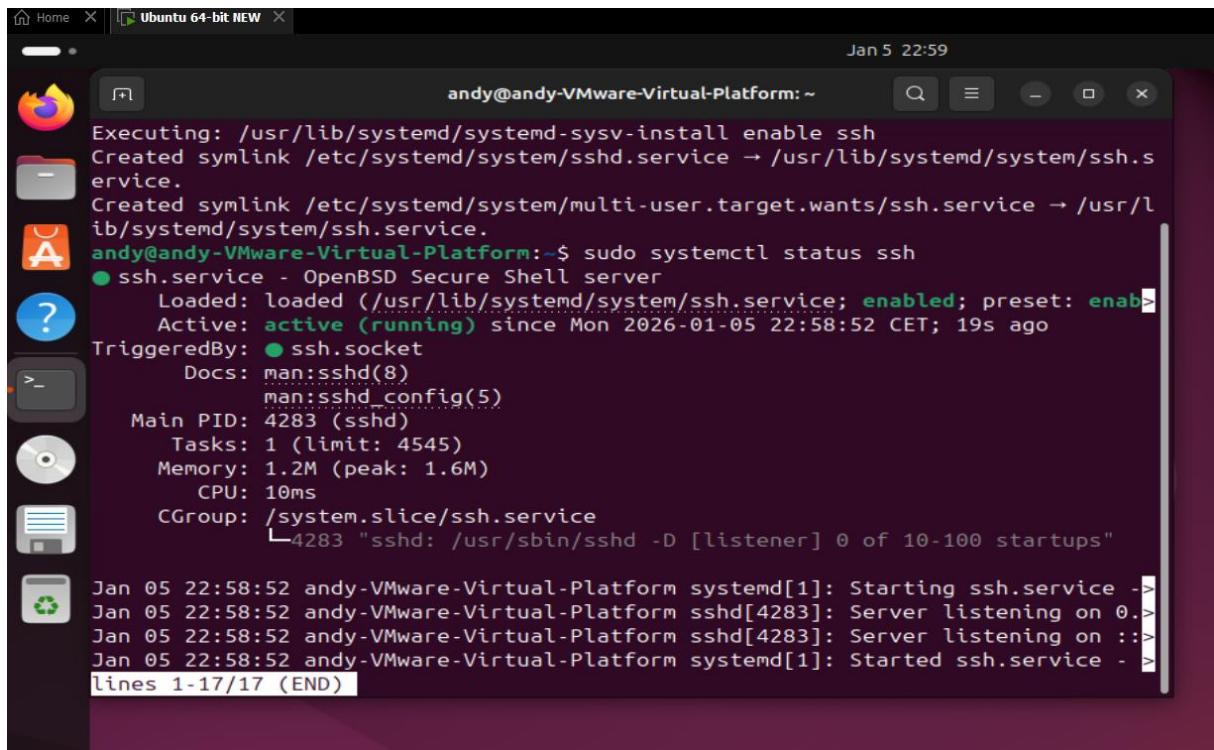


Template Week 6 – Networking

Student number:

Assignment 6.1: Working from home

Screenshot installation openssh-server:

