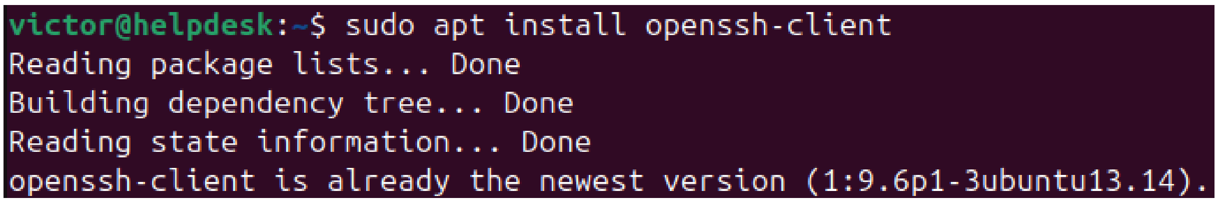
Template Week 6 – Networking

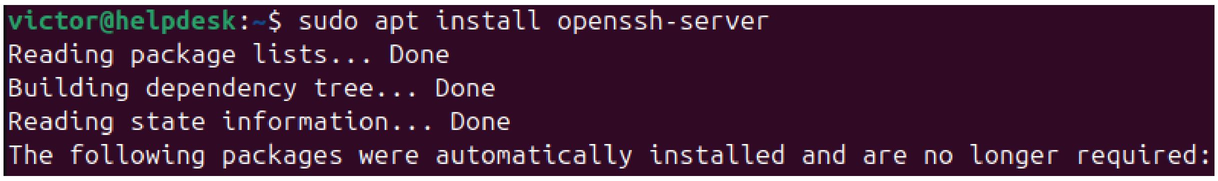
Student number: 588421

Assignment 6.1: Working from home

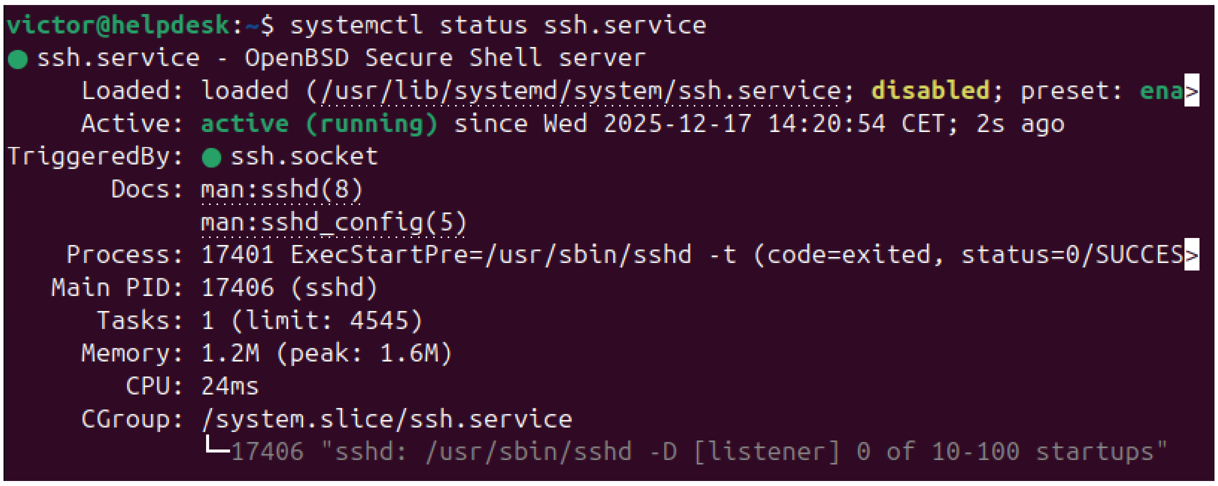
Screenshot installation openssh-server:



```
victor@helpdesk:~$ sudo apt install openssh-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-client is already the newest version (1:9.6p1-3ubuntu13.14).
```

