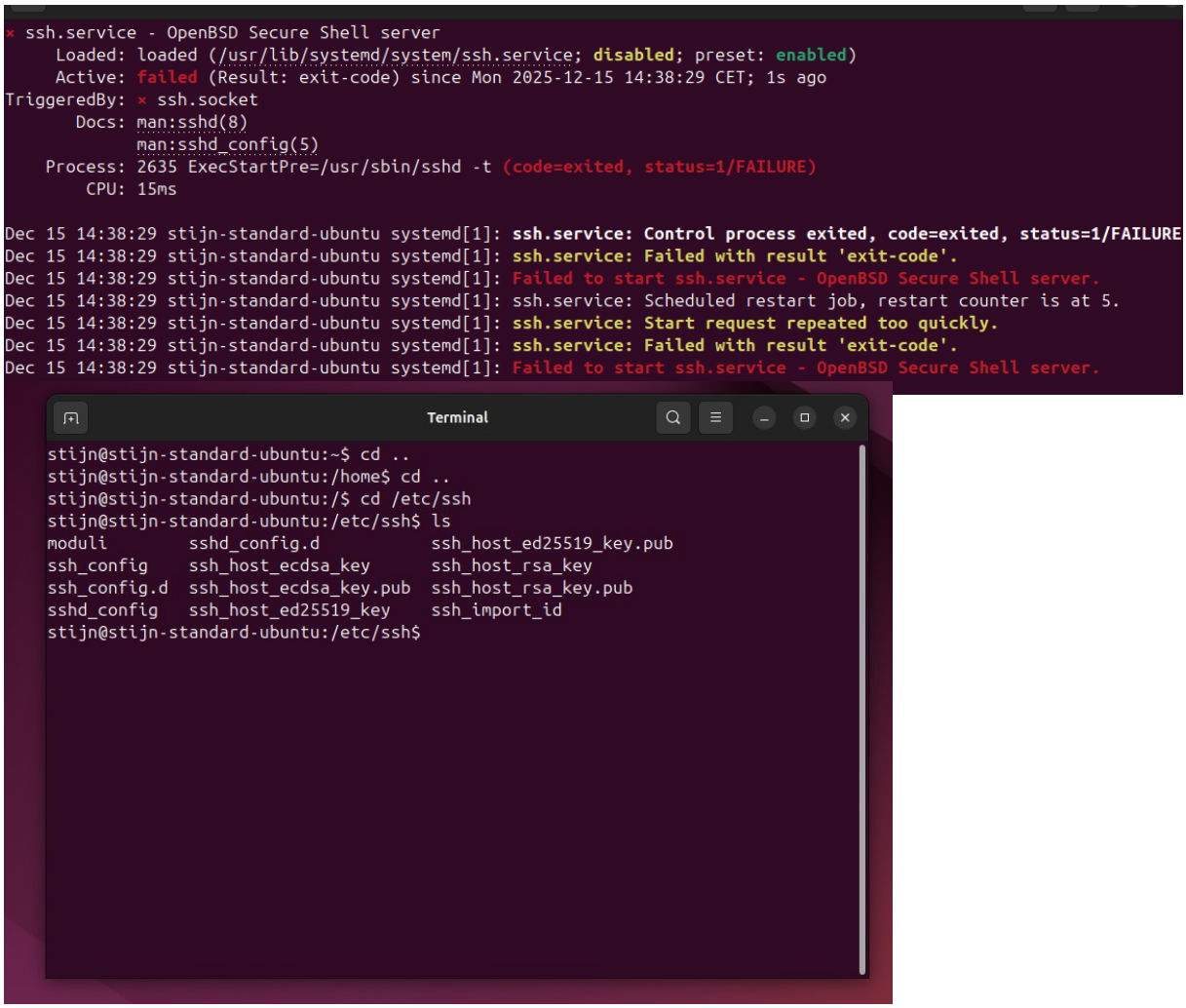
Template Week 6 – Networking

Student number: 591527

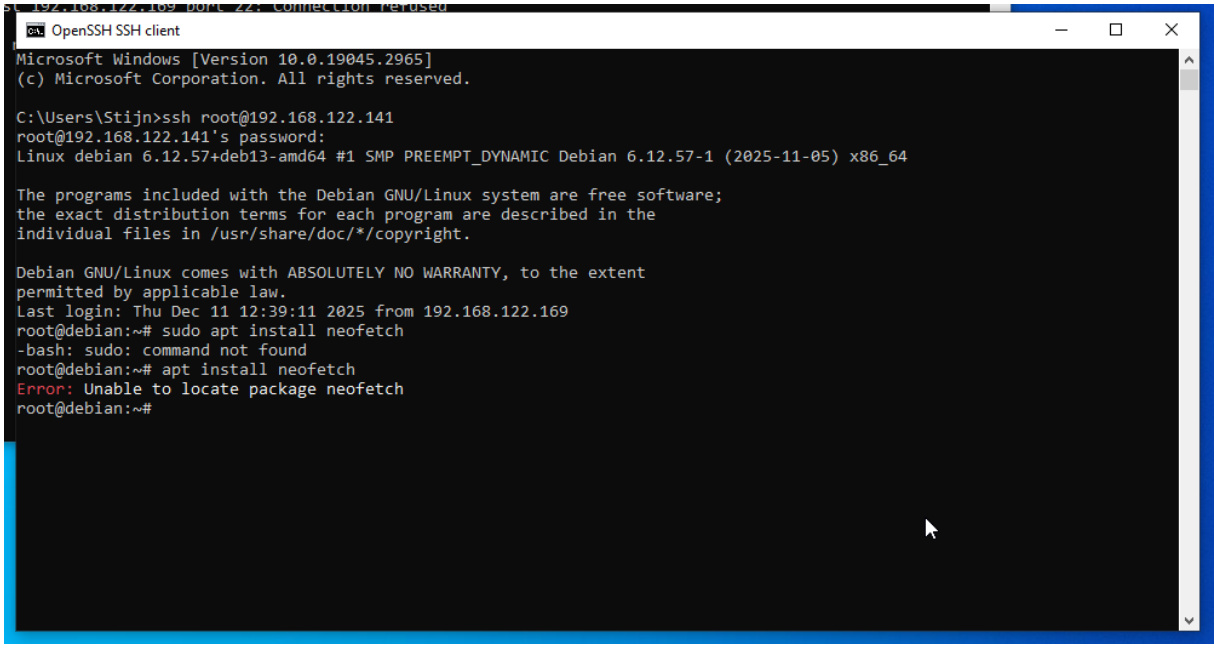
Assignment 6.1: Working from home

Screenshot installation openssh-server:

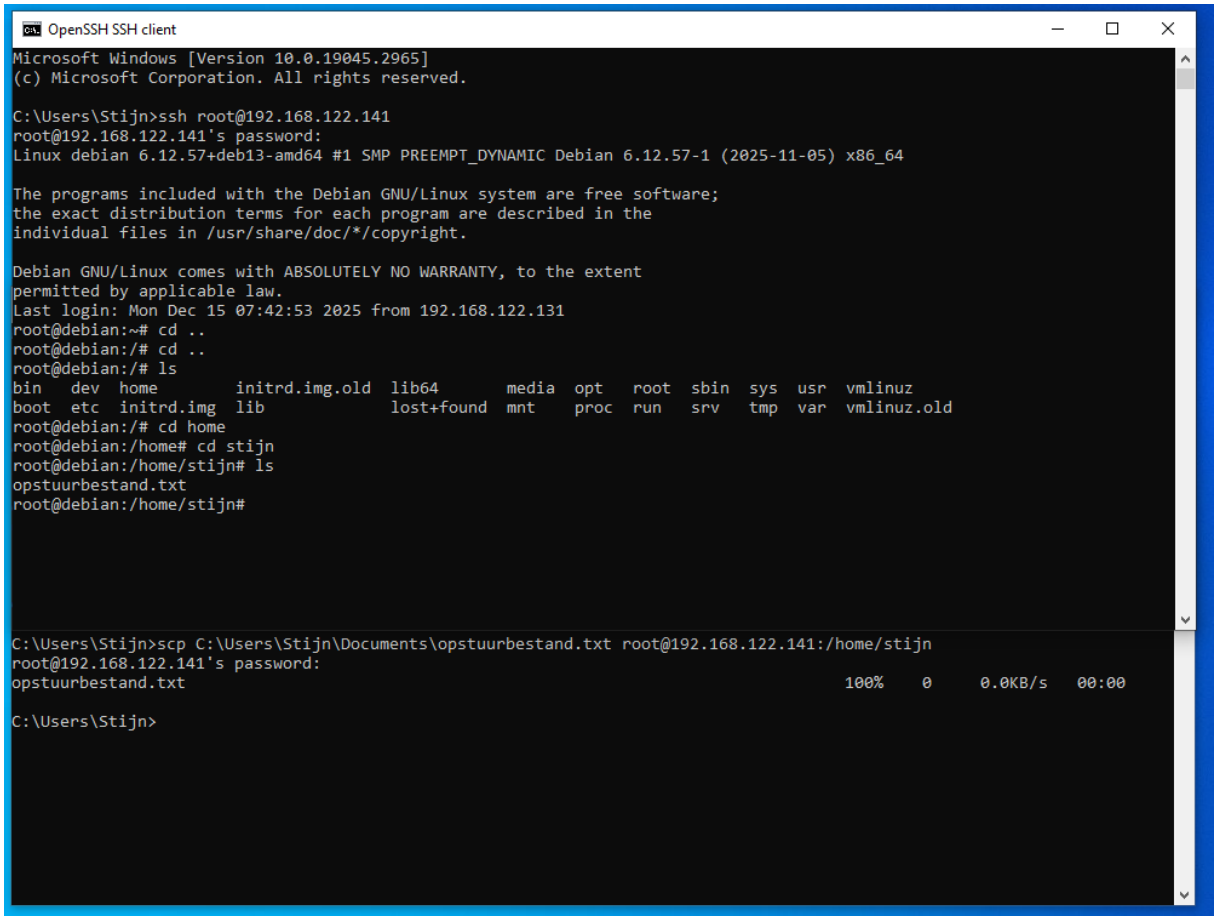
Ik kan helaas geen succesvolle verbinding krijgen met mijn Ubuntu Workstation aangezien de SSH service niet wilt starten. (Zie onderstaand screenshot). Ik ga daarom verbinding maken met de Debian Server machine. Dit is ondanks de succesvolle installatie van Openssh-server.



