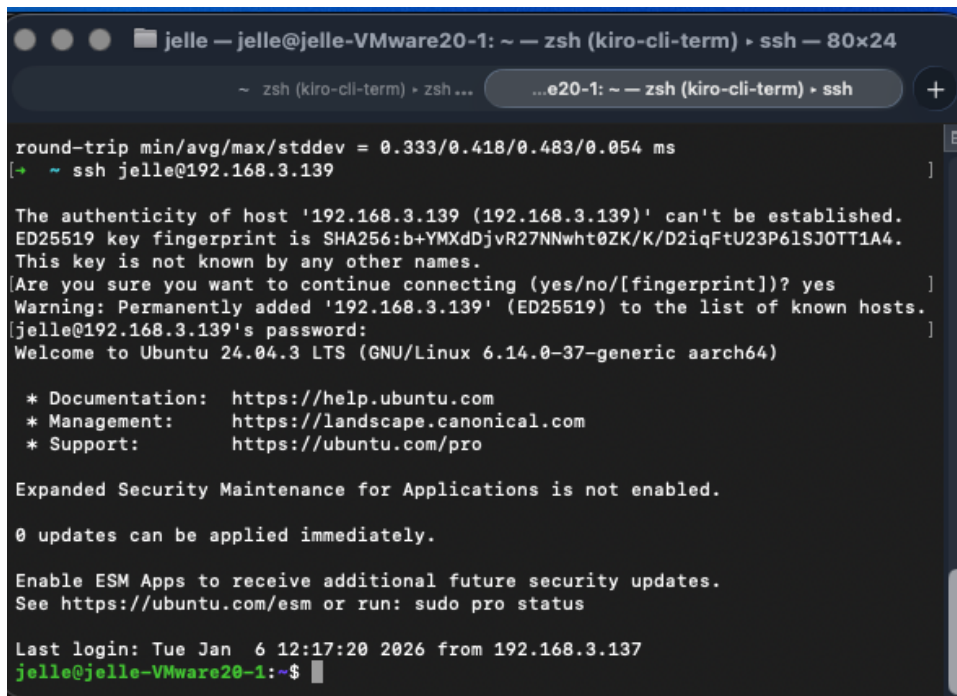


Template Week 6 – Networking

Student number: 589530

Assignment 6.1: Working from home

Screenshot installation openssh-server:



```
jelle — jelle@jelle-VMware20-1: ~ — zsh (kiri-cli-term) » ssh — 80x24
~ zsh (kiri-cli-term) » zsh ... ...e20-1: ~ — zsh (kiri-cli-term) » ssh

round-trip min/avg/max/stddev = 0.333/0.418/0.483/0.054 ms
[+] ~ ssh jelle@192.168.3.139

The authenticity of host '192.168.3.139 (192.168.3.139)' can't be established.
ED25519 key fingerprint is SHA256:b+YMXdDjvR27NNwht0ZK/K/D2iqFtU23P6lSJ0TT1A4.
This key is not known by any other names.
[Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.3.139' (ED25519) to the list of known hosts.
jelle@192.168.3.139's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-37-generic aarch64)

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

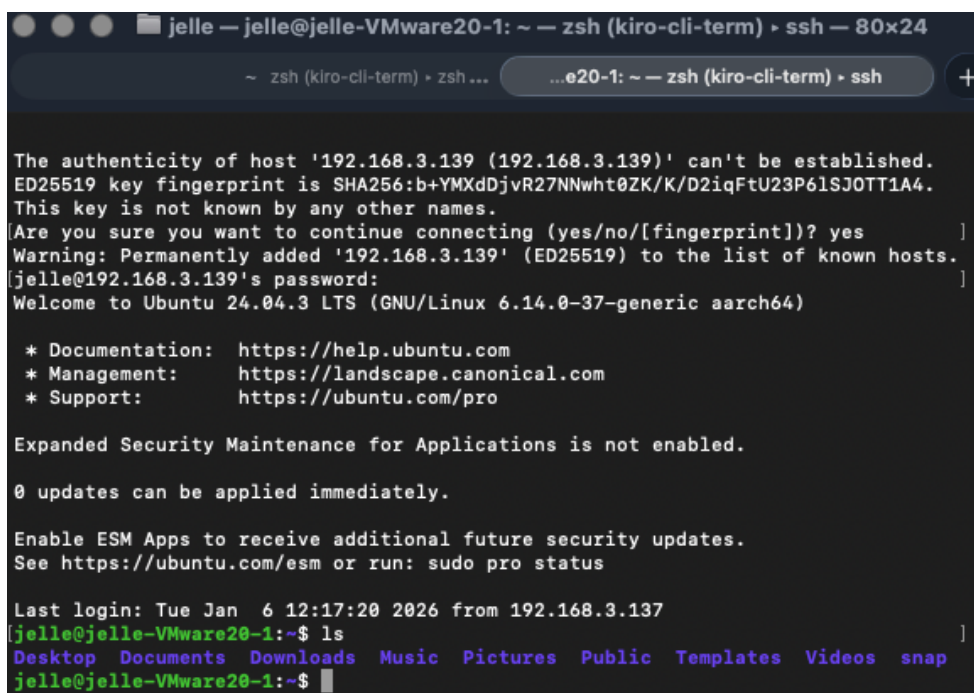
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

Last login: Tue Jan  6 12:17:20 2026 from 192.168.3.137
jelle@jelle-VMware20-1:~$
```