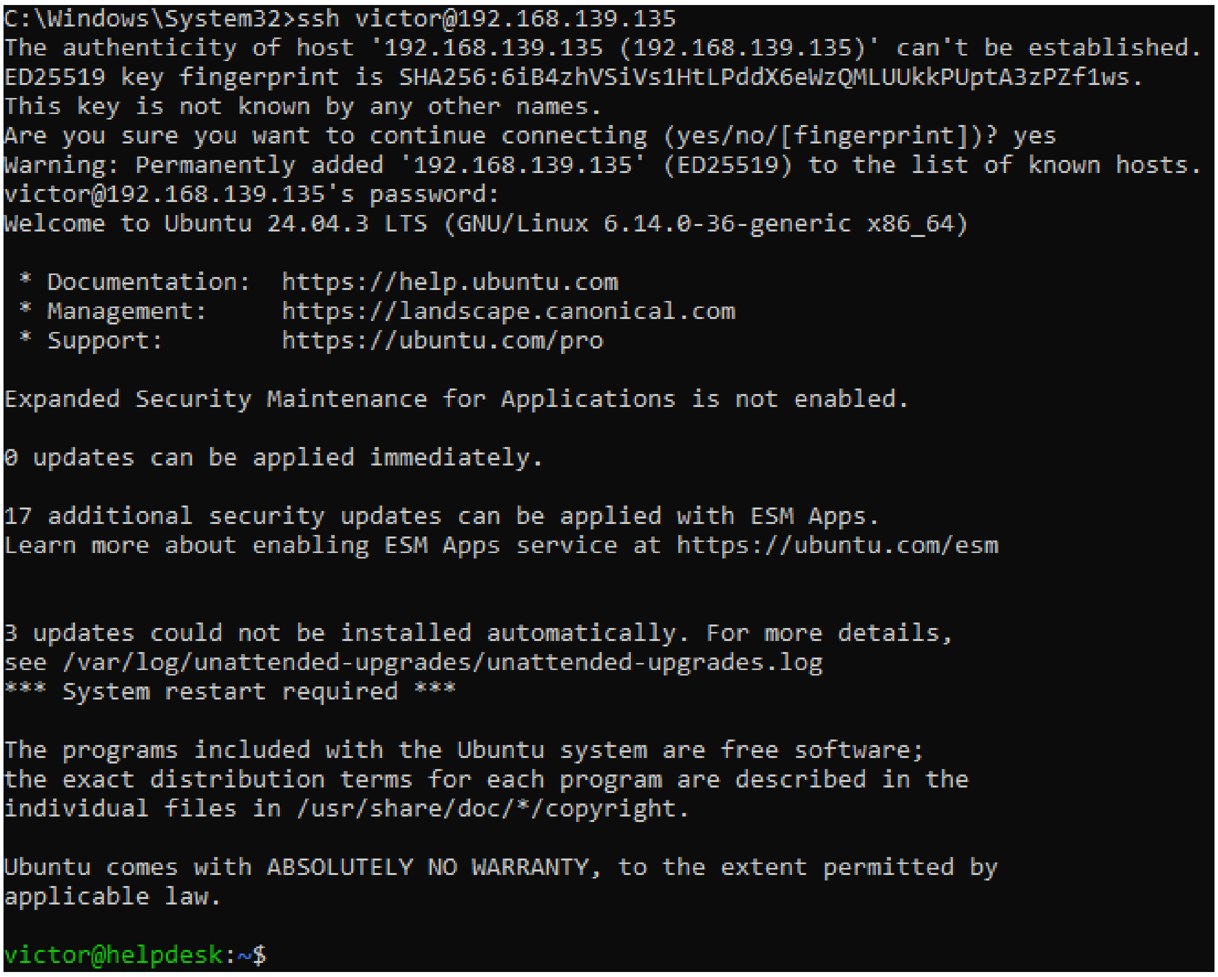


```
victor@helpdesk:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
```



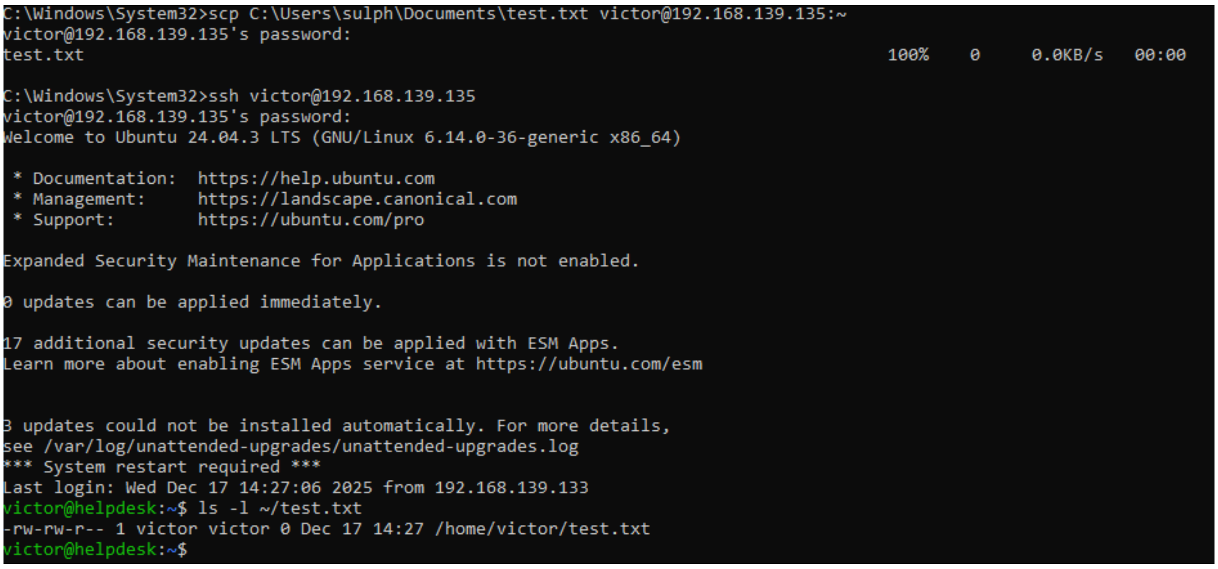
```
victor@helpdesk:~$ systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: ena>
     Active: active (running) since Wed 2025-12-17 14:20:54 CET; 2s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 17401 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCES>
   Main PID: 17406 (sshd)
      Tasks: 1 (limit: 4545)
     Memory: 1.2M (peak: 1.6M)
        CPU: 24ms
     CGroup: /system.slice/ssh.service
             └─17406 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

Screenshot successful SSH command execution:



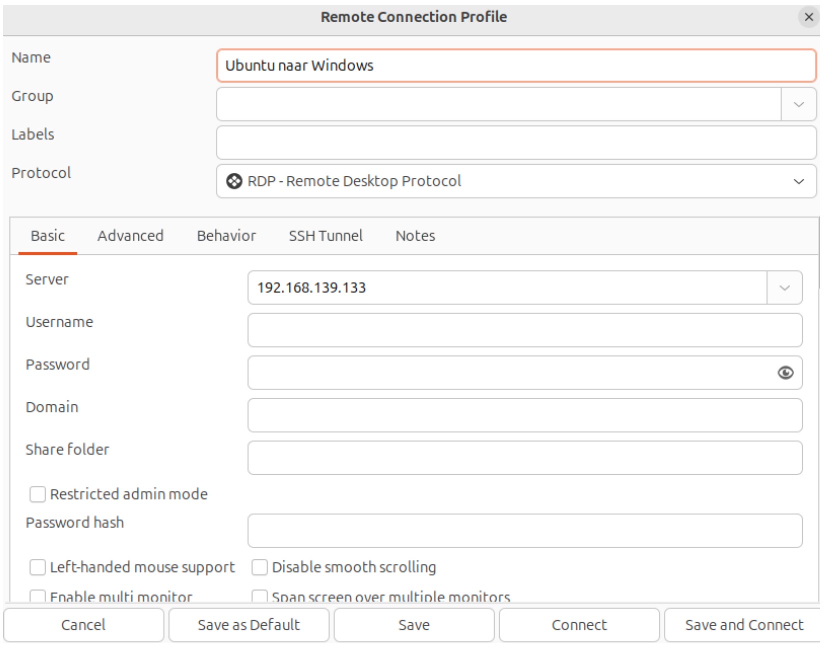
```
C:\Windows\System32>ssh victor@192.168.139.135
The authenticity of host '192.168.139.135 (192.168.139.135)' can't be established.
ED25519 key fingerprint is SHA256:6iB4zhVSiVs1HtLPddX6eWzQMLUUkkPUptA3zPZf1ws.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.139.135' (ED25519) to the list of known hosts.
victor@192.168.139.135's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-36-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


