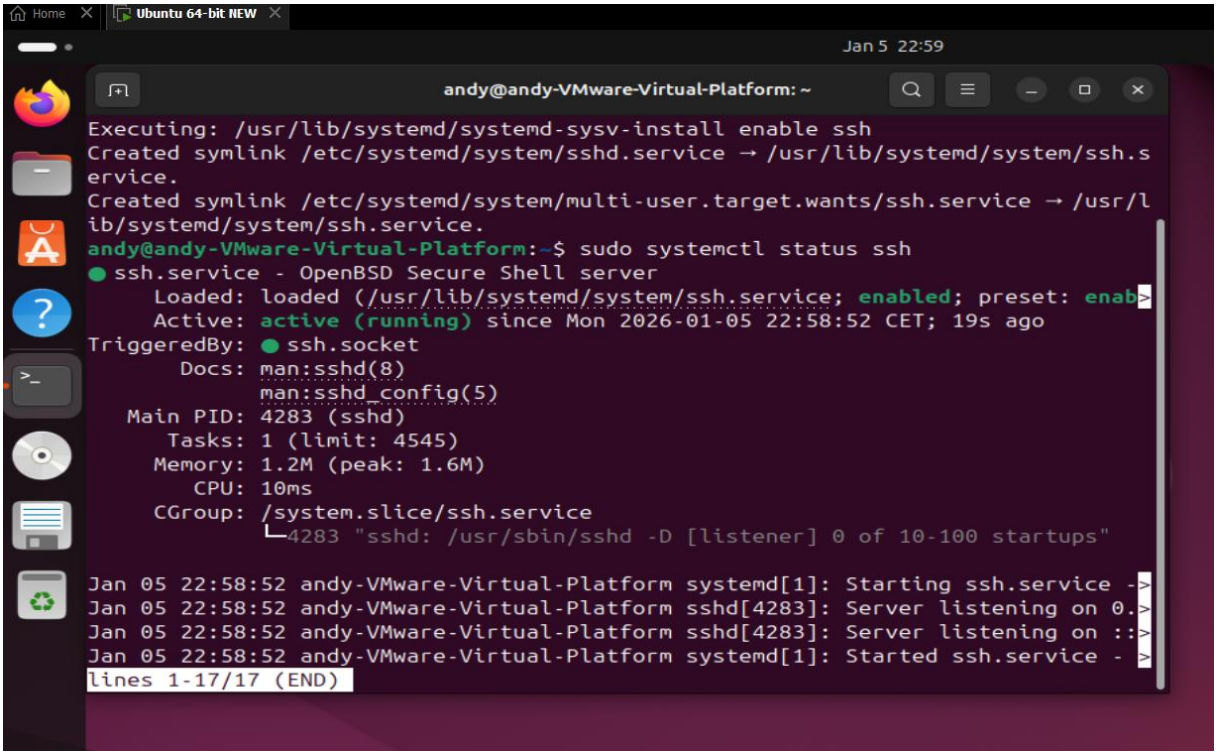
Template Week 6 – Networking

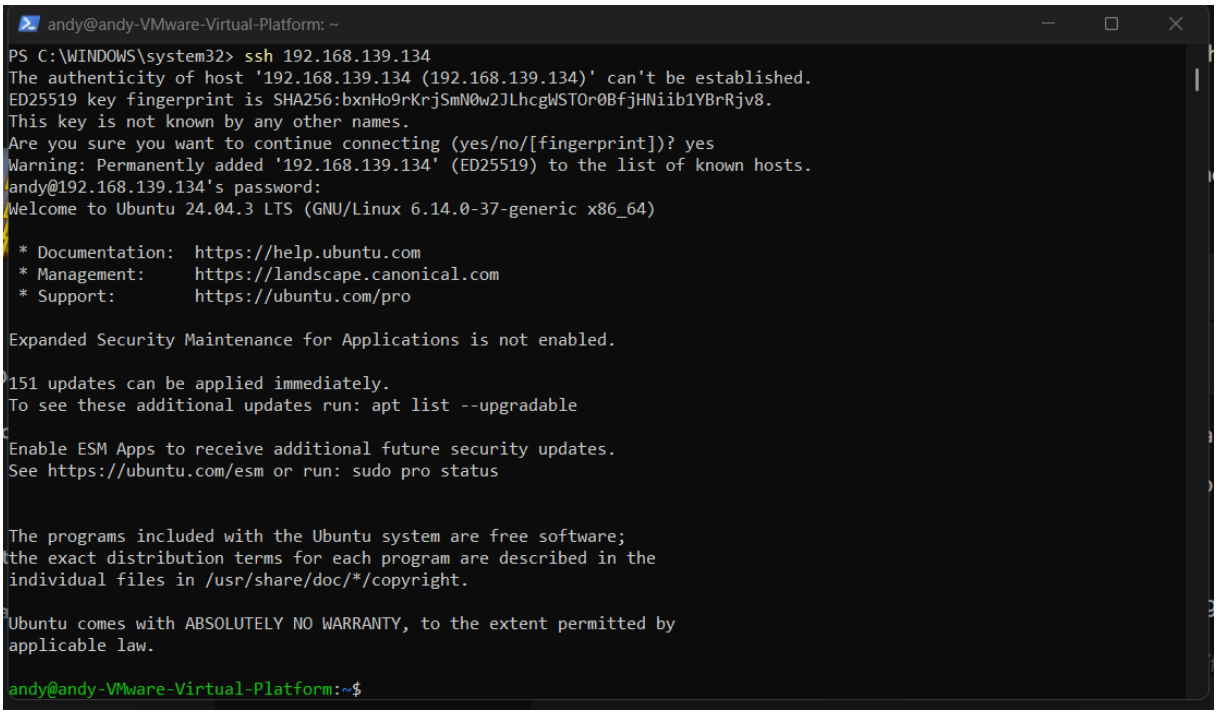
Student number:

Assignment 6.1: Working from home

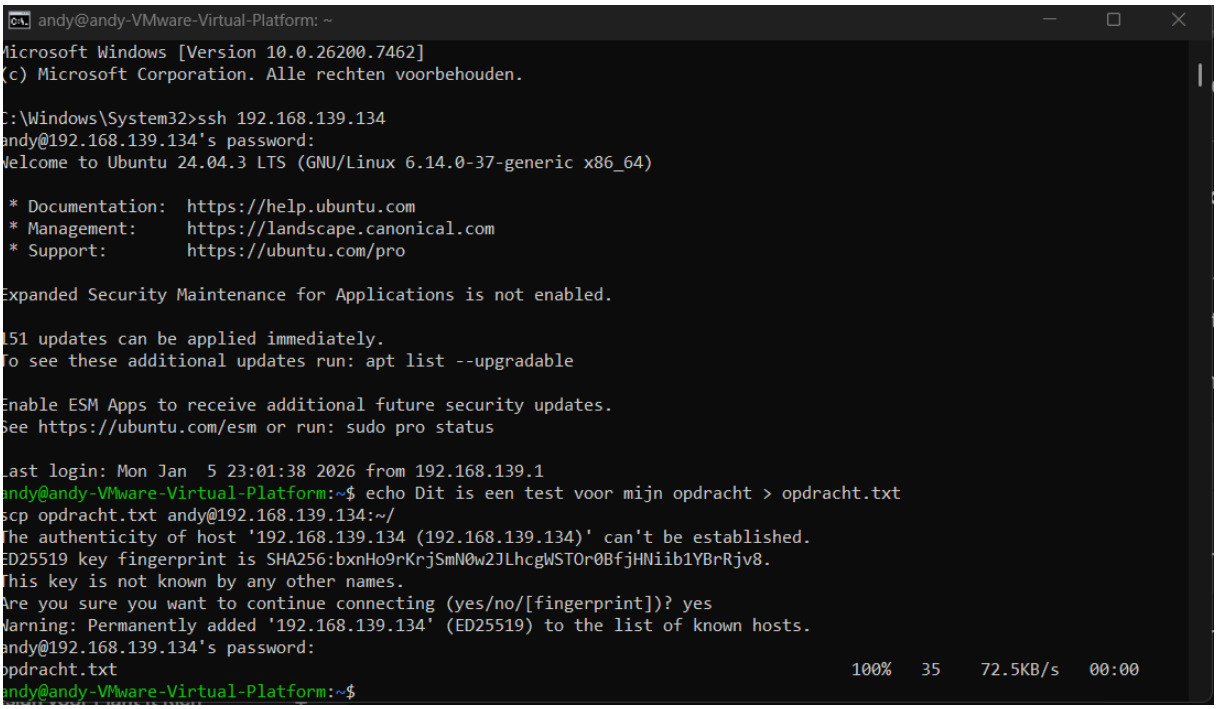
Screenshot installation openssh-server:

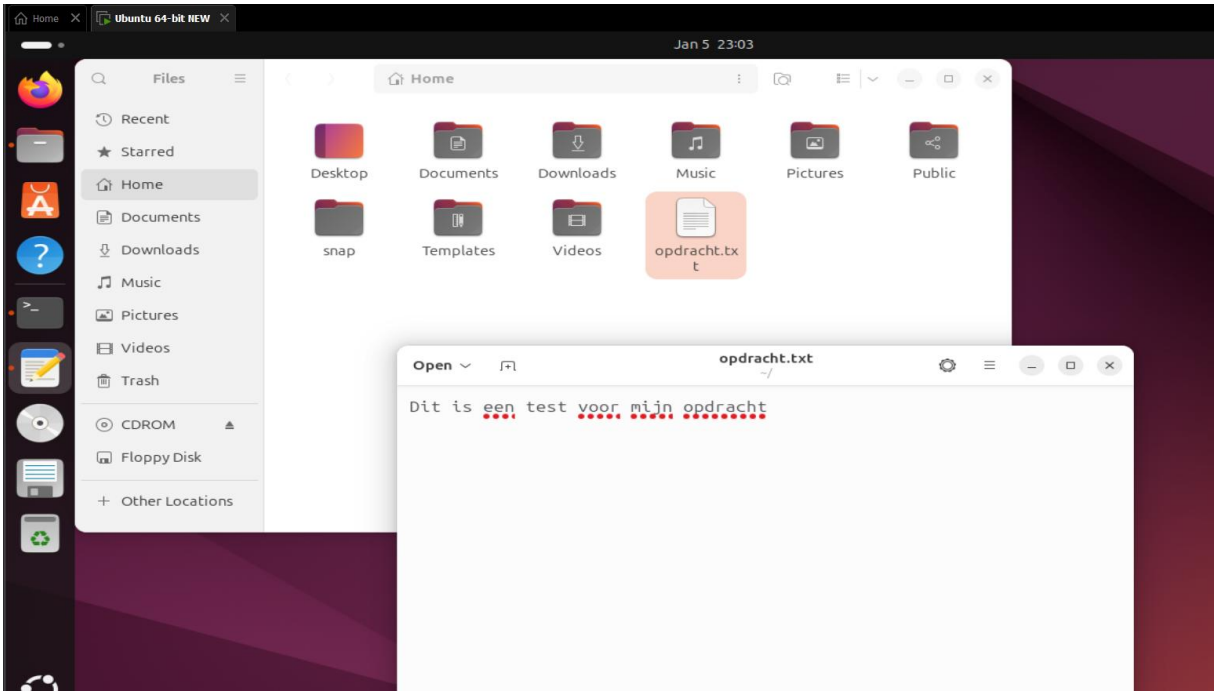


Screenshot successful SSH command execution:

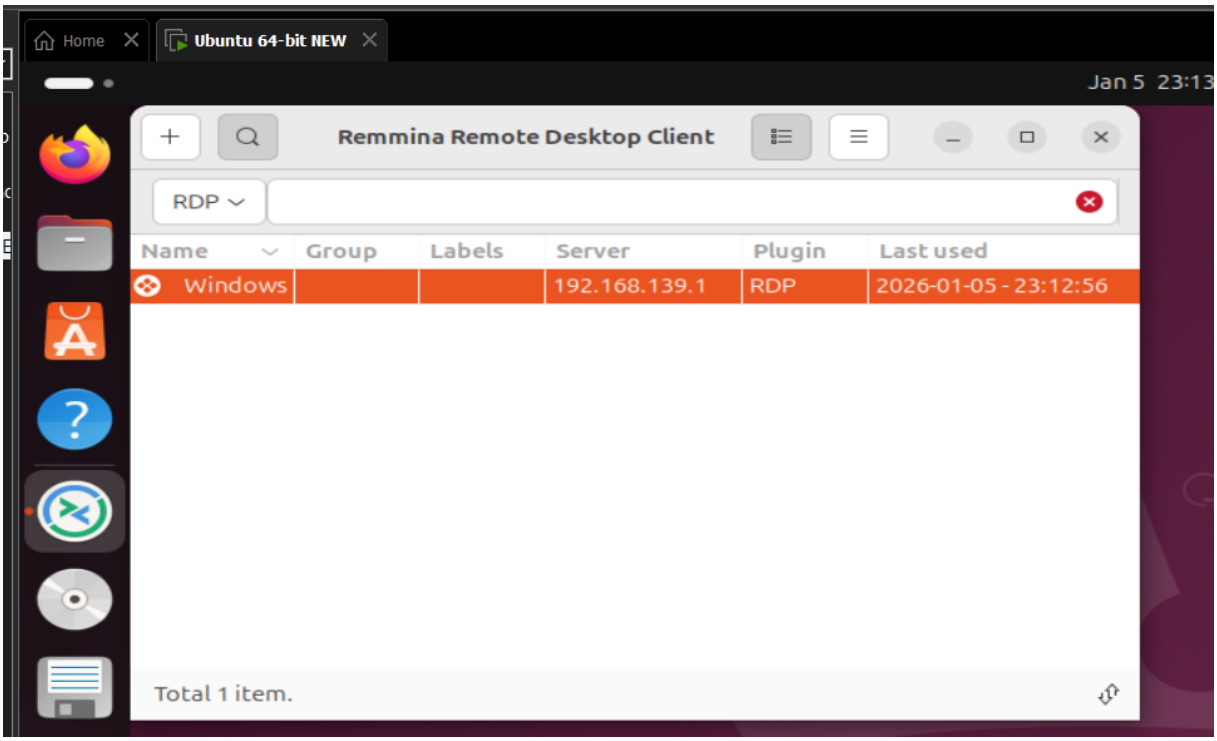


Screenshot successful execution SCP command:



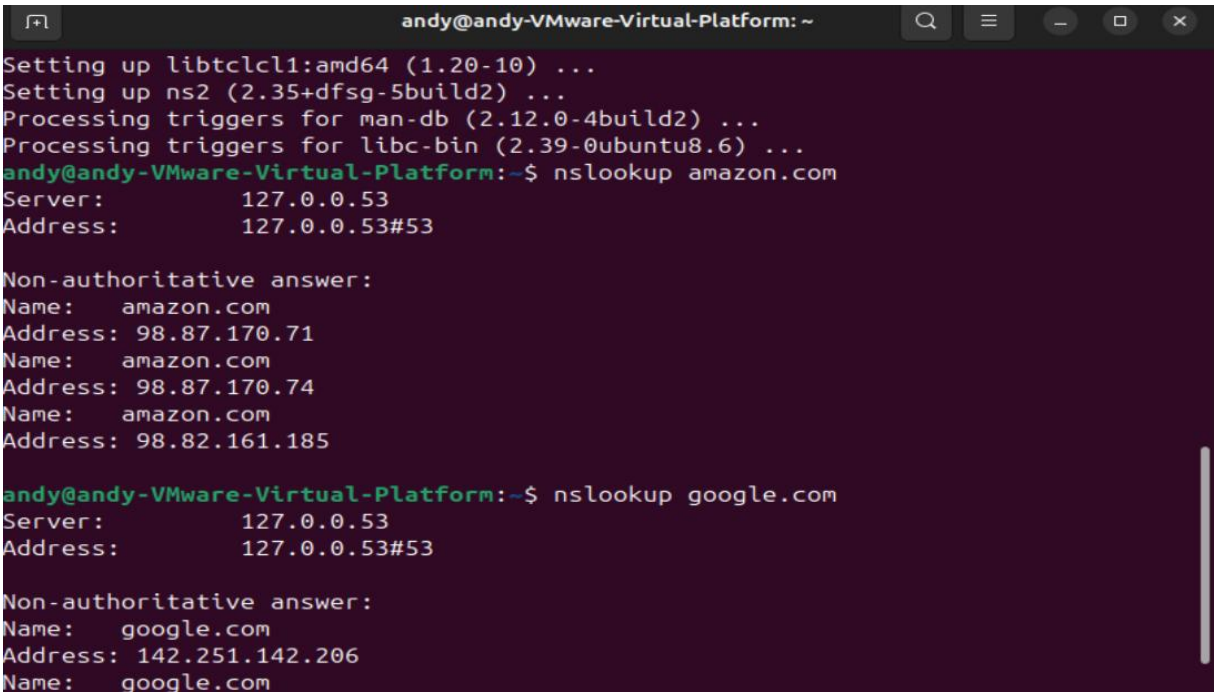


Screenshot remmina:

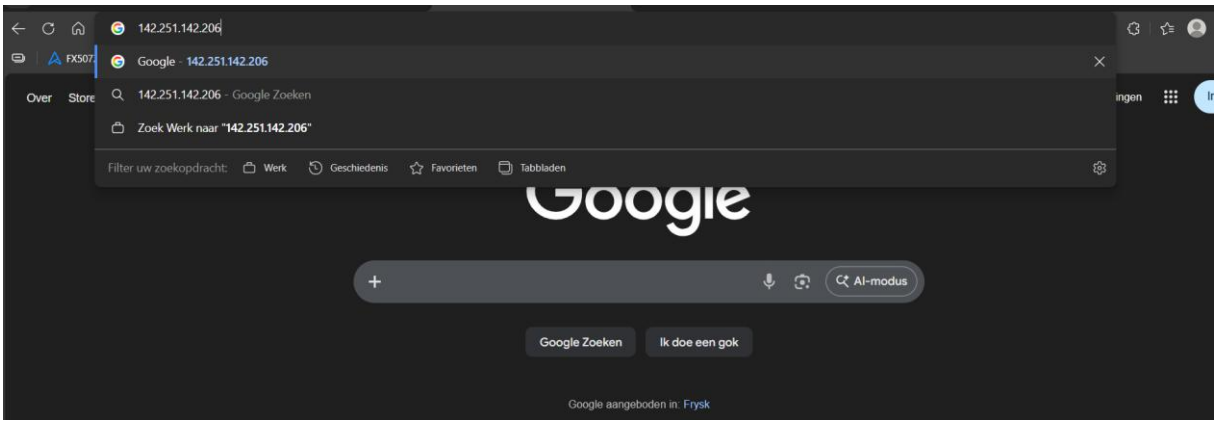


Assignment 6.2: IP addresses websites

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?

126

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

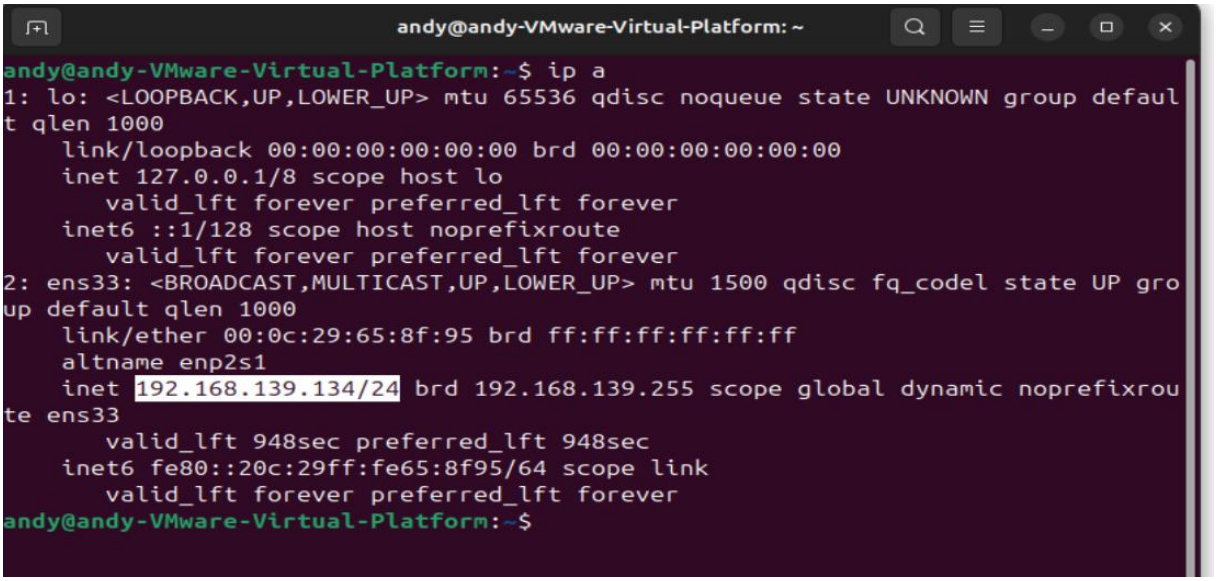
```
andy@andy-VMware-Virtual-Platform:~$ ipcalc 192.168.110.128/25
Address:   192.168.110.128       11000000.10101000.01101110.1 0000000
Netmask:   255.255.255.128 = 25  11111111.11111111.11111111.1 0000000
Wildcard:  0.0.0.127             00000000.00000000.00000000.0 1111111
=>
Network:   192.168.110.128/25    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
```

Explain the above calculation in your own words.