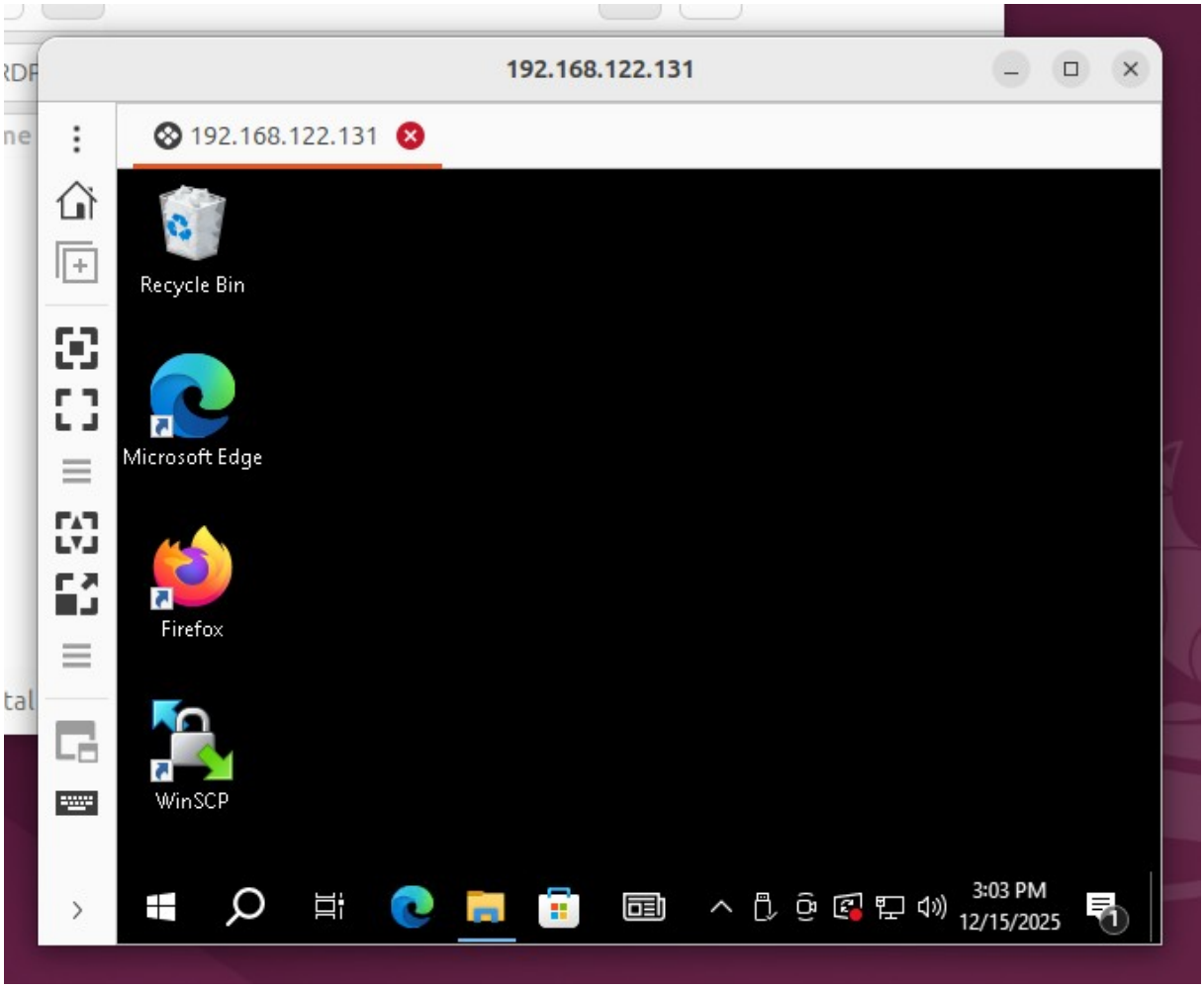
Screenshot successful SSH command execution:



Screenshot successful execution SCP command:

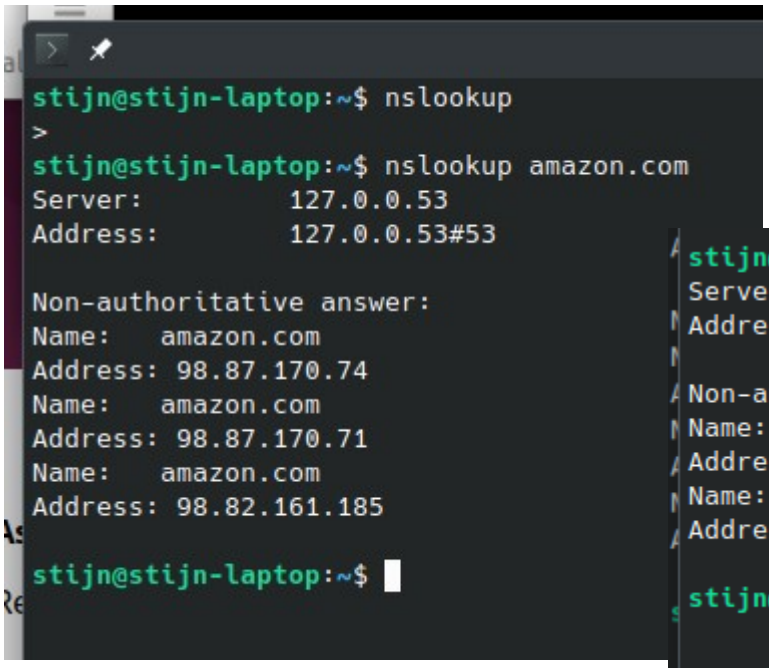


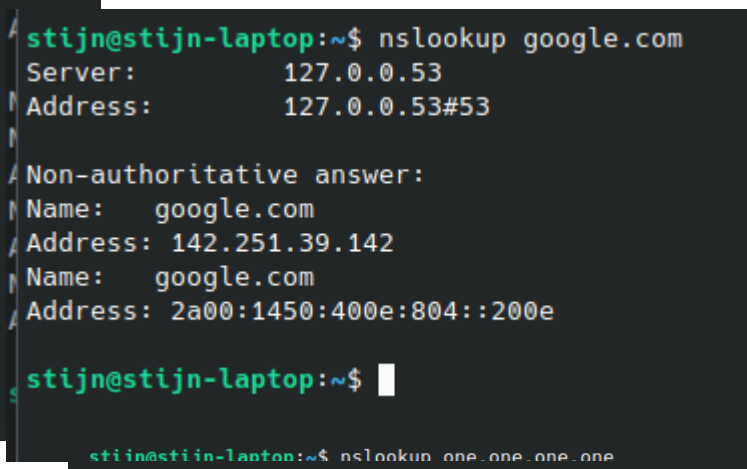
Screenshot remmina:

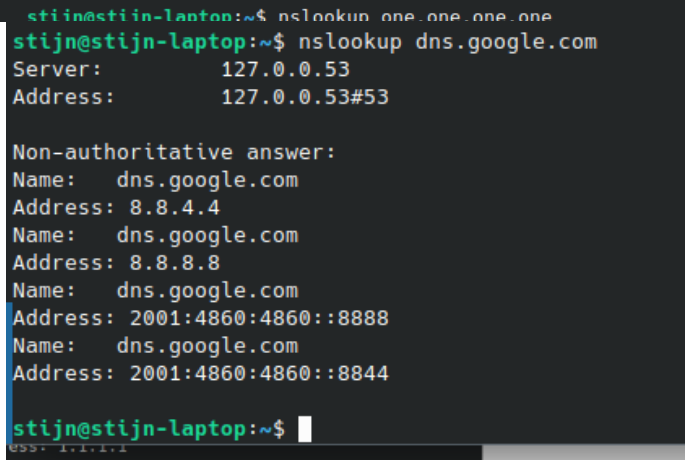


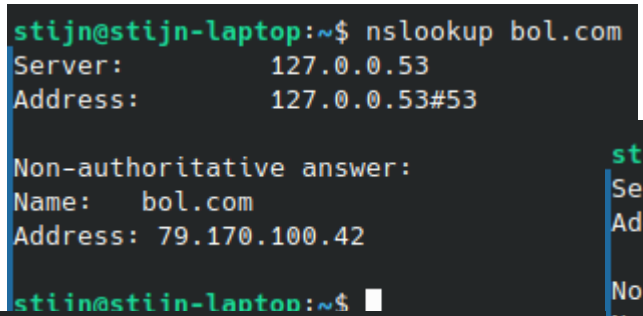
Assignment 6.2: IP addresses websites

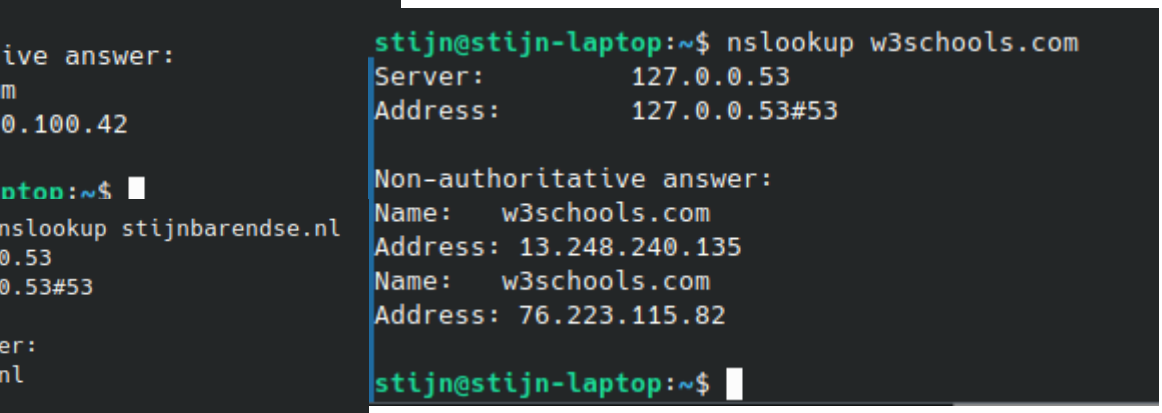
Relevant screenshots nslookup command:

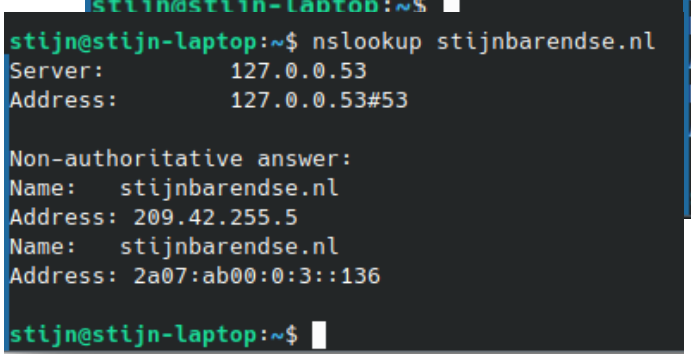




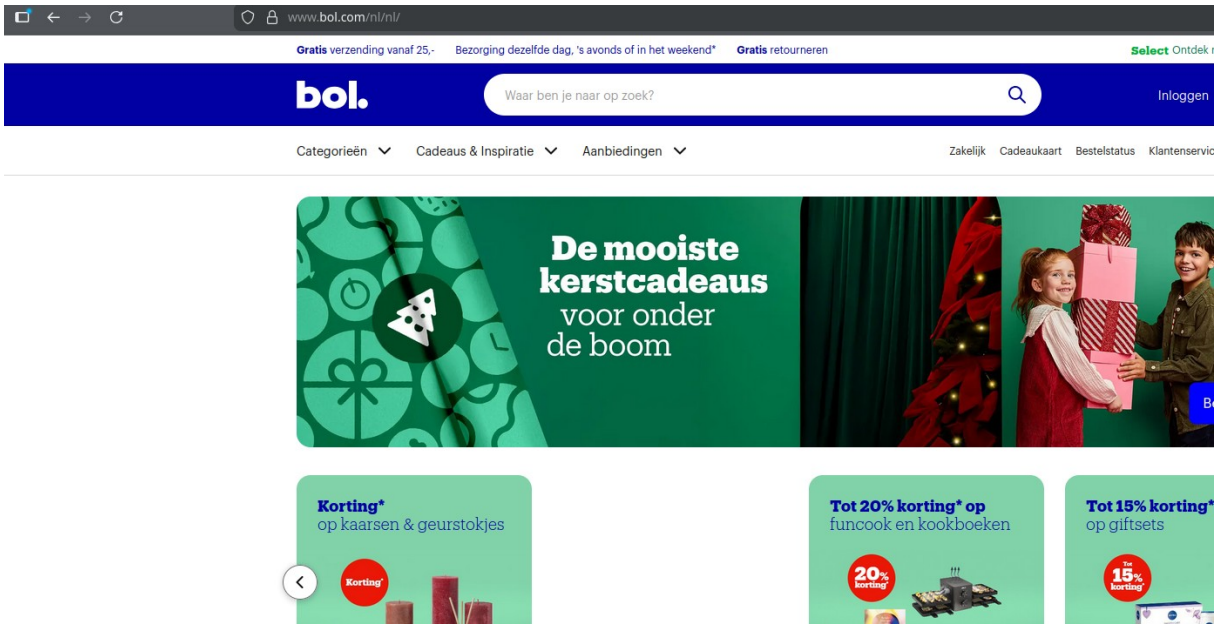








Screenshot website visit via IP address:



Als je met een grote website verbinding maakt via het IP adres, word je automatisch geredirect naar de site met DNS naam. Kan dus niet tonen.

Assignment 6.3: subnetting

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

Je hebt in totaal 128 IP adressen, maar aangezien er 1 gereserveerd is voor network, en er ook 1 is gereserveerd voor het broadcast adres zijn er 126 bruikbare adressen.

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

Dat is dus 126 ;)

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

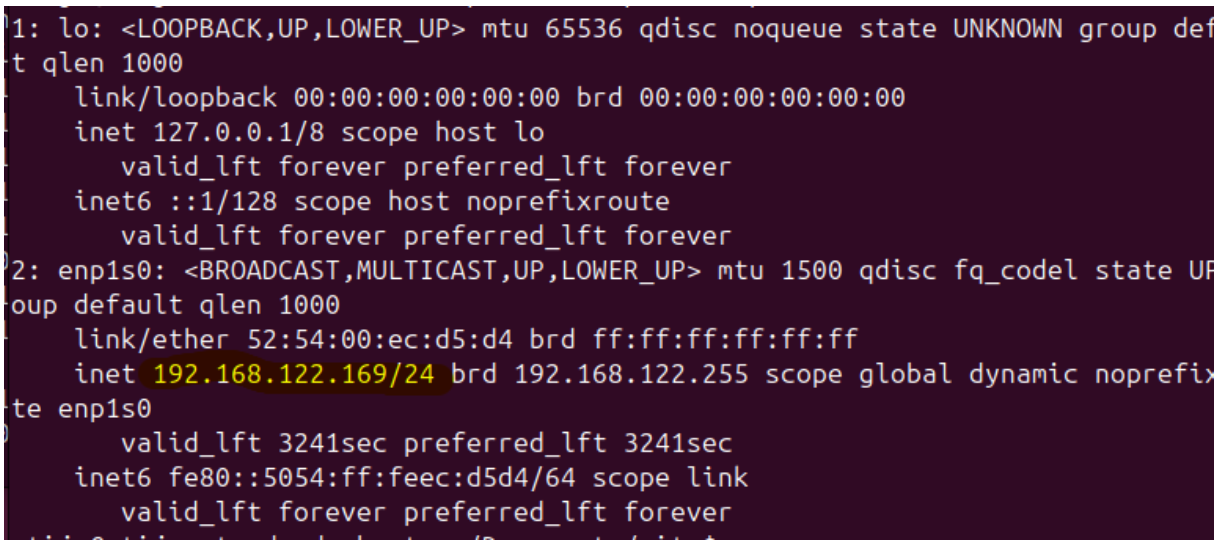
```
Processing triggers for man-db (2.12.0-4build2) ...
stijn@stijn-laptop:~$ ipcalc 192.168.110.128/25
Address:   192.168.110.128       11000000.10101000.01101110.1 0000000
Netmask:   255.255.255.128 = 25  11111111.11111111.11111111.1 0000000
Wildcard:  0.0.0.127             00000000.00000000.00000000.0 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

stijn@stijn-laptop:~$
```

Explain the above calculation in your own words.

Oke. /25 betekend dat de eerste 25 bits van het adres zijn gereserveerd voor het netwerk.
Een IPV4 adres bestaat uit 32 bits. Je doet dus 32-25=7.
Je doet vervolgens deze 7 tot de macht 2, dit is 128. Je hebt dus in totaal 128 adressen in het netwerk.
Maar aangezien er twee adressen zijn gereserveerd voor het netwerk en broadcastgedeelte, zijn er 126 bruikbare adressen.