3 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log
*** System restart required ***

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.

victor@helpdesk:~$
```

Screenshot successful execution SCP command:



```
C:\Windows\System32>scp C:\Users\sulph\Documents\test.txt victor@192.168.139.135:~
victor@192.168.139.135's password:
test.txt                                                                    100%    0     0.0KB/s   00:00

C:\Windows\System32>ssh victor@192.168.139.135
victor@192.168.139.135's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-36-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


3 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log
*** System restart required ***
Last login: Wed Dec 17 14:27:06 2025 from 192.168.139.133
victor@helpdesk:~$ ls -l ~/test.txt
-rw-rw-r-- 1 victor victor 0 Dec 17 14:27 /home/victor/test.txt
victor@helpdesk:~$
```

Screenshot remmina:



Remote Connection Profile

Name	Ubuntu naar Windows
Group	
Labels	
Protocol	RDP - Remote Desktop Protocol

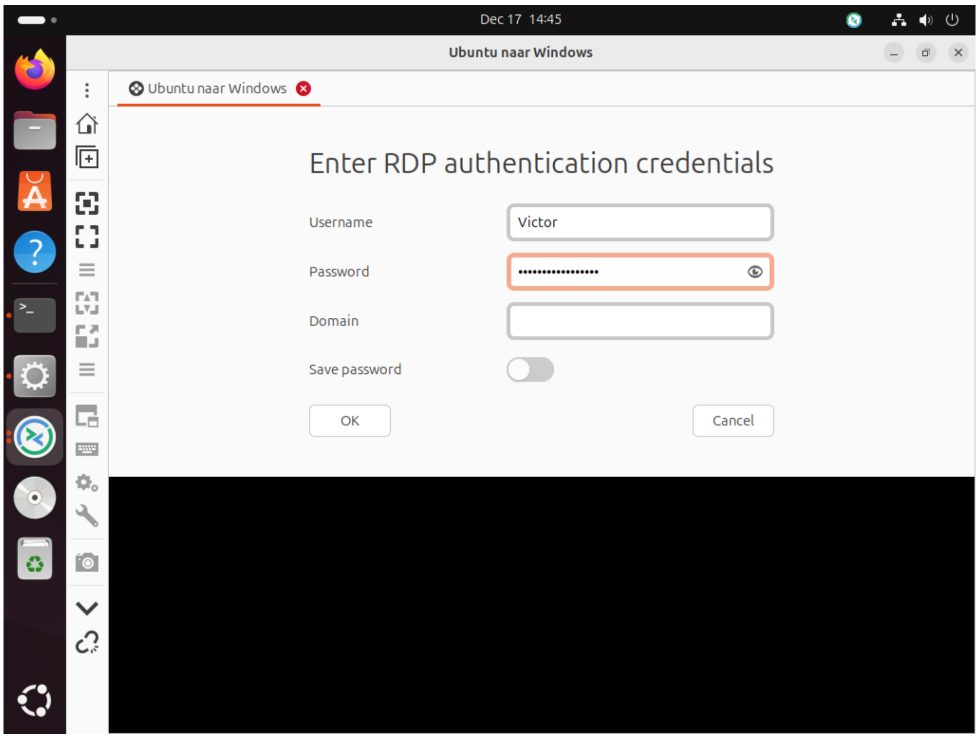
Server	192.168.139.133
Username	
Password	
Domain	
Share folder	

☐ Restricted admin mode

Password hash

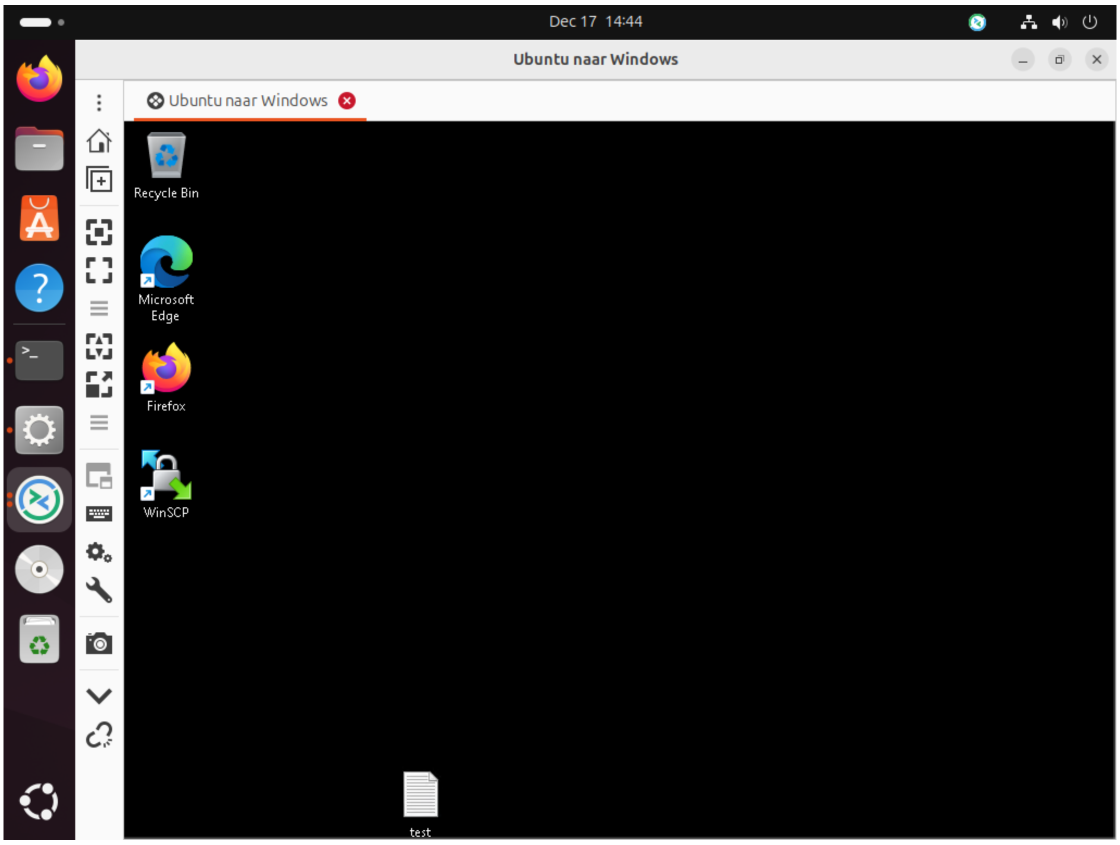
☐ Left-handed mouse support ☐ Disable smooth scrolling

☐ Enable multi monitor ☐ Span screen over multiple monitors



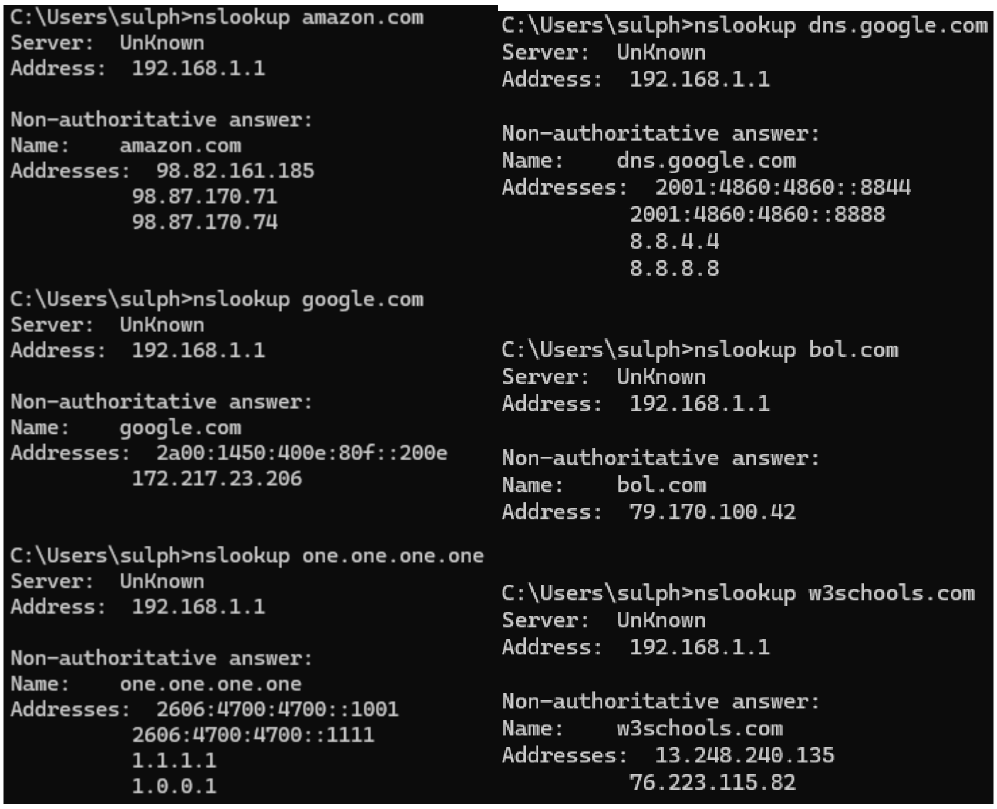
Enter RDP authentication credentials

Username	Victor
Password	••••••••••••••••
Domain	
Save password	



Assignment 6.2: IP addresses websites

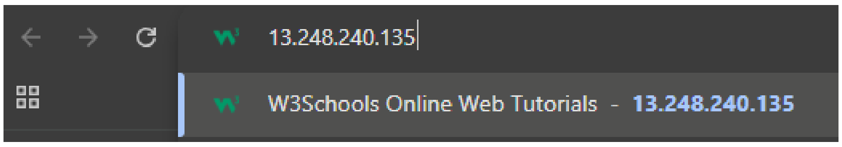
Relevant screenshots nslookup command:



```
C:\Users\sulph>nslookup amazon.com
Server:  UnKnown
Address:  192.168.1.1

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


C:\Users\sulph>nslookup google.com
Server:  UnKnown
Address:  192.168.1.1

Non-authoritative answer:
Name:    google.com
Addresses:  2a00:1450:400e:80f::200e
          172.217.23.206


C:\Users\sulph>nslookup one.one.one.one
Server:  UnKnown
Address:  192.168.1.1

Non-authoritative answer:
Name:    one.one.one.one
Addresses:  2606:4700:4700::1001
          2606:4700:4700::1111
          1.1.1.1
          1.0.0.1
```

```
C:\Users\sulph>nslookup dns.google.com
Server:  UnKnown
Address:  192.168.1.1

Non-authoritative answer:
Name:    dns.google.com
Addresses:  2001:4860:4860::8844
          2001:4860:4860::8888
          8.8.4.4
          8.8.8.8


C:\Users\sulph>nslookup bol.com
Server:  UnKnown
Address:  192.168.1.1

Non-authoritative answer:
Name:    bol.com
Address:  79.170.100.42


C:\Users\sulph>nslookup w3schools.com
Server:  UnKnown
Address:  192.168.1.1

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

Screenshot website visit via IP address:

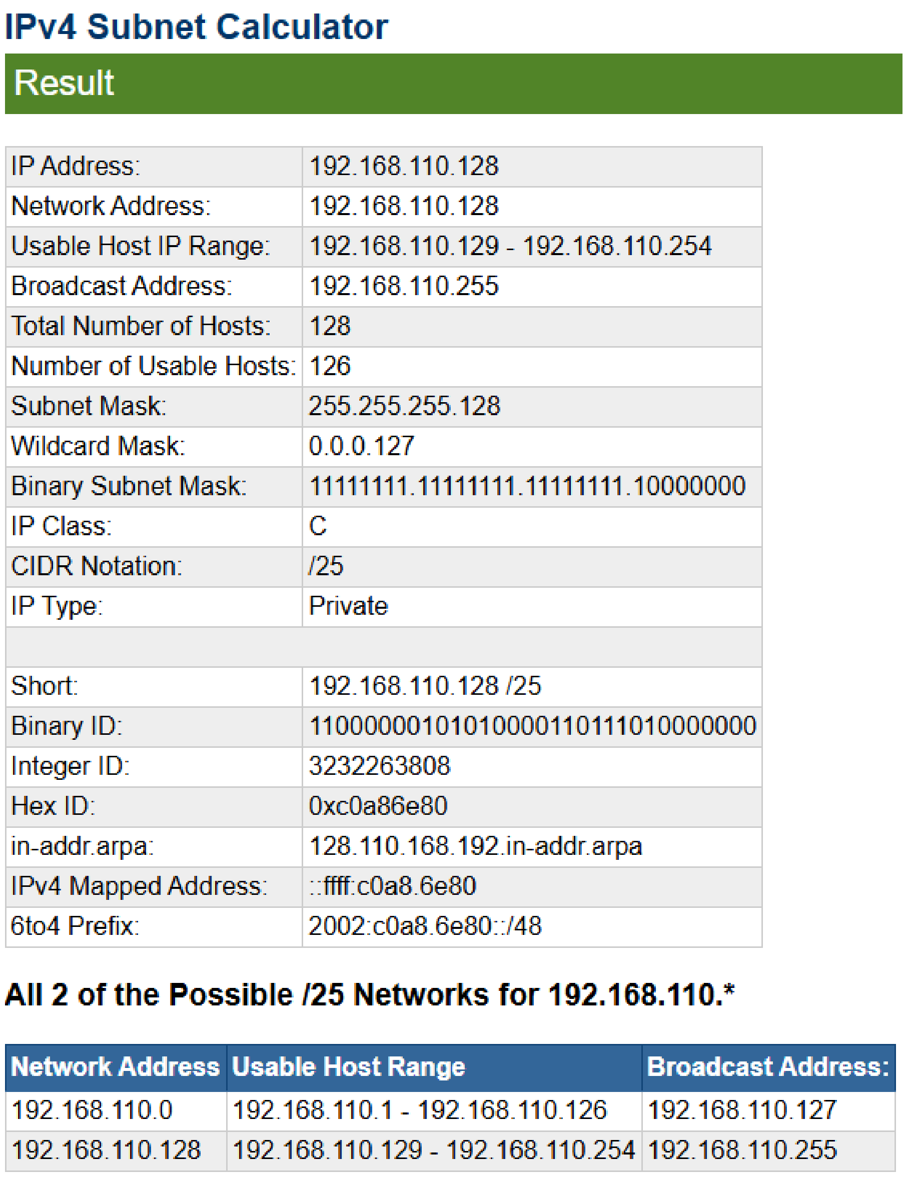


13.248.240.135

W3Schools Online Web Tutorials - 13.248.240.135

Assignment 6.3: subnetting

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



IPv4 Subnet Calculator

Result

IP Address:	192.168.110.128
Network Address:	192.168.110.128
Usable Host IP Range:	192.168.110.129 - 192.168.110.254
Broadcast Address:	192.168.110.255
Total Number of Hosts:	128
Number of Usable Hosts:	126
Subnet Mask:	255.255.255.128
Wildcard Mask:	0.0.0.127
Binary Subnet Mask:	11111111.11111111.11111111.10000000
IP Class:	C
CIDR Notation:	/25
IP Type:	Private
Short:	192.168.110.128 /25
Binary ID:	11000000101010000110111010000000
Integer ID:	3232263808
Hex ID:	0xc0a86e80
in-addr.arpa:	128.110.168.192.in-addr.arpa
IPv4 Mapped Address:	::ffff:c0a8.6e80
6to4 Prefix:	2002:c0a8.6e80::/48

