot

Dit is een screenshot van mijn favoriete game.

Mijn Gaming Setup

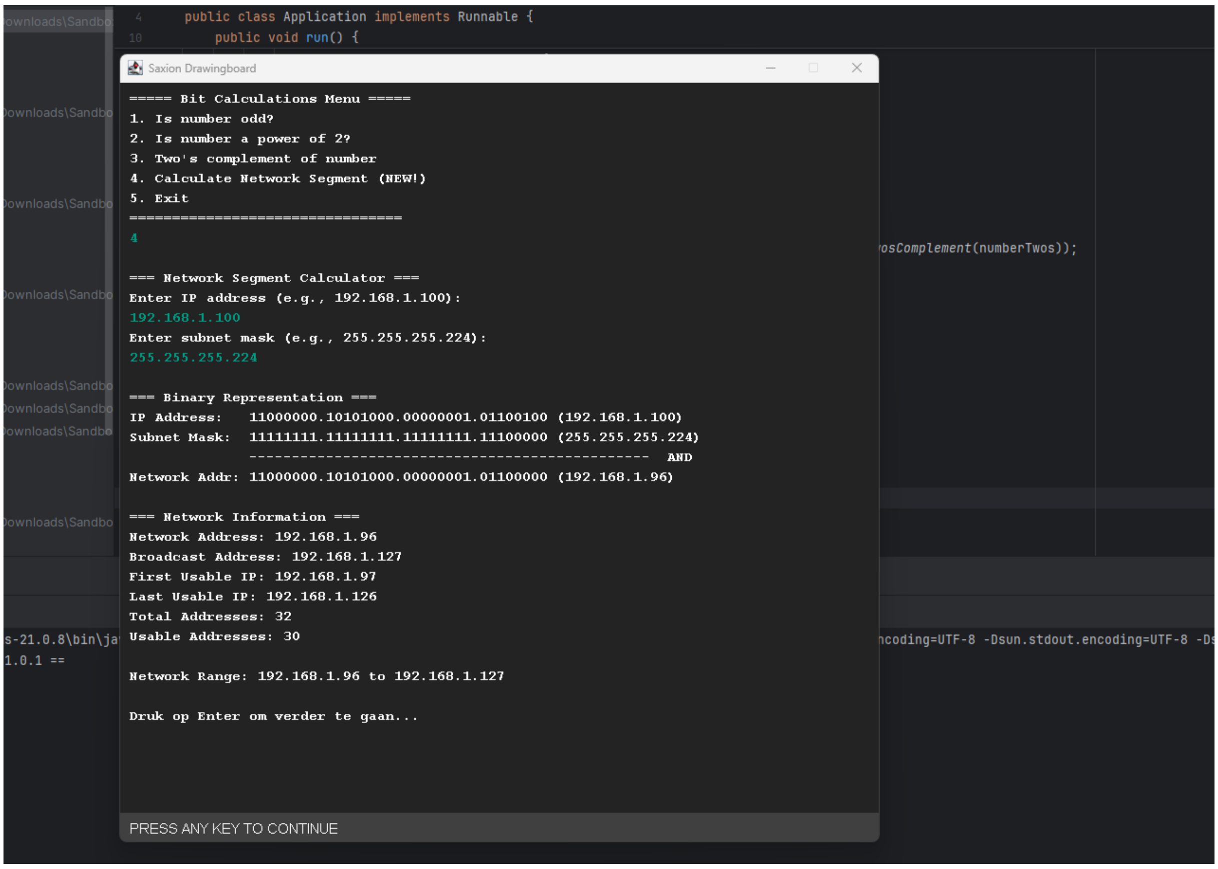
Assignment 6.5: Network segment

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 (2⁵).
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.