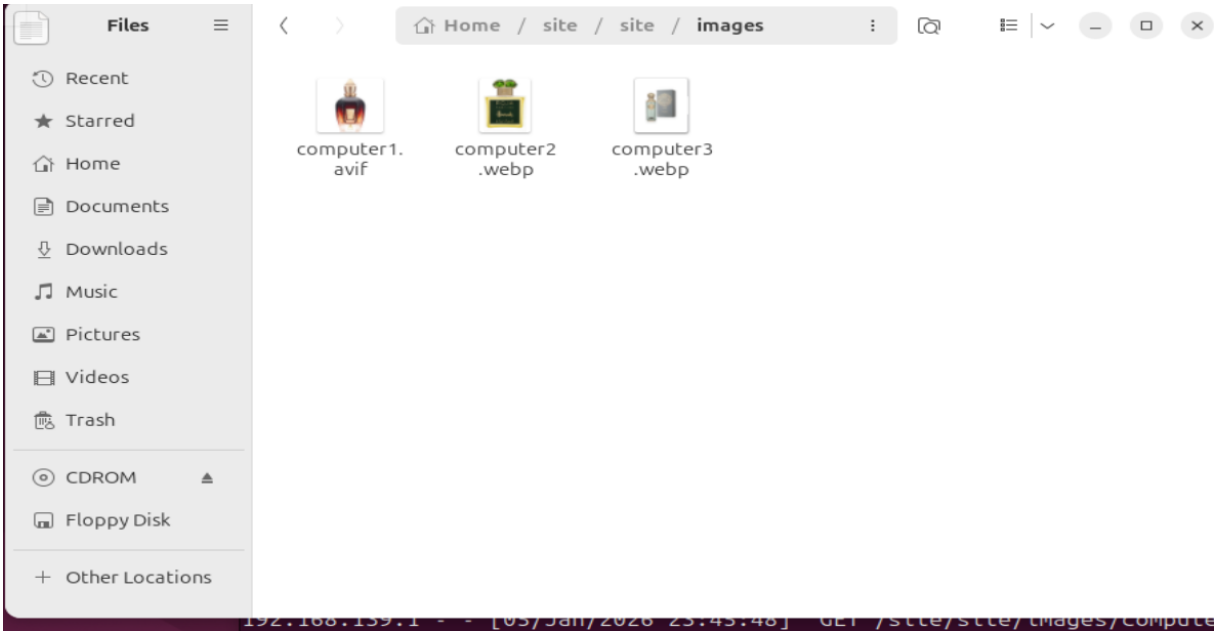
Subnet van 25, dus 32 (want zoveel bits bestaat een ipv4 adres uit) dus 32-25 = 7, dus 2^7 = 128 totaal adressen, maar de eerste en laatste zijn niet bruikbaar dus -2 dus 126