All 2 of the Possible /25 Networks for 192.168.110.*

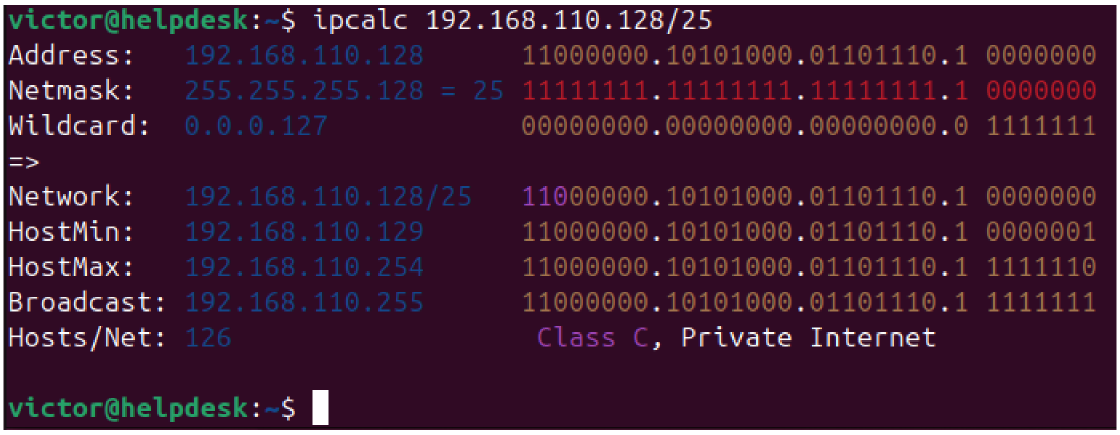
Network Address	Usable Host Range	Broadcast Address:
192.168.110.0	192.168.110.1 - 192.168.110.126	192.168.110.127
192.168.110.128	192.168.110.129 - 192.168.110.254	192.168.110.255

126 bruikbare hosts

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

192.168.110.129 – 192.168.110.254

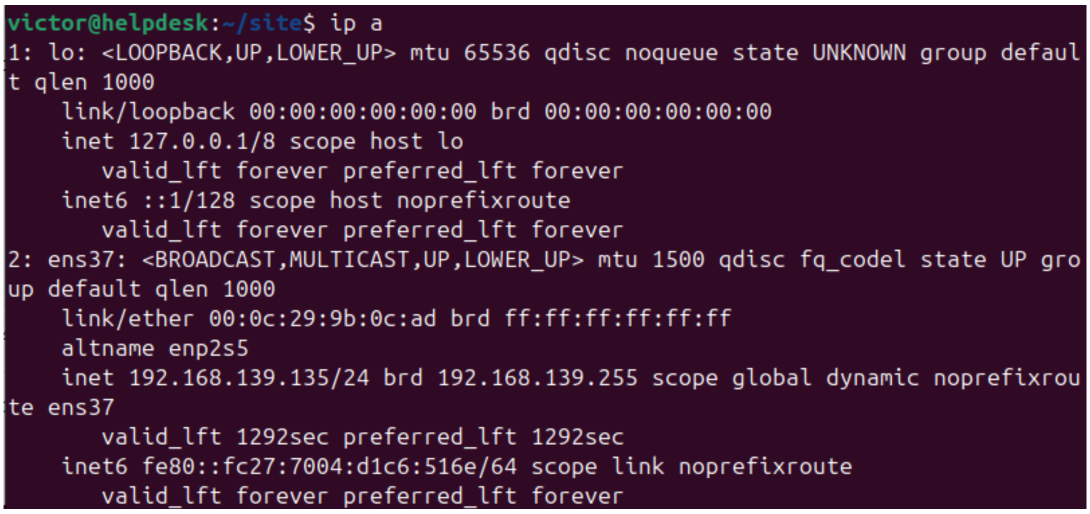
Check your two previous answers with this Linux command: `ipcalc 192.168.110.128/25`


```
victor@helpdesk:~$ 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

victor@helpdesk:~$
```

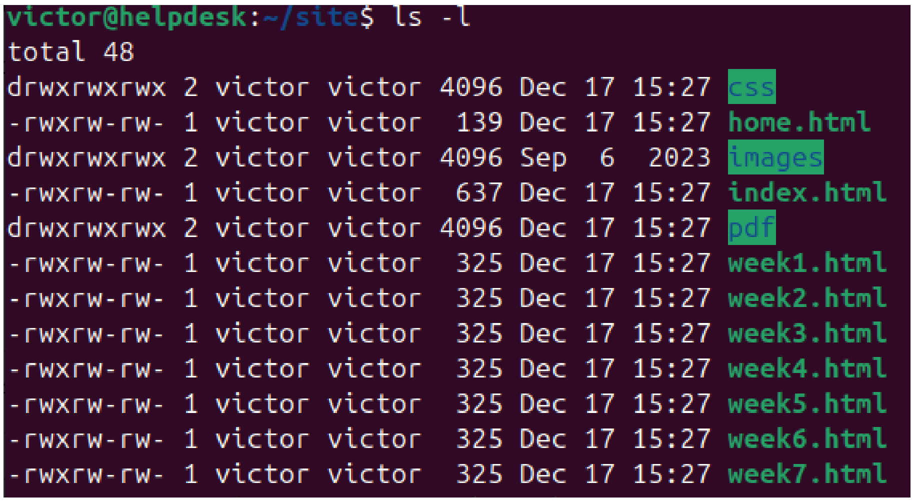
Explain the above calculation in your own words.