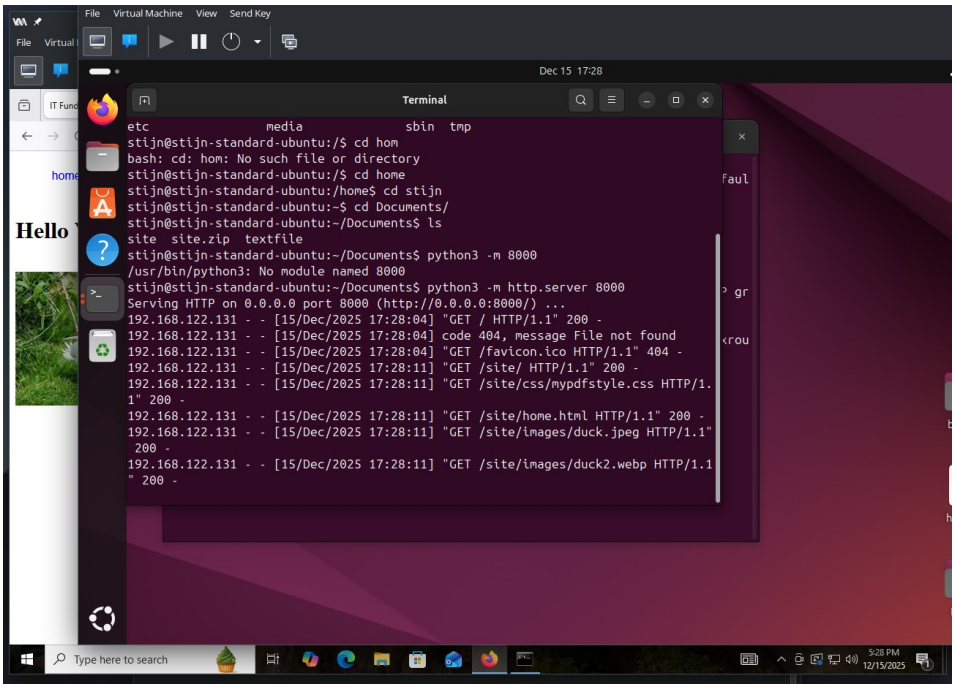
Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

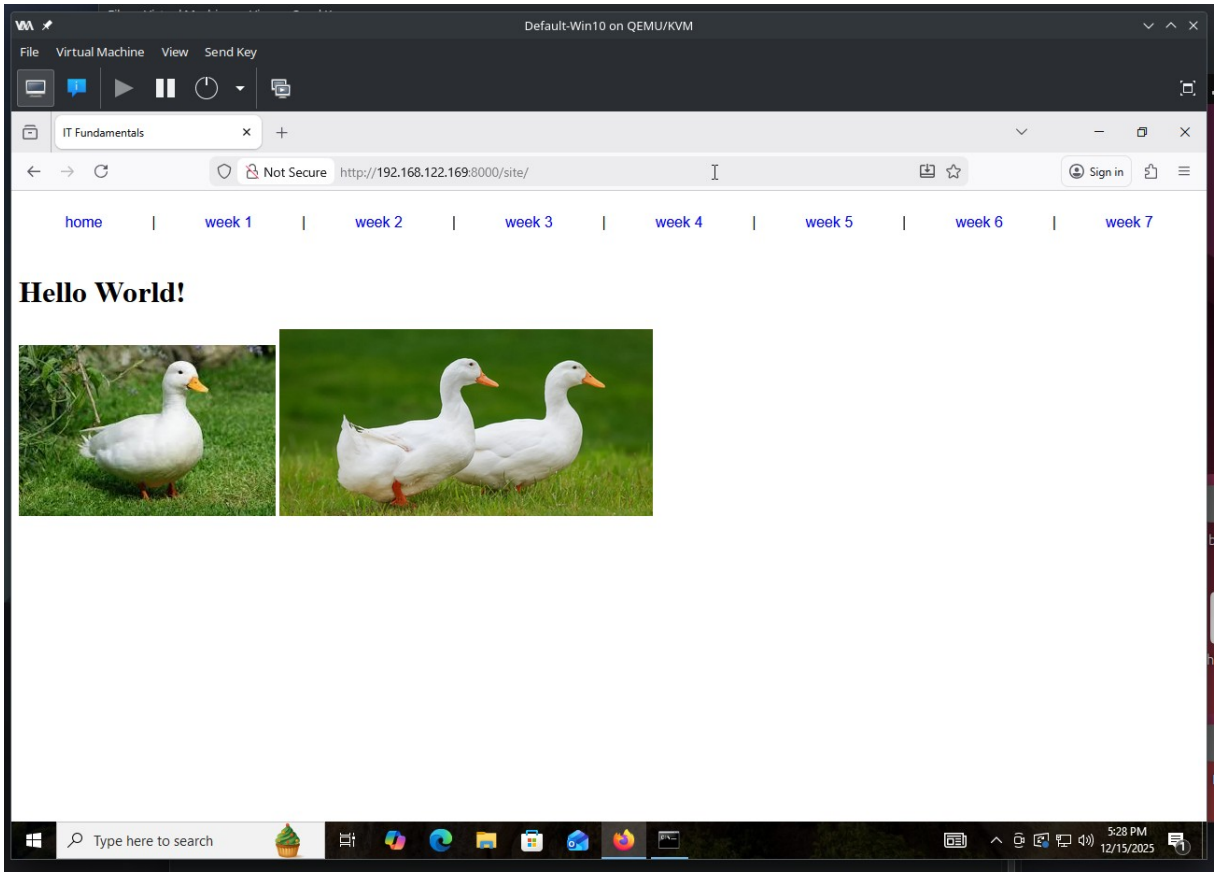


Screenshot of Site directory contents:

Screenshot python3 webserver command:



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.

```
i import java.util.Scanner;


public class Main {


   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);
      boolean programmaDraait = true;


      while (programmaDraait) {
         System.out.println("\n--- MENU ---");
         System.out.println("1. Is nummer oneven?");
         System.out.println("2. Is nummer een macht van 2?");
         System.out.println("3. Two's complement berekenen");
         System.out.println("4. Netwerk segment berekenen (IP & Subnet)");
         System.out.println("5. Stoppen");
         System.out.print("Kies een optie: ");


         int keuze = input.nextInt();


         if (keuze == 5) {
            System.out.println("Doei!");
            programmaDraait = false;
         } else if (keuze == 4) {
            // Nieuwe optie voor netwerk
            System.out.print("Voer IP in (bijv. 192.168.1.100): ");
            String ip = input.next();
```

```
                System.out.print("Voer Subnet in (bijv. 255.255.255.224): ");
                String subnet = input.next();
                berekenNetwerk(ip, subnet);
            } else {
                System.out.print("Voer een getal in: ");
                int getal = input.nextInt();

                if (keuze == 1) {
                    checkOneven(getal);
                } else if (keuze == 2) {
                    checkMachtVanTwee(getal);
                } else if (keuze == 3) {
                    berekenTwosComplement(getal);
                }
            }
        }
    }

    // --- Bestaande methodes ---
    public static void checkOneven(int n) {
        if ((n & 1) == 1) System.out.println("Oneven!");
        else System.out.println("Even!");
    }

    public static void checkMachtVanTwee(int n) {
        if (n > 0 && (n & (n - 1)) == 0) System.out.println("Macht van 2!");
        else System.out.println("Geen macht van 2.");
    }

    public static void berekenTwosComplement(int n) {
```

```
        System.out.println("Resultaat: " + (~n + 1));
    }

    // --- Nieuwe methode voor Netwerk Segment ---
    public static void berekenNetwerk(String ipStr, String subStr) {
        // Splits de strings op de punten
        String[] ipDelen = ipStr.split("\\.");
        String[] subDelen = subStr.split("\\.");

        // We maken arrays om de resultaten in op te slaan
        int[] netwerkAdres = new int[4];

        for (int i = 0; i < 4; i++) {
            int ipNummer = Integer.parseInt(ipDelen[i]);
            int subNummer = Integer.parseInt(subDelen[i]);

            // Gebruik de bitwise AND (&)
            netwerkAdres[i] = ipNummer & subNummer;
        }

        // Print het resultaat
        System.out.println("Netwerk Adres: " +
            netwerkAdres[0] + "." + netwerkAdres[1] + "." +
            netwerkAdres[2] + "." + netwerkAdres[3]);

        // Voor een /27 subnet (255.255.255.224) kun je de range ook laten zien
        // Dit is een simpele extra berekening:
        int segmentGrootte = 256 - Integer.parseInt(subDelen[3]);
        int start = netwerkAdres[3];
        int eind = start + (segmentGrootte - 1);
```

```
    System.out.println("Range: " + netwerkAdres[0] + "." + netwerkAdres[1] + "." +
                       netwerkAdres[2] + "." + start + " t/m " +
                       netwerkAdres[0] + "." + netwerkAdres[1] + "." +
                       netwerkAdres[2] + "." + eind);
  }
}
```

```
=== BITWISE OPERATIES MENU ===
1. Is getal oneven?
2. Is getal een macht van 2?
3. Two's complement van getal?
4. Bereken netwerksegment
5. Afsluiten
==============================
Voer je keuze in (1-5): 4

Voer IP-adres in (bijv. 192.168.1.100): 10.0.0.1
Voer subnetmasker in (bijv. 255.255.255.224): 255.255.255.128

=== NETWERKSEGMENT BEREKENING ===
IP-adres:        00001010.00000000.00000000.00000001
                 10.0.0.1

Subnetmasker:    11111111.11111111.11111111.10000000
                 255.255.255.128 (/25)

------------------------------------------------
Netwerkadres:    00001010.00000000.00000000.00000000
                 10.0.0.0

Netwerkbereik:   10.0.0.0 tot 10.0.0.127
Totaal Adressen: 128
Bruikbare Hosts: 126 (exclusief netwerk en broadcast)

ram op Enter om door te gaan...
```

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