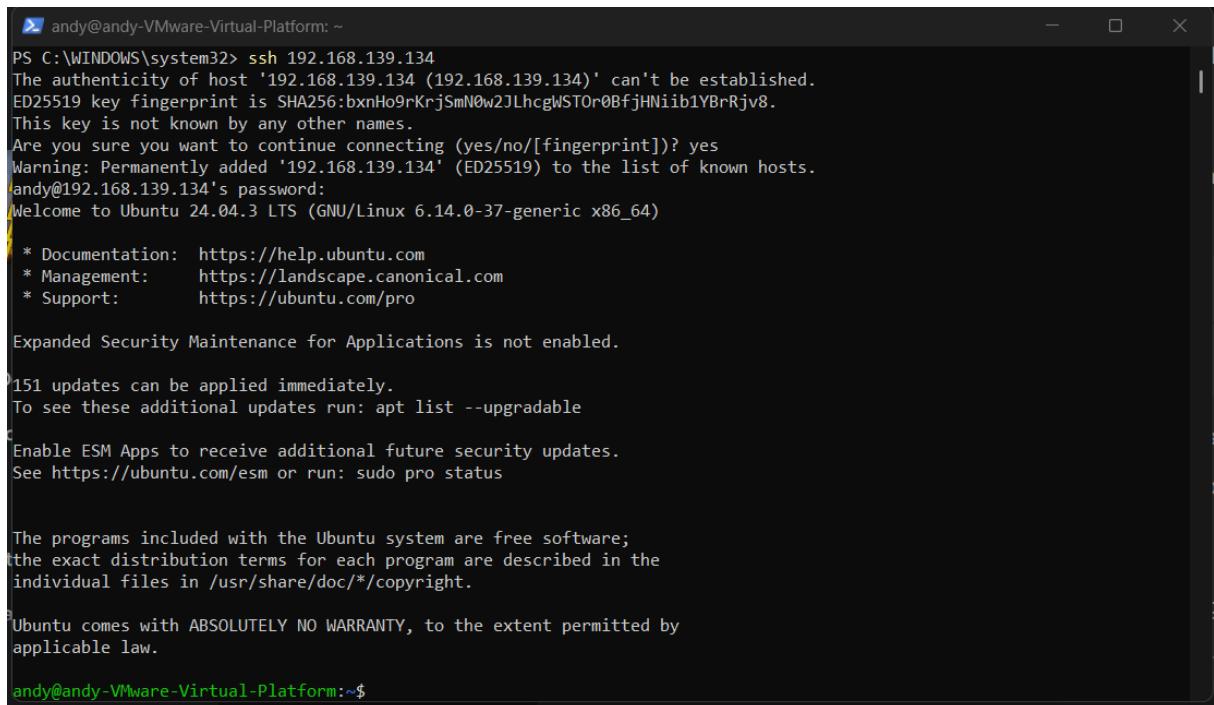


The screenshot shows a terminal window titled "Ubuntu 64-bit NEW" running on an Ubuntu system. The command `sudo systemctl status ssh` is being run, and the output shows the ssh service is active (running) since January 5, 2026. The terminal also displays log messages indicating the service is starting and listening on port 0 and ::. The bottom of the terminal shows the command history with the last 17 lines.

```
andy@andy-VMware-Virtual-Platform:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
  Active: active (running) since Mon 2026-01-05 22:58:52 CET; 19s ago
    TriggeredBy: ● ssh.socket
      Docs: man:sshd(8)
             man:sshd_config(5)
    Main PID: 4283 (sshd)
      Tasks: 1 (limit: 4545)
     Memory: 1.2M (peak: 1.6M)
        CPU: 10ms
       CGroup: /system.slice/ssh.service
               └─4283 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 05 22:58:52 andy-VMware-Virtual-Platform systemd[1]: Starting ssh.service ->
Jan 05 22:58:52 andy-VMware-Virtual-Platform sshd[4283]: Server listening on 0.0.0.0:22
Jan 05 22:58:52 andy-VMware-Virtual-Platform sshd[4283]: Server listening on :::22
Jan 05 22:58:52 andy-VMware-Virtual-Platform systemd[1]: Started ssh.service ->
lines 1-17/17 (END)
```

Screenshot successful SSH command execution:



The screenshot shows a Windows terminal window titled "andy@andy-VMware-Virtual-Platform:~". A user is connecting via SSH to an Ubuntu host at 192.168.139.134. The terminal shows the SSH key fingerprint and asks for confirmation to add the host to the list of known hosts. Once connected, the user is prompted for their password, and the Ubuntu shell prompt is shown. The terminal then displays standard Ubuntu system information, including updates, ESM Apps, and copyright notices.

```
PS C:\WINDOWS\system32> ssh 192.168.139.134
The authenticity of host '192.168.139.134 (192.168.139.134)' can't be established.
ED25519 key fingerprint is SHA256:bxnHo9rKrjSmN0w2JLhcgWST0r0BfjHNIib1YBrRjv8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.139.134' (ED25519) to the list of known hosts.
andy@192.168.139.134'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.

151 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

Screenshot successful execution SCP command:

```
andy@andy-VMware-Virtual-Platform: ~
Microsoft Windows [Version 10.0.26200.7462]
(c) Microsoft Corporation. Alle rechten voorbehouden.