Screenshot successful SSH command execution:



```
jelle — jelle@jelle-VMware20-1: ~ — zsh (kiri-cli-term) » ssh — 80x24
~ zsh (kiri-cli-term) » zsh ... ...e20-1: ~ — zsh (kiri-cli-term) » ssh

The authenticity of host '192.168.3.139 (192.168.3.139)' can't be established.
ED25519 key fingerprint is SHA256:b+YMXdDjvR27NNwht0ZK/K/D2iqFtU23P6lSJ0TT1A4.
This key is not known by any other names.
[Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.3.139' (ED25519) to the list of known hosts.
jelle@192.168.3.139's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-37-generic aarch64)

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

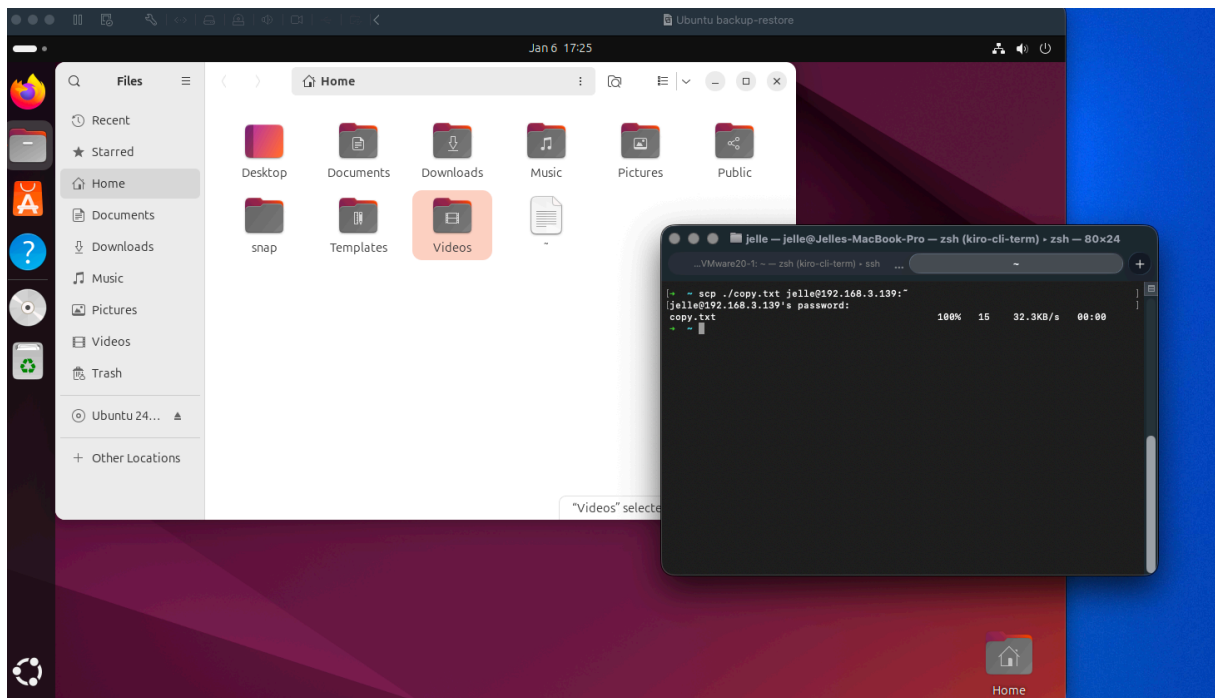
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

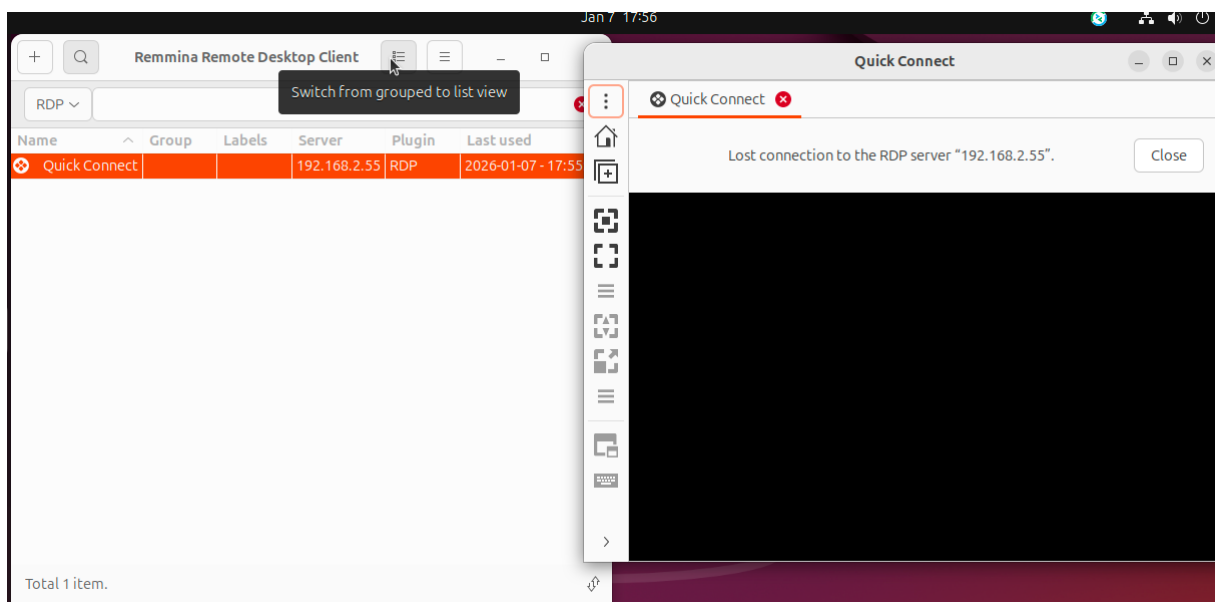
Last login: Tue Jan  6 12:17:20 2026 from 192.168.3.137
jelle@jelle-VMware20-1:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos snap
jelle@jelle-VMware20-1:~$
```