```
public class Application implements Runnable {
    public void run() {
```

```
===== Bit Calculations Menu =====
1. Is number odd?
2. Is number a power of 2?
3. Two's complement of number
4. Calculate Network Segment (NEW!)
5. Exit
=================================
4

=== Network Segment Calculator ===
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

=== Binary Representation ===
IP Address:   11000000.10101000.00000001.01100100 (192.168.1.100)
Subnet Mask:  11111111.11111111.11111111.11100000 (255.255.255.224)
              -----------------------------------------------  AND
Network Addr: 11000000.10101000.00000001.01100000 (192.168.1.96)

=== Network Information ===
Network Address: 192.168.1.96
Broadcast Address: 192.168.1.127
First Usable IP: 192.168.1.97
Last Usable IP: 192.168.1.126
Total Addresses: 32
Usable Addresses: 30

Network Range: 192.168.1.96 to 192.168.1.127

Druk op Enter om verder te gaan...
```

```
import nl.saxion.app.SaxionApp;
import java.awt.*;

public class Application implements Runnable {

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

  public void run() {
    int choice;

    do {
      SaxionApp.clear(); // scherm leegmaken

      // Menu tonen
      SaxionApp.printLine("===== Bit Calculations Menu =====");
      SaxionApp.printLine("1. Is number odd?");
      SaxionApp.printLine("2. Is number a power of 2?");
      SaxionApp.printLine("3. Two's complement of number");
      SaxionApp.printLine("4. Calculate Network Segment (NEW!)");
      SaxionApp.printLine("5. Exit");
      SaxionApp.printLine("================================");

      // Keuze van gebruiker
      choice = SaxionApp.readInt("Enter your choice (1-5): ");

      if (choice == 1) {
        SaxionApp.printLine("Voer een getal/cijfer in:");
        int numberOdd = SaxionApp.readInt("");
        if (isOdd(numberOdd)) {
          SaxionApp.printLine(numberOdd + " is odd.");
        } else {
          SaxionApp.printLine(numberOdd + " is even.");
        }
      }
      else if (choice == 2) {
        SaxionApp.printLine("Voer een getal/cijfer in:");
        int numberPower = SaxionApp.readInt("");
        if (isPowerOfTwo(numberPower)) {
          SaxionApp.printLine(numberPower + " is a power of 2.");
        } else {
          SaxionApp.printLine(numberPower + " is NOT a power of 2.");
        }
      }
      else if (choice == 3) {
        SaxionApp.printLine("Voer een getal/cijfer in:");
        int numberTwos = SaxionApp.readInt("");
```

```
            SaxionApp.printLine("Two's complement van " + numberTwos + " is: " +
twosComplement(numberTwos));
        }
        else if (choice == 4) {
            // NEW: Network Segment Calculation
            SaxionApp.printLine("\n=== Network Segment Calculator ===");
            SaxionApp.printLine("Enter IP address (e.g., 192.168.1.100):");
            String ipAddress = SaxionApp.readString();

            SaxionApp.printLine("Enter subnet mask (e.g., 255.255.255.224):");
            String subnetMask = SaxionApp.readString();

            calculateNetworkSegment(ipAddress, subnetMask);
        }
        else if (choice == 5) {
            SaxionApp.printLine("Exiting program...");
        }
        else {
            SaxionApp.printLine("Ongeldige keuze. Kies 1-5.");
        }

        if (choice != 5) {
            SaxionApp.printLine("\nDruk op Enter om verder te gaan...");
            SaxionApp.pause(); // wacht op Enter
        }

    } while (choice != 5);
}

// Controleer of een getal oneven is
public static boolean isOdd(int num) {
    return (num & 1) == 1;
}

// Controleer of een getal een macht van 2 is
public static boolean isPowerOfTwo(int num) {
    return num > 0 && (num & (num - 1)) == 0;
}

// Bereken twee's complement
public static int twosComplement(int num) {
    return ~num + 1;
}

// NEW: Calculate Network Segment using bitwise AND operator
public static void calculateNetworkSegment(String ipAddress, String subnetMask) {
    // Split IP and subnet into octets
    String[] ipOctets = ipAddress.split("\\.");
```

```
String[] maskOctets = subnetMask.split("\\.");

