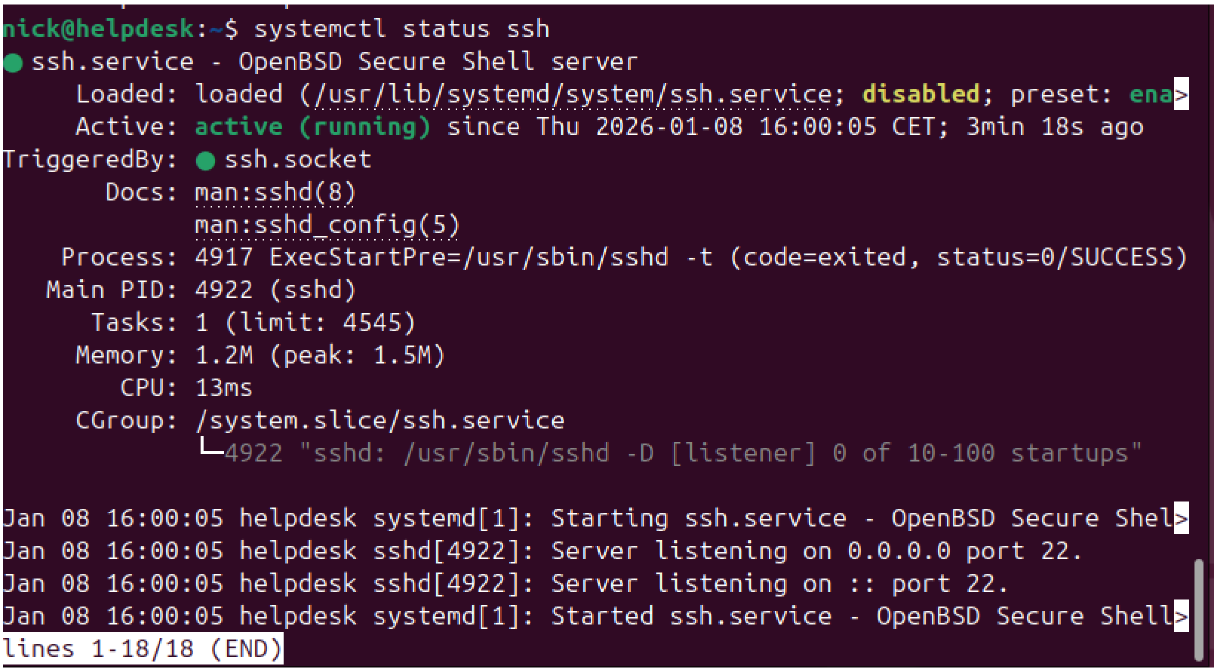
Template Week 6 – Networking

Student number:

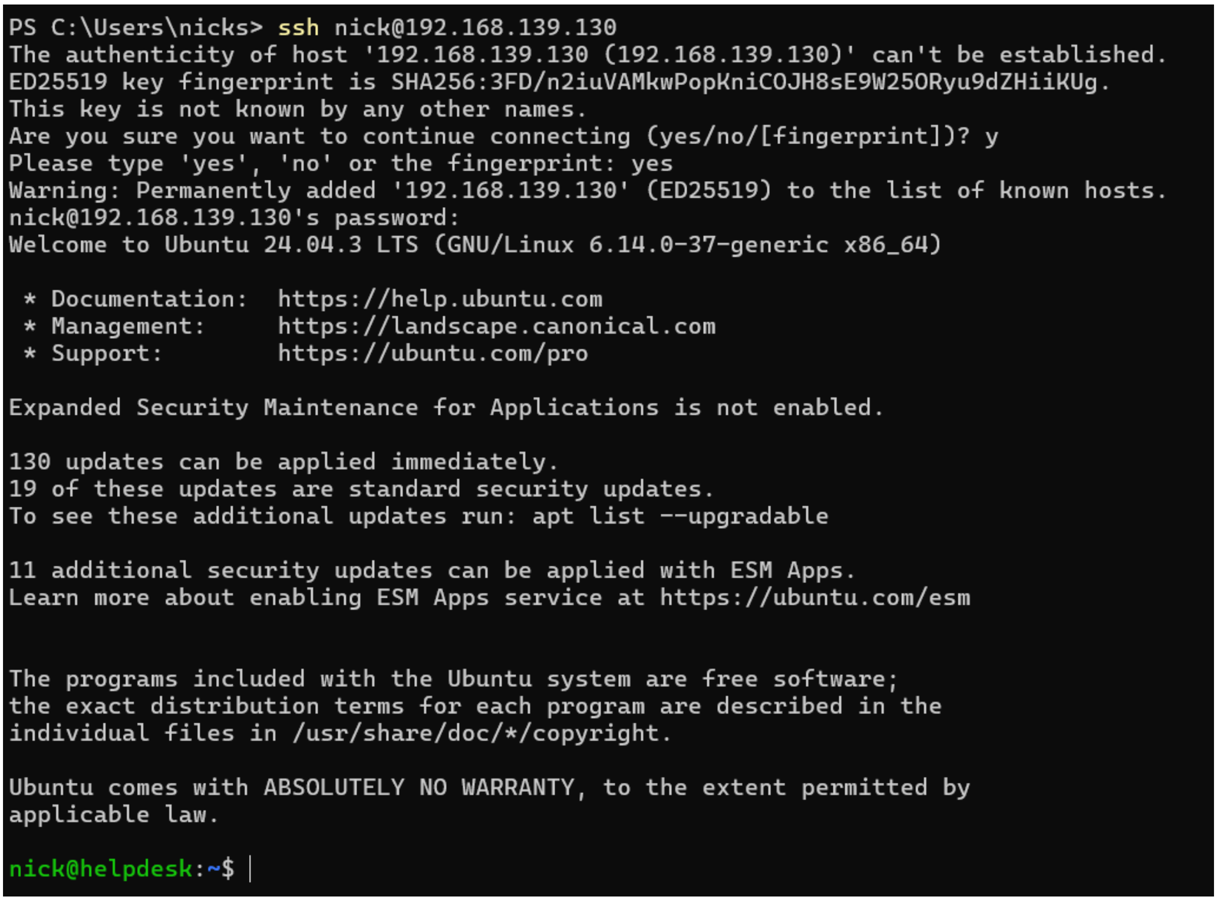
Assignment 6.1: Working from home

Screenshot installation openssh-server:



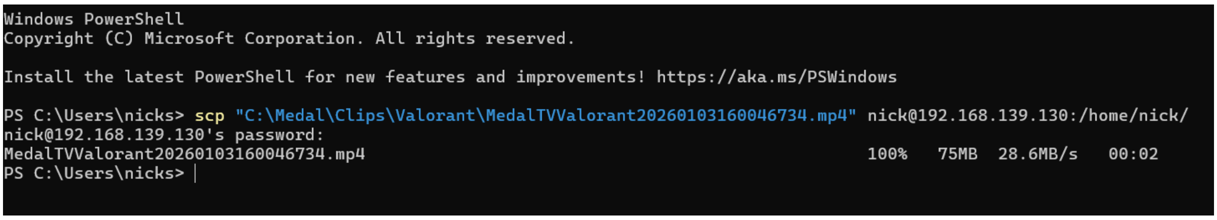
```
nick@helpdesk:~$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: ena>
     Active: active (running) since Thu 2026-01-08 16:00:05 CET; 3min 18s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 4917 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 4922 (sshd)
      Tasks: 1 (limit: 4545)
     Memory: 1.2M (peak: 1.5M)
        CPU: 13ms
     CGroup: /system.slice/ssh.service
             └─4922 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 08 16:00:05 helpdesk systemd[1]: Starting ssh.service - OpenBSD Secure Shel>
Jan 08 16:00:05 helpdesk sshd[4922]: Server listening on 0.0.0.0 port 22.
Jan 08 16:00:05 helpdesk sshd[4922]: Server listening on :: port 22.
Jan 08 16:00:05 helpdesk systemd[1]: Started ssh.service - OpenBSD Secure Shell>
lines 1-18/18 (END)
```

Screenshot successful SSH command execution:

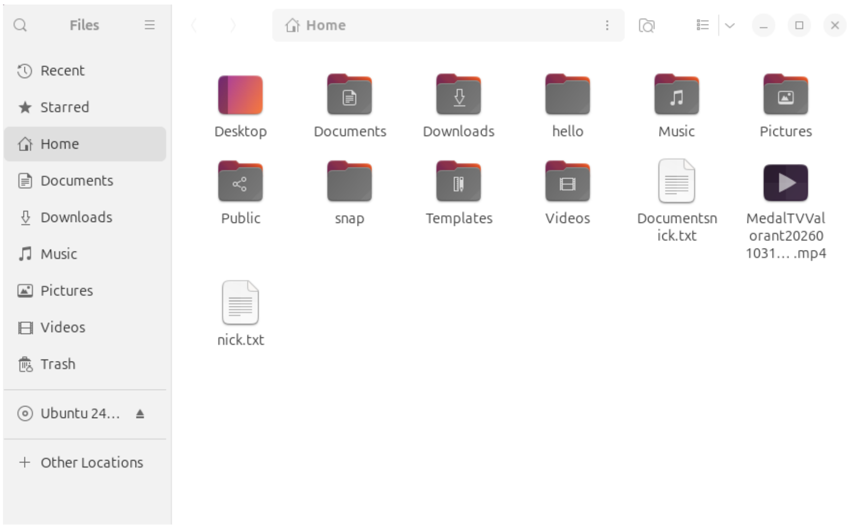


```
PS C:\Users\nicks> ssh nick@192.168.139.130
The authenticity of host '192.168.139.130 (192.168.139.130)' can't be established.
ED25519 key fingerprint is SHA256:3FD/n2iuVAMkwPopKniCOJH8sE9W25ORyu9dZHiiKUg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.139.130' (ED25519) to the list of known hosts.
nick@192.168.139.130'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.

130 updates can be applied immediately.
19 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

11 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


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.

nick@helpdesk:~$ 
```

Screenshot successful execution SCP command:



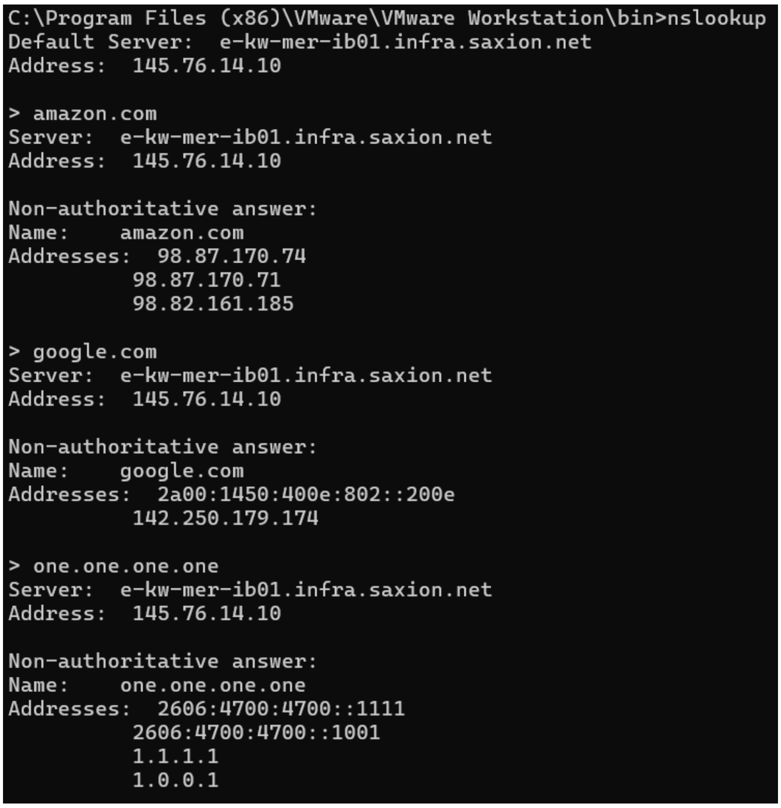
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\nicks> scp "C:\Medal\Clips\Valorant\MedalTVValorant20260103160046734.mp4" nick@192.168.139.130:/home/nick/
nick@192.168.139.130's password:
MedalTVValorant20260103160046734.mp4                                          100%   75MB  28.6MB/s   00:02
PS C:\Users\nicks> 
```



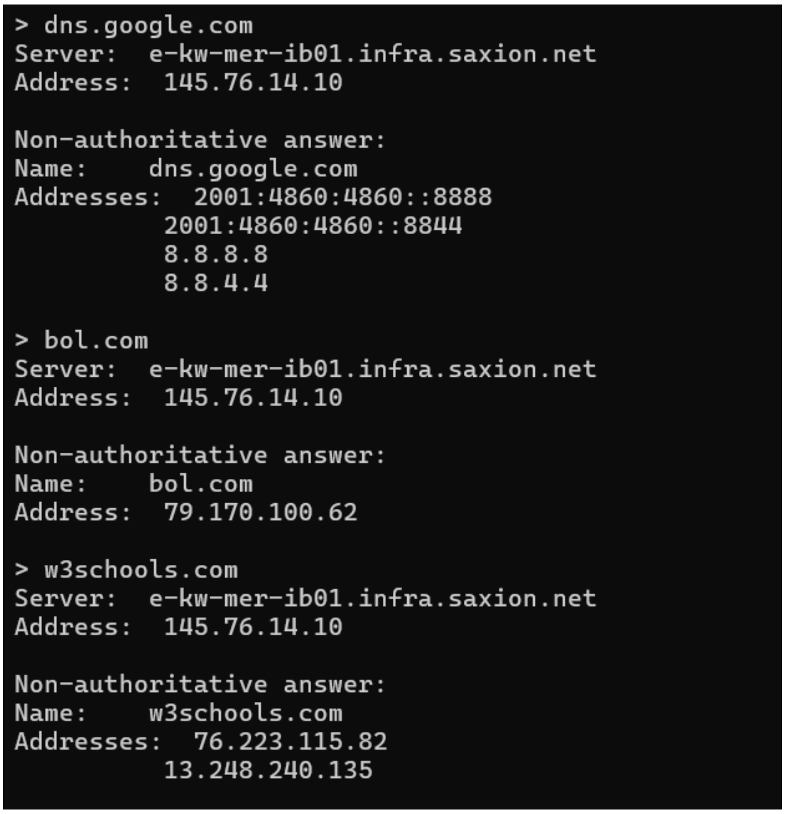
Screenshot remmina:

Assignment 6.2: IP addresses websites

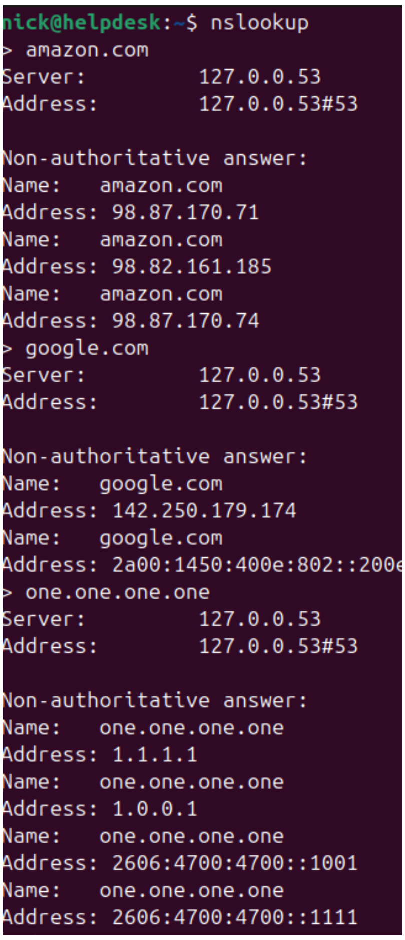
Relevant screenshots nslookup command:



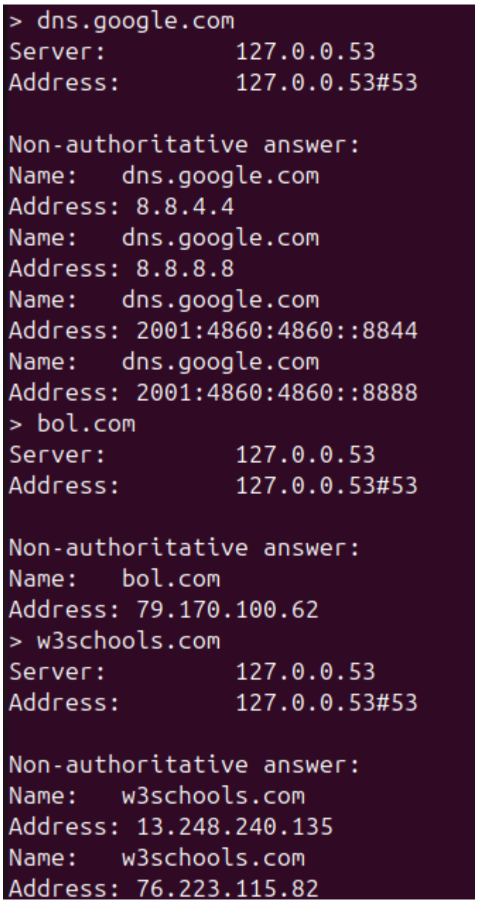
```
C:\Program Files (x86)\VMware\VMware Workstation\bin>nslookup
Default Server:  e-kw-mer-ib01.infra.saxion.net
Address:  145.76.14.10