Screenshot successful execution SCP command:



Screenshot remmina:

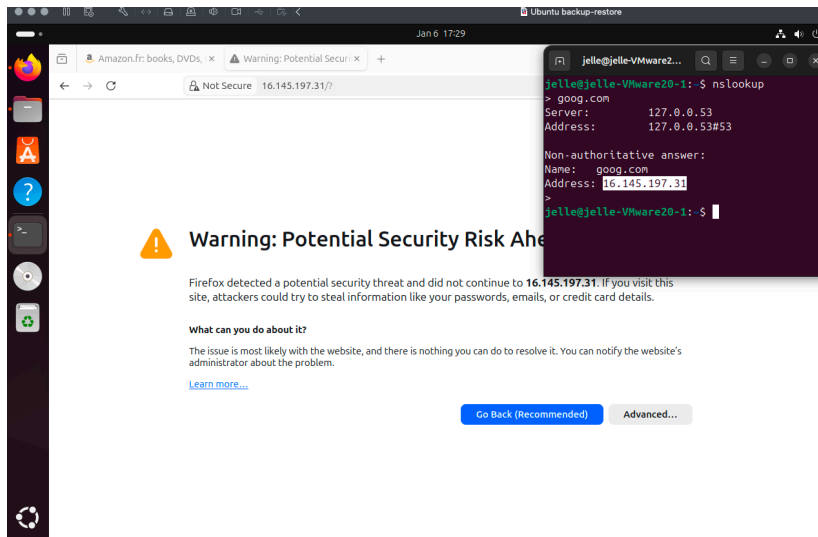
Na lang proberen kreeg ik het niet werkend helaas, aangezien dit anders werkt bij mac.



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?