C:\Windows\System32>ssh 192.168.139.134
andy@192.168.139.134's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-37-generic x86_64)

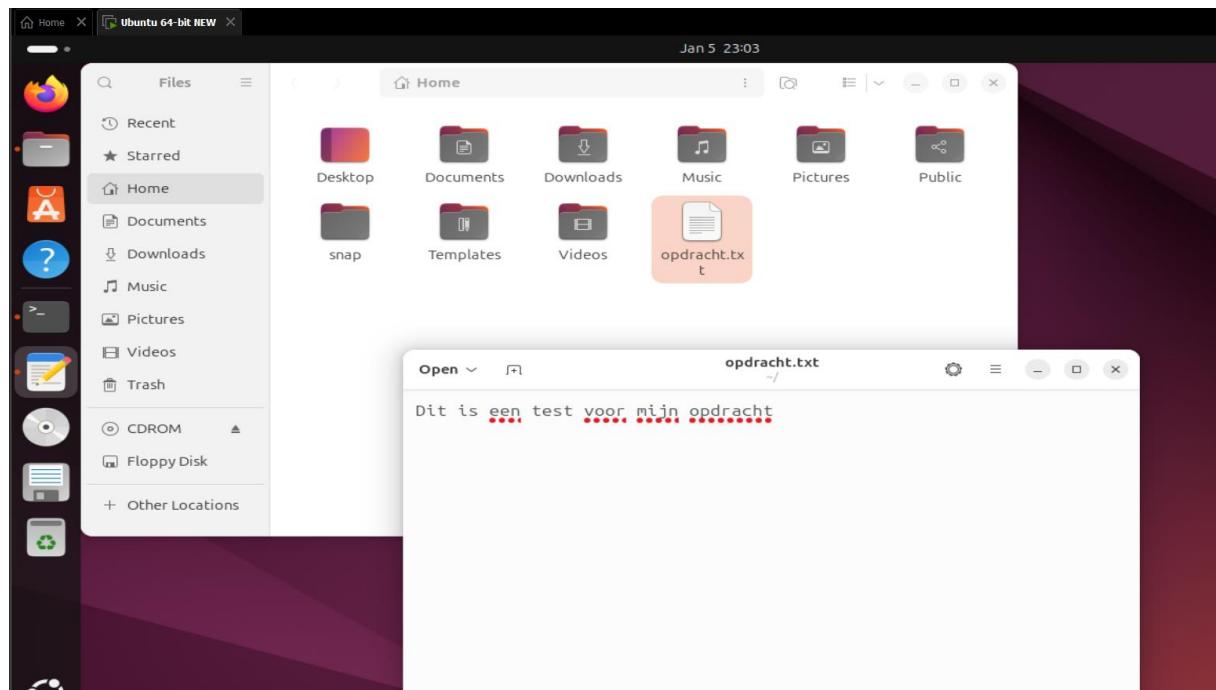
 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