Assignment 6.4: HTML

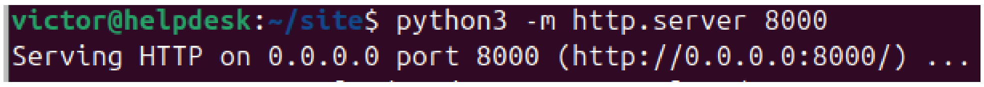
Screenshot IP address Ubuntu VM:


```
victor@helpdesk:~/site$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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: ens37: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gro
up default qlen 1000
    link/ether 00:0c:29:9b:0c:ad brd ff:ff:ff:ff:ff:ff
    altname enp2s5
    inet 192.168.139.135/24 brd 192.168.139.255 scope global dynamic noprefixrou
te ens37
       valid_lft 1292sec preferred_lft 1292sec
    inet6 fe80::fc27:7004:d1c6:516e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Screenshot of Site directory contents:

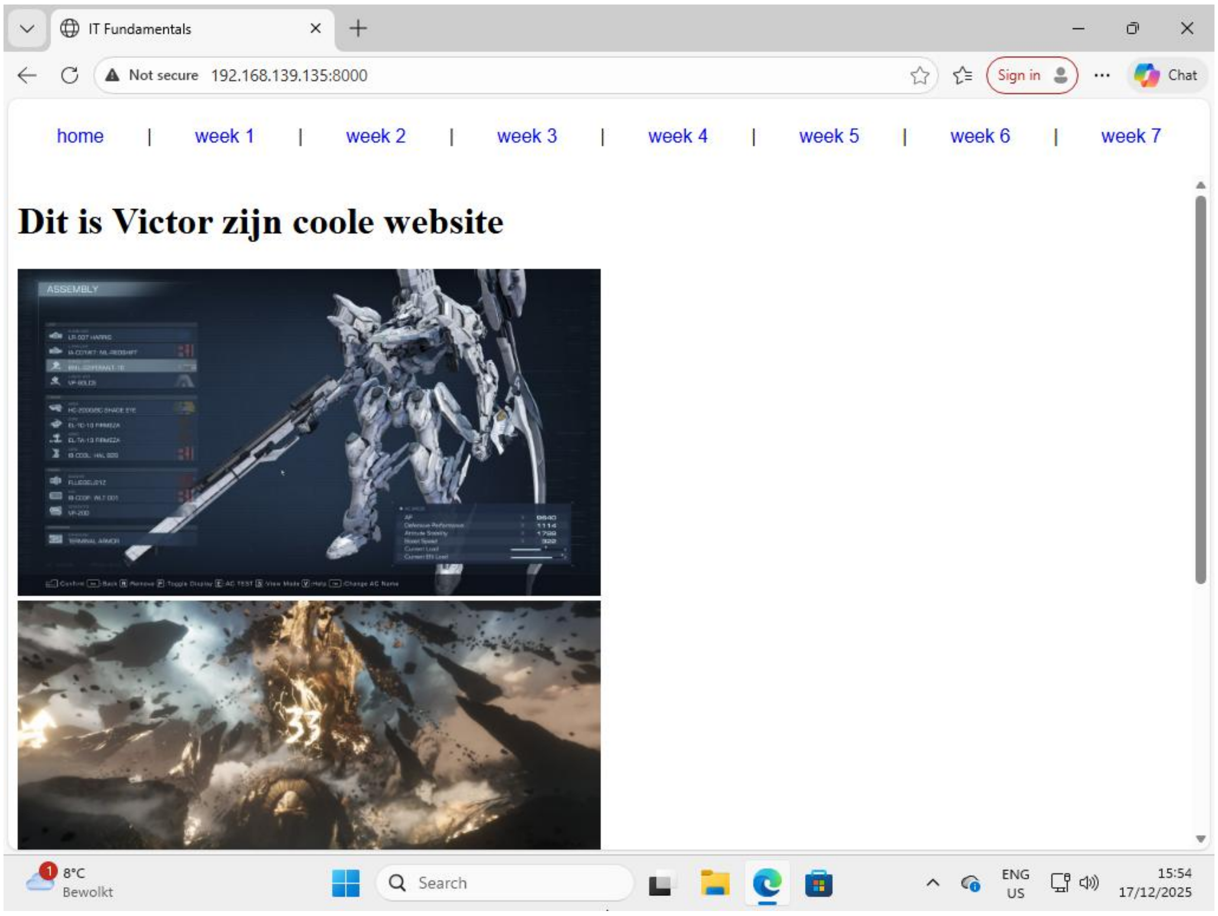

```
victor@helpdesk:~/site$ ls -l
total 48
drwxrwxrwx 2 victor victor 4096 Dec 17 15:27 css
-rwxrw-rw- 1 victor victor  139 Dec 17 15:27 home.html
drwxrwxrwx 2 victor victor 4096 Sep  6  2023 images
-rwxrw-rw- 1 victor victor  637 Dec 17 15:27 index.html
drwxrwxrwx 2 victor victor 4096 Dec 17 15:27 pdf
-rwxrw-rw- 1 victor victor  325 Dec 17 15:27 week1.html
-rwxrw-rw- 1 victor victor  325 Dec 17 15:27 week2.html
-rwxrw-rw- 1 victor victor  325 Dec 17 15:27 week3.html
-rwxrw-rw- 1 victor victor  325 Dec 17 15:27 week4.html
-rwxrw-rw- 1 victor victor  325 Dec 17 15:27 week5.html
-rwxrw-rw- 1 victor victor  325 Dec 17 15:27 week6.html
-rwxrw-rw- 1 victor victor  325 Dec 17 15:27 week7.html
```

Screenshot python3 webserver command:



```
victor@helpdesk:~/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


IT Fundamentals
Not secure 192.168.139.135:8000
Sign in
Chat
home | week 1 | week 2 | week 3 | week 4 | week 5 | week 6 | week 7
8°C
Bewolkt
Search
ENG US
15:54
17/12/2025