/25 betekend: $32 - 25 = 7$ en 2 tot de macht $7 = 128$, dus er kunnen 128 IP-adressen erop

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

192.168.110.128 tot 192.168.110.255 maar je moet ook altijd 1 netwerk en broadcast ipaddress hebben dus eigenlijk zijn het er 126 met een range van: 192.168.110.129 – 192.168.110.254

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

```
[jelle@jelle-VMware20-1:~$ 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 11111111
=>
Network: 192.168.110.128/25 11000000.10101000.01101110.1 00000000
HostMin: 192.168.110.129 11000000.10101000.01101110.1 00000001
HostMax: 192.168.110.254 11000000.10101000.01101110.1 11111110
Broadcast: 192.168.110.255 11000000.10101000.01101110.1 11111111
Hosts/Net: 126 Class C, Private Internet
```

Explain the above calculation in your own words.

Address: hexadecimal naar binary

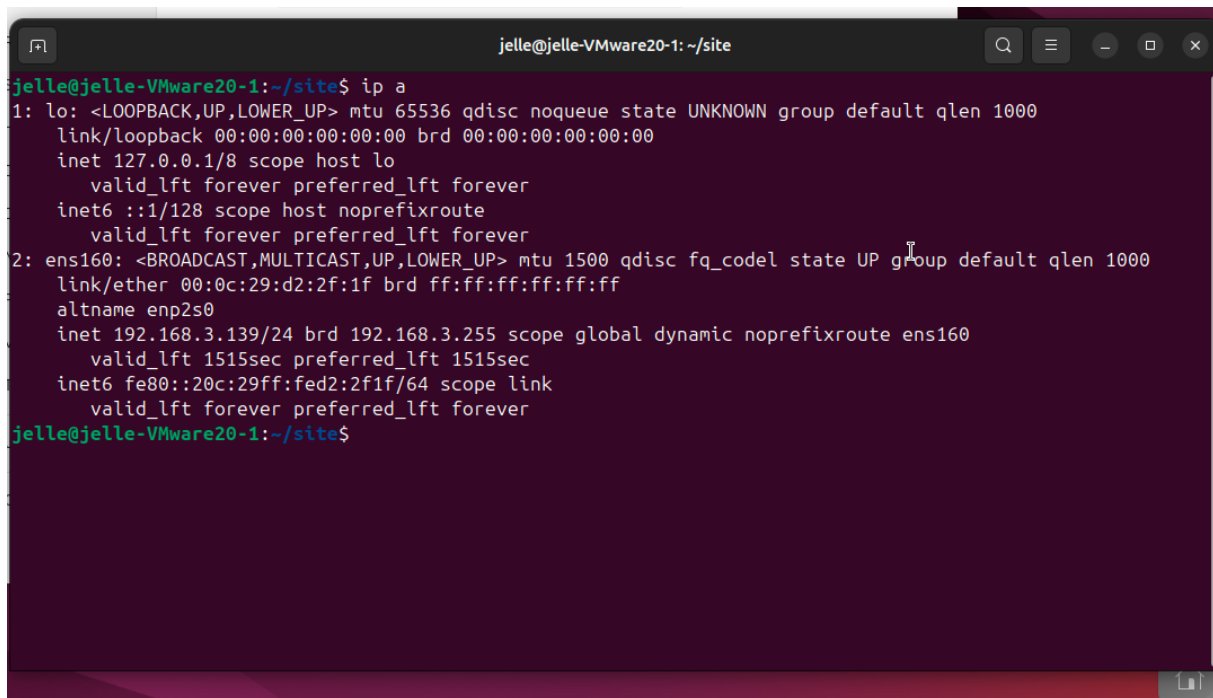
Netmask: hexadecimal naar binary, en de laatste stuk is 0, die zijn bruikbaar, verder omdat die allemaal bruikbaar zijn kan je ook de Host/ Network verklaren aangezien hij in klasse C zit. Aangezien je alleen de laatste stuk mag gebruiken.

Verder hoe er bij de network bij bijgekomen is door een OR te gebruiken op de Address en de Netmask.

Host/Net: aantal bruikbare ip adressen.