// Validate input
if (ipOctets.length != 4 || maskOctets.length != 4) {
   SaxionApp.printLine("Error: Invalid IP or subnet mask format!");
   return;
}

SaxionApp.printLine("\n=== Binary Representation ===");

// Convert IP to binary
SaxionApp.print("IP Address:   ");
int[] ipBinary = new int[4];
for (int i = 0; i < 4; i++) {
   ipBinary[i] = Integer.parseInt(ipOctets[i]);
   SaxionApp.print(toBinaryString(ipBinary[i]));
   if (i < 3) SaxionApp.print(".");
}
SaxionApp.printLine(" (" + ipAddress + ")");

// Convert subnet to binary
SaxionApp.print("Subnet Mask:  ");
int[] maskBinary = new int[4];
for (int i = 0; i < 4; i++) {
   maskBinary[i] = Integer.parseInt(maskOctets[i]);
   SaxionApp.print(toBinaryString(maskBinary[i]));
   if (i < 3) SaxionApp.print(".");
}
SaxionApp.printLine(" (" + subnetMask + ")");

// Bitwise AND Operation
SaxionApp.printLine("            -----------------------------------------------  AND");
SaxionApp.print("Network Addr: ");
int[] networkAddress = new int[4];
for (int i = 0; i < 4; i++) {
   networkAddress[i] = ipBinary[i] & maskBinary[i];  // BITWISE AND
   SaxionApp.print(toBinaryString(networkAddress[i]));
   if (i < 3) SaxionApp.print(".");
}

String networkAddressStr = networkAddress[0] + "." + networkAddress[1] +
        "." + networkAddress[2] + "." + networkAddress[3];
SaxionApp.printLine(" (" + networkAddressStr + ")");

// Calculate broadcast address
SaxionApp.printLine("\n=== Network Information ===");
SaxionApp.printLine("Network Address: " + networkAddressStr);
```

```
    int[] broadcastAddress = new int[4];
    for (int i = 0; i < 4; i++) {
        int invertedMask = (~maskBinary[i]) & 0xFF;
        broadcastAddress[i] = networkAddress[i] | invertedMask;
    }

    String broadcastAddressStr = broadcastAddress[0] + "." + broadcastAddress[1] +
            "." + broadcastAddress[2] + "." + broadcastAddress[3];

    SaxionApp.printLine("Broadcast Address: " + broadcastAddressStr);

    // Calculate total addresses
    int hostBits = countHostBits(subnetMask);
    int totalAddresses = (int) Math.pow(2, hostBits);
    int usableAddresses = totalAddresses - 2;

    String firstUsable = incrementIP(networkAddress);
    String lastUsable = decrementIP(broadcastAddress);

    SaxionApp.printLine("First Usable IP: " + firstUsable);
    SaxionApp.printLine("Last Usable IP: " + lastUsable);
    SaxionApp.printLine("Total Addresses: " + totalAddresses);
    SaxionApp.printLine("Usable Addresses: " + usableAddresses);
    SaxionApp.printLine("\nNetwork Range: " + networkAddressStr + " to " + broadcastAddressStr);
}

// Convert number to 8-bit binary string
public static String toBinaryString(int num) {
    String binary = Integer.toBinaryString(num);
    while (binary.length() < 8) {
        binary = "0" + binary;
    }
    return binary;
}

// Count host bits (zeros at the end of subnet mask)
public static int countHostBits(String subnetMask) {
    String[] octets = subnetMask.split("\\.");
    int hostBits = 0;

    for (int i = 3; i >= 0; i--) {
        int octet = Integer.parseInt(octets[i]);
        String binary = toBinaryString(octet);

        for (int j = 7; j >= 0; j--) {
            if (binary.charAt(j) == '0') {
                hostBits++;
            } else {
                return hostBits;
            }
        }
    }
    return hostBits;
}

// Increment IP address by 1
public static String incrementIP(int[] ip) {
    int[] newIP = ip.clone();

    for (int i = 3; i >= 0; i--) {
        if (newIP[i] < 255) {
            newIP[i]++;
            break;
        } else {
            newIP[i] = 0;
        }
    }

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

// Decrement IP address by 1
public static String decrementIP(int[] ip) {
    int[] newIP = ip.clone();

    for (int i = 3; i >= 0; i--) {
        if (newIP[i] > 0) {
            newIP[i]--;
            break;
        } else {
            newIP[i] = 255;
        }
    }

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

Ready? Save this file and export it as a pdf file with the name: **week6.pdf**