Assignment 6.4: HTML

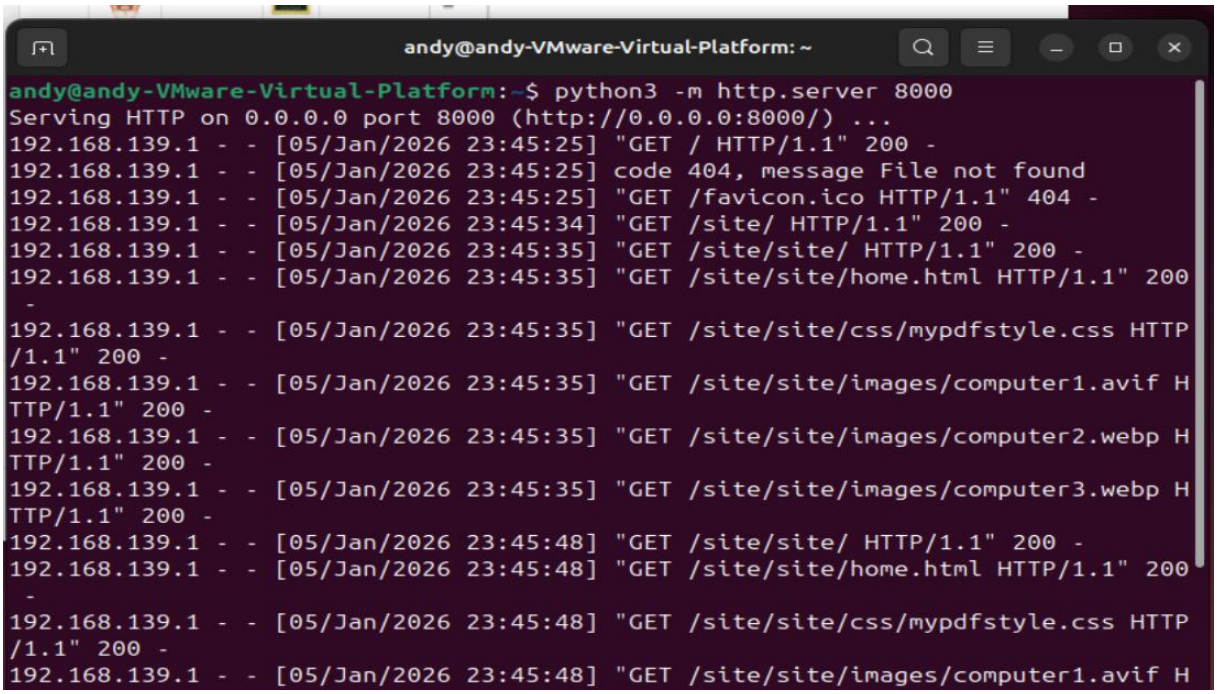
Screenshot IP address Ubuntu VM:



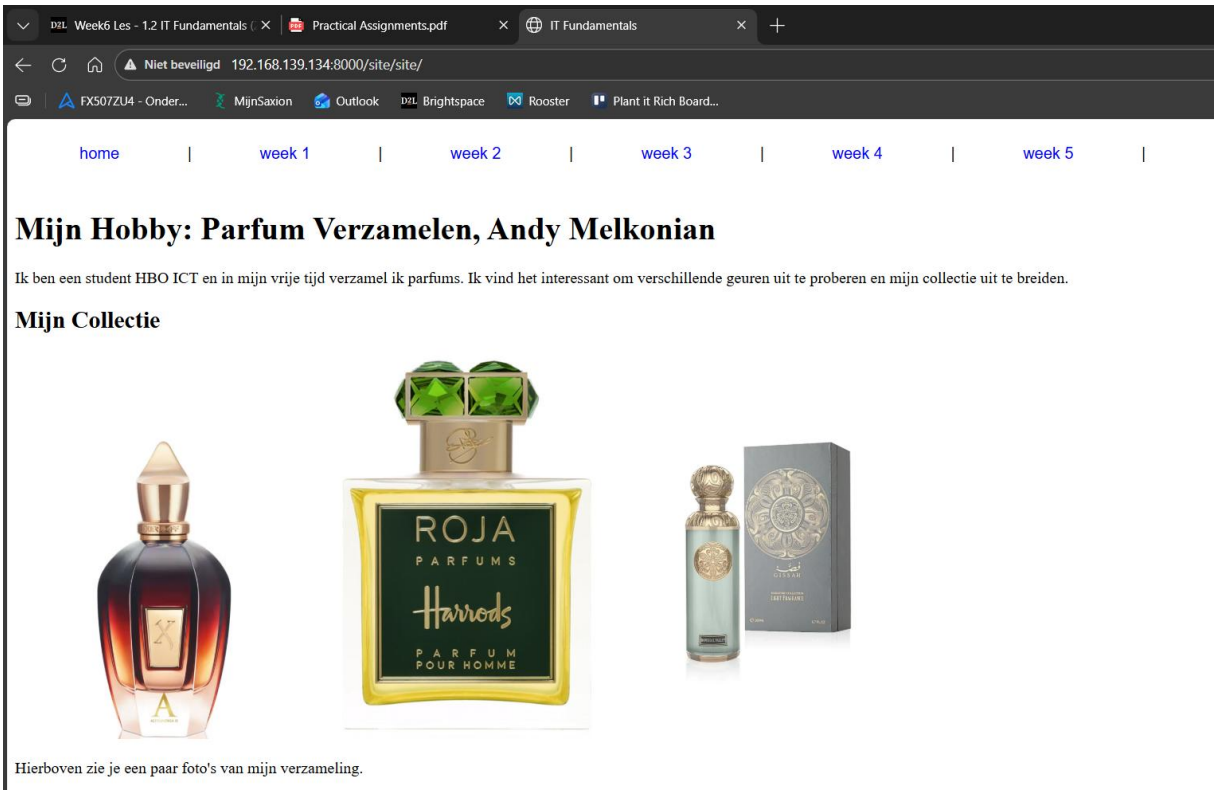
Screenshot of Site directory contents:



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

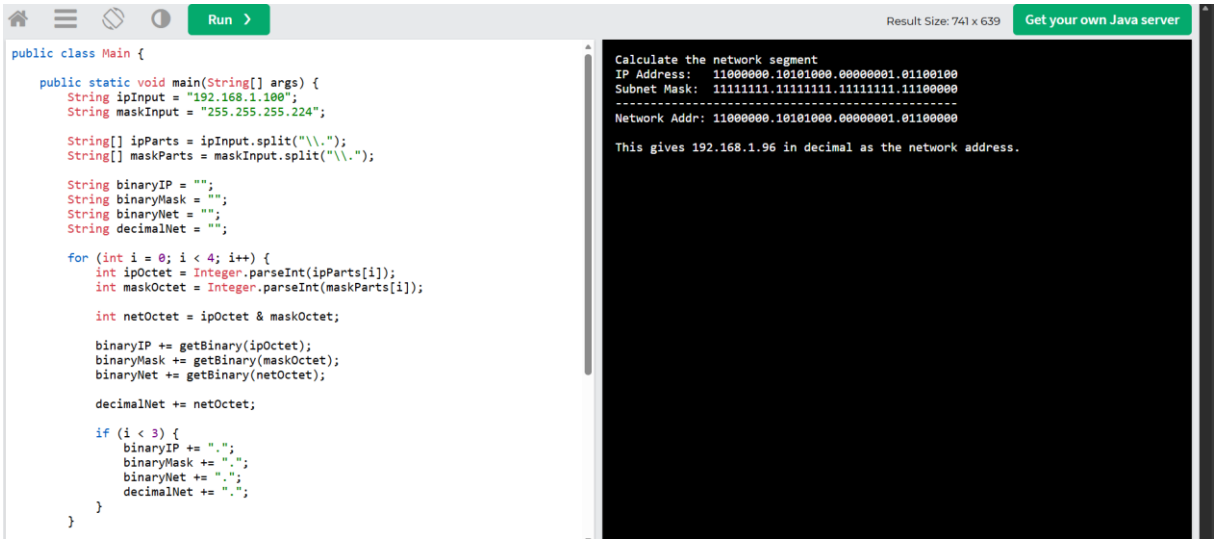
```
Example: 192.168.1.100/27
Calculate the network segment
IP Address:   11000000.10101000.00000001.01100100
Subnet Mask:  11111111.11111111.11111111.11100000
-----------------------------------------------
Network Addr: 11000000.10101000.00000001.01100000

This gives 192.168.1.96 in decimal as the network address.
For a /27 subnet, each segment (or subnet) has 32 IP addresses (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 Main {


   public static void main(String[] args) {
      String ipInput = "192.168.1.100";
      String maskInput = "255.255.255.224";


      String[] ipParts = ipInput.split("\\.");
      String[] maskParts = maskInput.split("\\.");


      String binaryIP = "";
      String binaryMask = "";
      String binaryNet = "";
      String decimalNet = "";


      for (int i = 0; i < 4; i++) {
         int ipOctet = Integer.parseInt(ipParts[i]);
         int maskOctet = Integer.parseInt(maskParts[i]);
```

```
        int netOctet = ipOctet & maskOctet;

        binaryIP += getBinary(ipOctet);
        binaryMask += getBinary(maskOctet);
        binaryNet += getBinary(netOctet);

        decimalNet += netOctet;

        if (i < 3) {
            binaryIP += ".";
            binaryMask += ".";
            binaryNet += ".";
            decimalNet += ".";
        }
    }

    System.out.println("Calculate the network segment");
    System.out.println("IP Address:   " + binaryIP);
    System.out.println("Subnet Mask:  " + binaryMask);
    System.out.println("-----------------------------------------------");
    System.out.println("Network Addr: " + binaryNet);
    System.out.println("\nThis gives " + decimalNet + " in decimal as the network address.");
}

public static String getBinary(int n) {
    String b = Integer.toBinaryString(n);
    while (b.length() < 8) {
        b = "0" + b;
    }
    return b;
}
```

}



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