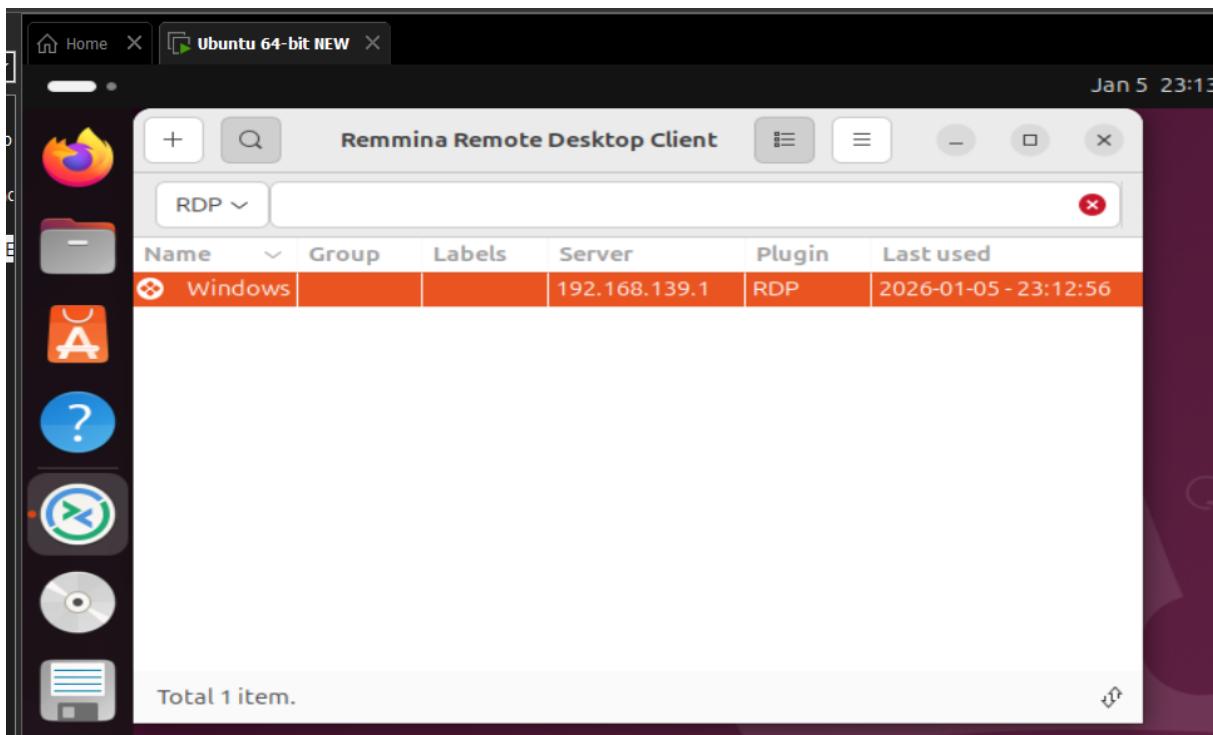
151 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Mon Jan  5 23:01:38 2026 from 192.168.139.1
andy@andy-VMware-Virtual-Platform:~$ echo Dit is een test voor mijn opdracht > opdracht.txt
scp opdracht.txt andy@192.168.139.134:~/
The authenticity of host '192.168.139.134 (192.168.139.134)' can't be established.
ED25519 key fingerprint is SHA256:bxnHo9rKrjSmN0w2JLhcgWSTOr0BfjHNiib1YBrRjv8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.139.134' (ED25519) to the list of known hosts.
andy@192.168.139.134's password:
opdracht.txt
andy@andy-VMware-Virtual-Platform:~$
```



Screenshot remmina:



Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

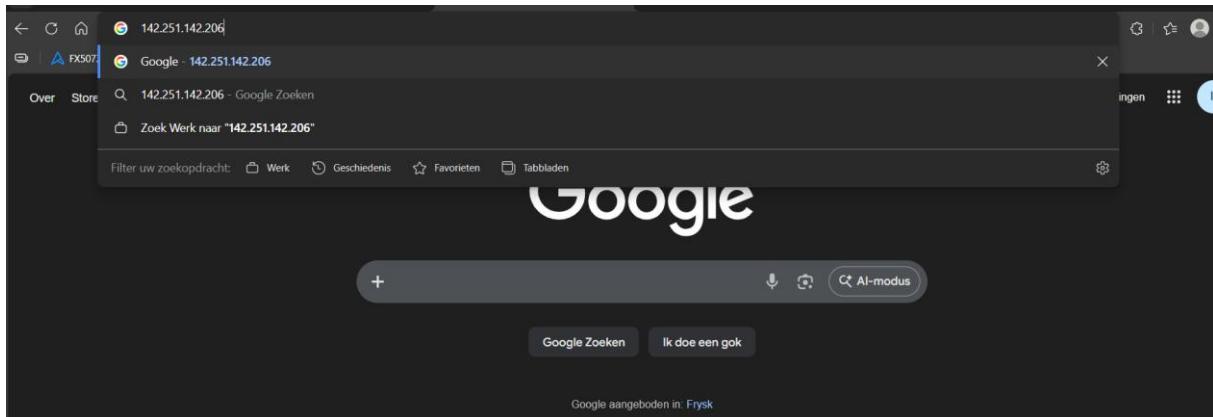
```
andy@andy-VMware-Virtual-Platform:~$ nslookup amazon.com
Setting up libtclcl1:amd64 (1.20-10) ...
Setting up ns2 (2.35+dfsg-5build2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
andy@andy-VMware-Virtual-Platform:~$ nslookup amazon.com
Server:      127.0.0.53
Address:     127.0.0.53#53