Dit is Victor zijn coole website

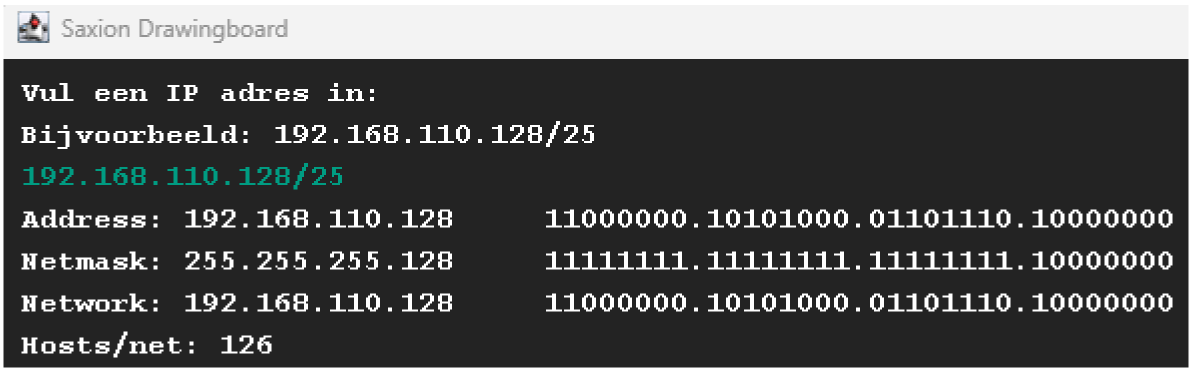
Assignment 6.5: Network segment

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.


Saxion Drawingboard

```
Vul een IP adres in:
Bijvoorbeeld: 192.168.110.128/25
192.168.110.128/25
Address: 192.168.110.128     11000000.10101000.01101110.10000000
Netmask: 255.255.255.128     11111111.11111111.11111111.10000000
Network: 192.168.110.128     11000000.10101000.01101110.10000000
Hosts/net: 126
```

```
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() {
      calculateNetworkSegment();
   }

   public void calculateNetworkSegment(){
      SaxionApp.printLine("Vul een IP adres in:");
      SaxionApp.printLine("Bijvoorbeeld: 192.168.110.128/25");
      String ipAddress = SaxionApp.readString();
      //String ipAddress = "192.168.110.128/25";
      String regex = "[.\\/]";
      int[] ipAddressParts = new int[5];
      byte[] bytesIp = new byte[4];
      byte[] bytesNetmask = new byte[4];
      byte[] networkAddress = new byte[4];
      int[] subnetParts = new int[4];
      int[] mask = {0, 128, 192, 224, 240, 248, 252, 254, 255};
      String ipAddressBinary = "";
      String ipAddressOnly = "";
      String subnetBinary = "";
      String subnetOnly = "";
      String networkBinary = "";
      String networkOnly = "";
      String[] splitIpaddress = ipAddress.split(regex);
      for(int i = 0; i < splitIpaddress.length; i++){
         ipAddressParts[i] = Integer.parseInt(splitIpaddress[i]);
```

```
        }
        int subnetTeller = ipAddressParts[4];
        //Gebruik byte array
        //255<<(8-n)
        for(int i = 0; i < 4; i++){
            //SaxionApp.printLine(subnetTeller);
            subnetParts[i] = Math.min(8, subnetTeller);
            subnetTeller -= subnetParts[i];
            bytesNetmask[i] = (byte) mask[subnetParts[i]];
        }

//      for(int part : subnetParts){
//          SaxionApp.printLine(part);
//      }

        for(int i = 0; i < ipAddressParts.length-1; i++){
        //for(int part : ipAddressParts){
            //SaxionApp.printLine(ipAddressParts[i]);
            byte binaryByte = (byte) ipAddressParts[i];
            bytesIp[i] = binaryByte;
        }

        for(int i = 0; i < 4; i++){
            ipAddressOnly += (bytesIp[i] & 0xFF);
            ipAddressBinary += byteToBinary(bytesIp[i]);
            subnetOnly += (bytesNetmask[i] & 0xFF);
            subnetBinary += byteToBinary(bytesNetmask[i]);

            networkAddress[i] = (byte) (bytesIp[i] & bytesNetmask[i]);

            networkOnly += (networkAddress[i] & 0xFF);
            networkBinary += byteToBinary(networkAddress[i]);
            if(i < 3){
                ipAddressOnly += ".";
                ipAddressBinary += ".";
                subnetOnly += ".";
                subnetBinary += ".";
                networkOnly += ".";
                networkBinary += ".";
            }
        }

        double usableHosts = Math.pow(2, (32 - ipAddressParts[4])) - 2;
        SaxionApp.printLine("Address: " + ipAddressOnly + "     " + ipAddressBinary);
        SaxionApp.printLine("Netmask: " + subnetOnly + "     " + subnetBinary);
        SaxionApp.printLine("Network: " + networkOnly + "     " + networkBinary);
        SaxionApp.printLine("Hosts/net: " + (int)usableHosts);
```

```
//      byte binaryOne = (byte) 192;
//      byte binaryTwo = (byte) 255;
//      byte binaryAnswer = (byte) (binaryOne & binaryTwo);
//      SaxionApp.printLine(binaryAnswer & 0xFF);
//      SaxionApp.printLine(byteToBinary(binaryAnswer));
    }

    public String byteToBinary(byte binary){
        String stringOne = String.format("%8s", Integer.toBinaryString(binary & 0xFF)).replace(' ', '0');
        return stringOne;
    }
}
```

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