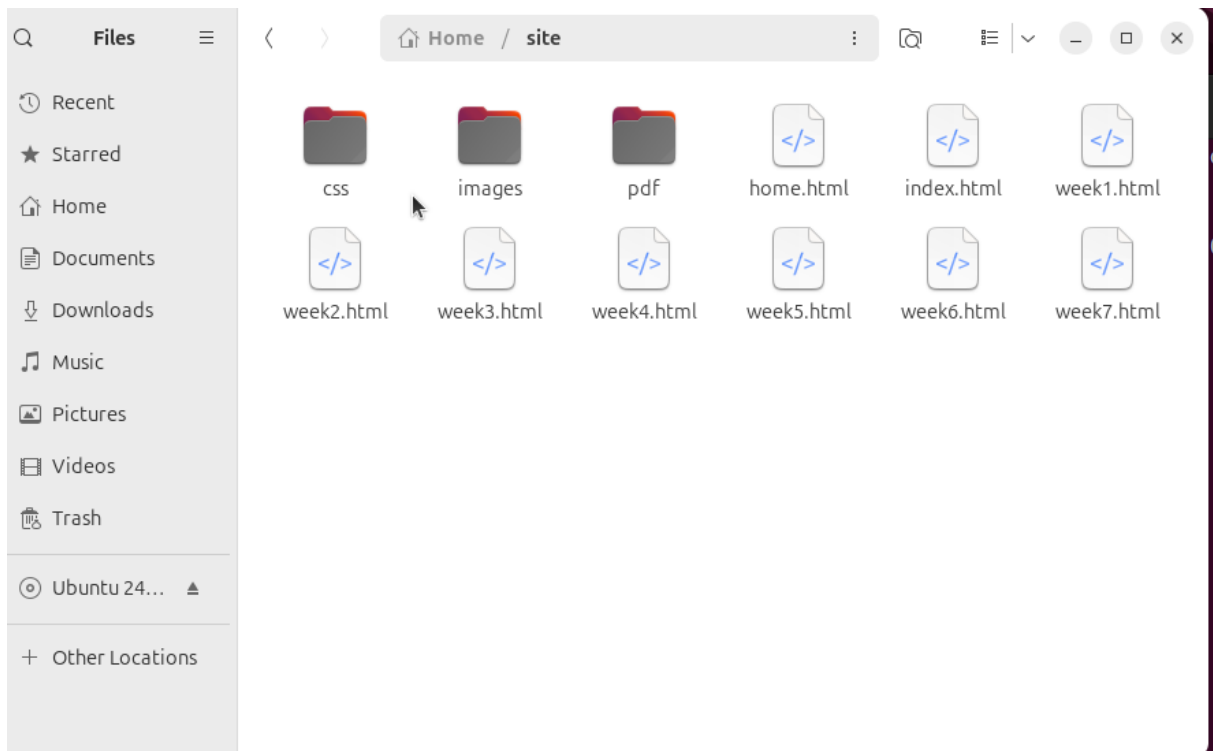
Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:



```
jelle@jelle-VMware20-1: ~/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: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:d2:2f:1f brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    inet 192.168.3.139/24 brd 192.168.3.255 scope global dynamic noprefixroute ens160
        valid_lft 1515sec preferred_lft 1515sec
    inet6 fe80::20c:29ff:fed2:2f1f/64 scope link
        valid_lft forever preferred_lft forever
jelle@jelle-VMware20-1: ~/site$
```

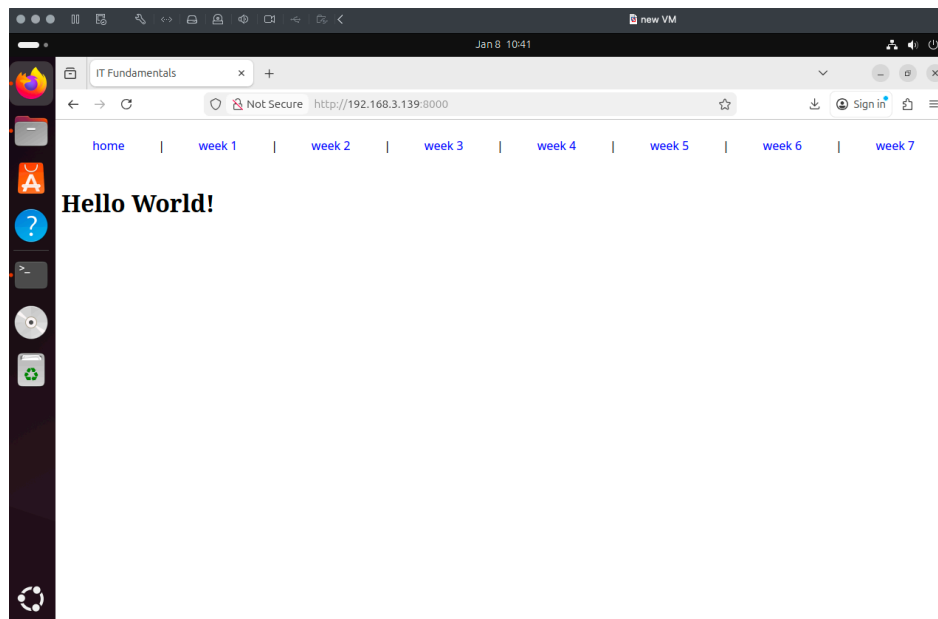
Screenshot of Site directory contents:



Screenshot python3 webserver command:

```
jelle@jelle-VMware20-i:~/site$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [08/Jan/2026 10:40:50] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [08/Jan/2026 10:40:50] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
127.0.0.1 - - [08/Jan/2026 10:40:50] "GET /home.html HTTP/1.1" 200 -
127.0.0.1 - - [08/Jan/2026 10:40:50] code 404, message File not found
127.0.0.1 - - [08/Jan/2026 10:40:50] "GET /favicon.ico HTTP/1.1" 404 -
192.168.3.139 - - [08/Jan/2026 10:40:58] "GET / HTTP/1.1" 200 -
192.168.3.139 - - [08/Jan/2026 10:40:58] "GET /home.html HTTP/1.1" 200 -
192.168.3.139 - - [08/Jan/2026 10:40:58] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
192.168.3.139 - - [08/Jan/2026 10:40:58] code 404, message File not found
192.168.3.139 - - [08/Jan/2026 10:40:58] "GET /favicon.ico HTTP/1.1" 404 -
```

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.

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

```
Please enter an number
4
is the number odd?: false
is the power of 2?: true
Ip binary: 11000000.10101000.1.1100100.
Subnet binary: 11111111.11111111.11111111.11100000.
-----
network address: 10100001100010001100000000000001

Process finished with exit code 0
```

```

1 import java.util.Scanner;
2 import java.util.StringTokenizer;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner scn = new Scanner(System.in);
7
8         System.out.println("Please enter an number");
9         int number = scn.nextInt();
10
11         System.out.println("is the number odd?: " +
12             isNumberOdd(number));
13         System.out.println("is the power of 2?: " +
14             isNumberPowerOfTwo(number));
15
16         calcNetworkAddress();
17     }
18
19     public static boolean isNumberOdd(int number) {
20         return (number & 1) == 1;
21     }
22
23     public static boolean isNumberPowerOfTwo(int number) {
24         return (number & (number - 1)) == 0;
25     }
26
27     public static void calcNetworkAddress() {
28         String ipString = "192.168.1.100";
29         //255.255.255.224
30         String subnetString = "255.255.255.224";
31
32         // rewrite the sting, other no longer useful cleaner this
33         way.
34         ipString = stringToBinair(ipString);
35         subnetString = stringToBinair(subnetString);
36         System.out.println("Ip binary: " + ipString);
37         System.out.println("Subnet binary: " + subnetString);
38         System.out.println("-----");
39
40         String cmpResult = compareBinaryBitwise(ipString,
41             subnetString);
42         System.out.println("network address: " + cmpResult);
43     }
44
45     public static String stringToBinair(String str){
46         StringTokenizer tokenizer = new StringTokenizer(str, ".");
47         StringBuilder binairysting = new StringBuilder();
48         while (tokenizer.hasMoreTokens()){
49             String octetStr = tokenizer.nextToken();
50             int octetInt = Integer.parseInt(octetStr);
51
52             // build string with .
53             binairysting.append(Integer.toBinaryString(octetInt));
54             binairysting.append(".");
55         }
56         return binairysting.toString();
57     }
58
59     public static String compareBinaryBitwise(String str1, String
60         str2){
61         int amountOfTimes = 3; // 4
62         int count;
63
64         // split the sting based on .
65         String[] str1Split = str1.split("\\.");
66         String[] str2Split = str2.split("\\.");
67
68         StringBuilder result = new StringBuilder();
69         // for each octet do this:
70         for (count = 0; count <= amountOfTimes; count++){
71             int a = Integer.parseInt(str1Split[count]);
72             int b = Integer.parseInt(str2Split[count]);
73             int andResult = a & b;
74
75             // convert to 8 bit binary with 0's
76             String binary = String.format("%8s",
77                 Integer.toBinaryString(andResult)).replace(' ', '0');
78             result.append(binary);
79             count++;
80         }
81         return result.toString();
82     }
83 }

```