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

andy@andy-VMware-Virtual-Platform:~$ nslookup google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.251.142.206
Name:   google.com
```

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`

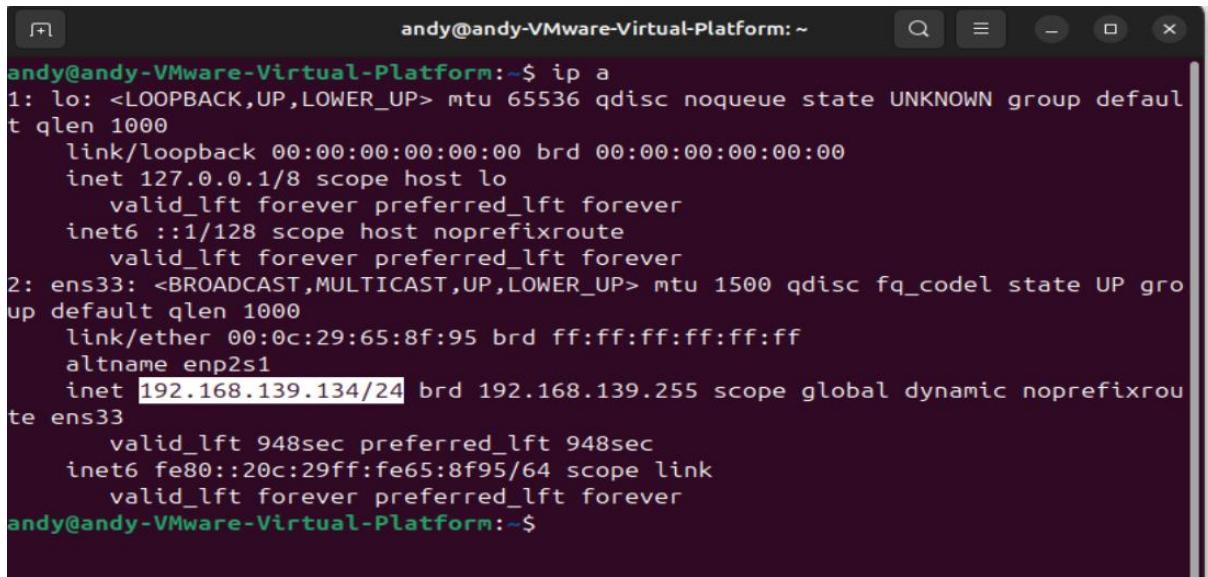
```
andy@andy-VMware-Virtual-Platform:~$ ipcalc 192.168.110.128/25
Address: 192.168.110.128      11000000.10101000.01101110.1 00000000
Netmask: 255.255.255.128 = 25 11111111.11111111.11111111.1 00000000
Wildcard: 0.0.0.127          00000000.00000000.00000000.0 1111111
=>
Network: 192.168.110.128/25 11000000.10101000.01101110.1 00000000
HostMin: 192.168.110.129    11000001.10101000.01101110.1 00000001
HostMax: 192.168.110.254    11000001.10101000.01101110.1 1111110
Broadcast: 192.168.110.255   11000001.10101000.01101110.1 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

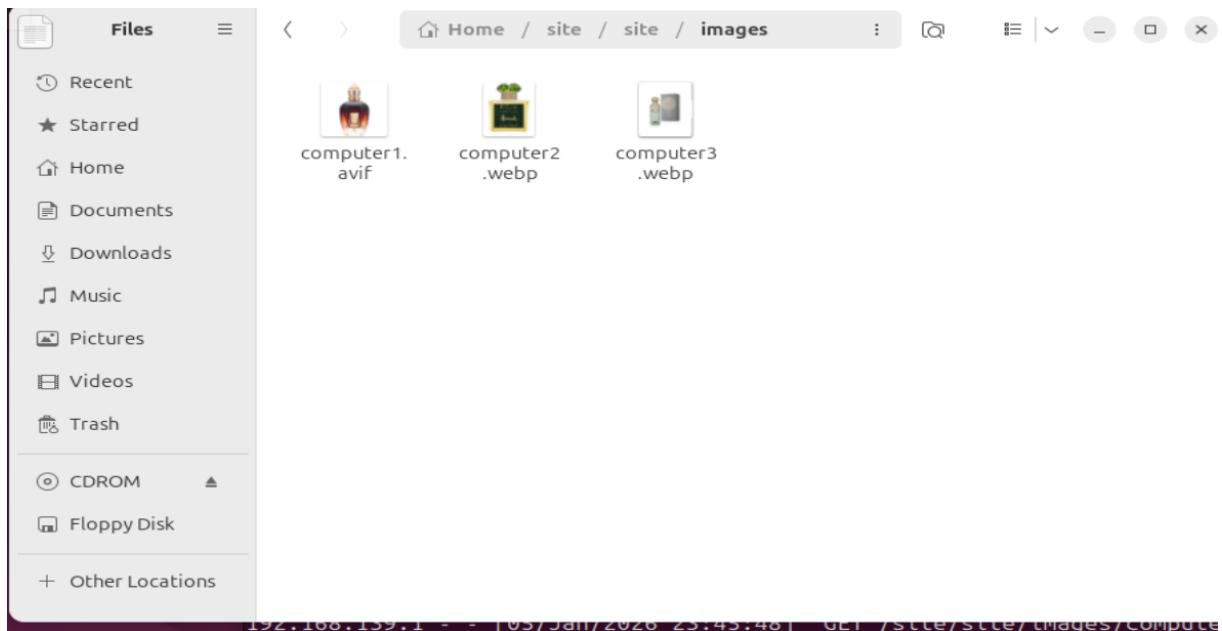
Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:



```
andy@andy-VMware-Virtual-Platform:~$ 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:29:65:8f:95 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.139.134/24 brd 192.168.139.255 scope global dynamic noprefixroute
        valid_lft 948sec preferred_lft 948sec
    inet6 fe80::20c:29ff:fe65:8f95/64 scope link
        valid_lft forever preferred_lft forever
andy@andy-VMware-Virtual-Platform:~$
```

Screenshot of Site directory contents:



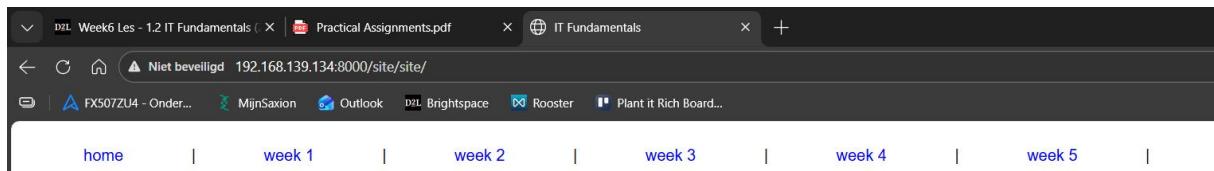
Screenshot python3 webserver command:

```

andy@andy-VMware-Virtual-Platform:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.139.1 - - [05/Jan/2026 23:45:25] "GET / HTTP/1.1" 200 -
192.168.139.1 - - [05/Jan/2026 23:45:25] code 404, message File not found
192.168.139.1 - - [05/Jan/2026 23:45:25] "GET /favicon.ico HTTP/1.1" 404 -
192.168.139.1 - - [05/Jan/2026 23:45:34] "GET /site/ HTTP/1.1" 200 -
192.168.139.1 - - [05/Jan/2026 23:45:35] "GET /site/site/ HTTP/1.1" 200 -
192.168.139.1 - - [05/Jan/2026 23:45:35] "GET /site/site/home.html HTTP/1.1" 200
-
192.168.139.1 - - [05/Jan/2026 23:45:35] "GET /site/site/css/mypdfstyle.css HTTP/1.1" 200 -
192.168.139.1 - - [05/Jan/2026 23:45:35] "GET /site/site/images/computer1.avif HTTP/1.1" 200 -
192.168.139.1 - - [05/Jan/2026 23:45:35] "GET /site/site/images/computer2.webp HTTP/1.1" 200 -
192.168.139.1 - - [05/Jan/2026 23:45:35] "GET /site/site/images/computer3.webp HTTP/1.1" 200 -
192.168.139.1 - - [05/Jan/2026 23:45:48] "GET /site/site/ HTTP/1.1" 200 -
192.168.139.1 - - [05/Jan/2026 23:45:48] "GET /site/site/home.html HTTP/1.1" 200
-
192.168.139.1 - - [05/Jan/2026 23:45:48] "GET /site/site/css/mypdfstyle.css HTTP/1.1" 200 -
192.168.139.1 - - [05/Jan/2026 23:45:48] "GET /site/site/images/computer1.avif HTTP/1.1" 200 -

```

Screenshot web browser visits your site



Mijn Hobby: Parfum Verzamelen, Andy Melkonian

Ik ben een student HBO ICT en in mijn vrije tijd verzamel ik parfums. Ik vind het interessant om verschillende geuren uit te proberen en mijn collectie uit te breiden.

Mijn Collectie



Hierboven zie je een paar foto's van mijn verzameling.

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.

```
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;
}

```

}

The screenshot shows a Java code editor with a green 'Run' button at the top. The code is as follows:

```
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 += ".";  
            }  
        }  
    }  
}
```

The output window shows the results of the calculation:

```
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.
```

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