> amazon.com
Server:  e-kw-mer-ib01.infra.saxion.net
Address:  145.76.14.10

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

> google.com
Server:  e-kw-mer-ib01.infra.saxion.net
Address:  145.76.14.10

Non-authoritative answer:
Name:    google.com
Addresses:  2a00:1450:400e:802::200e
          142.250.179.174

> one.one.one.one
Server:  e-kw-mer-ib01.infra.saxion.net
Address:  145.76.14.10

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



```
> dns.google.com
Server:  e-kw-mer-ib01.infra.saxion.net
Address:  145.76.14.10

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

> bol.com
Server:  e-kw-mer-ib01.infra.saxion.net
Address:  145.76.14.10

Non-authoritative answer:
Name:    bol.com
Address:  79.170.100.62

> w3schools.com
Server:  e-kw-mer-ib01.infra.saxion.net
Address:  145.76.14.10

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

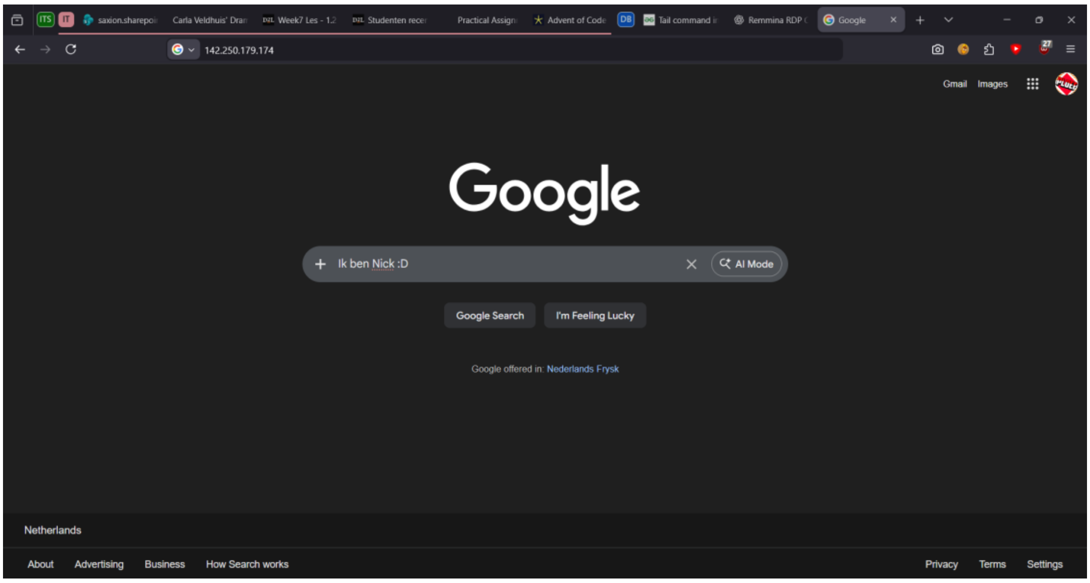


```
nick@helpdesk:~$ nslookup
> amazon.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   amazon.com
Address: 98.87.170.71
Name:   amazon.com
Address: 98.82.161.185
Name:   amazon.com
Address: 98.87.170.74
> google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.179.174
Name:   google.com
Address: 2a00:1450:400e:802::200e
> one.one.one.one
Server:         127.0.0.53
Address:        127.0.0.53#53

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



```
> dns.google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   dns.google.com
Address: 8.8.4.4
Name:   dns.google.com
Address: 8.8.8.8
Name:   dns.google.com
Address: 2001:4860:4860::8844
Name:   dns.google.com
Address: 2001:4860:4860::8888
> bol.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   bol.com
Address: 79.170.100.62
> w3schools.com
Server:         127.0.0.53
Address:        127.0.0.53#53

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

Screenshot website visit via IP address:



142.250.179.174

Ik ben Nick :D

Assignment 6.3: subnetting

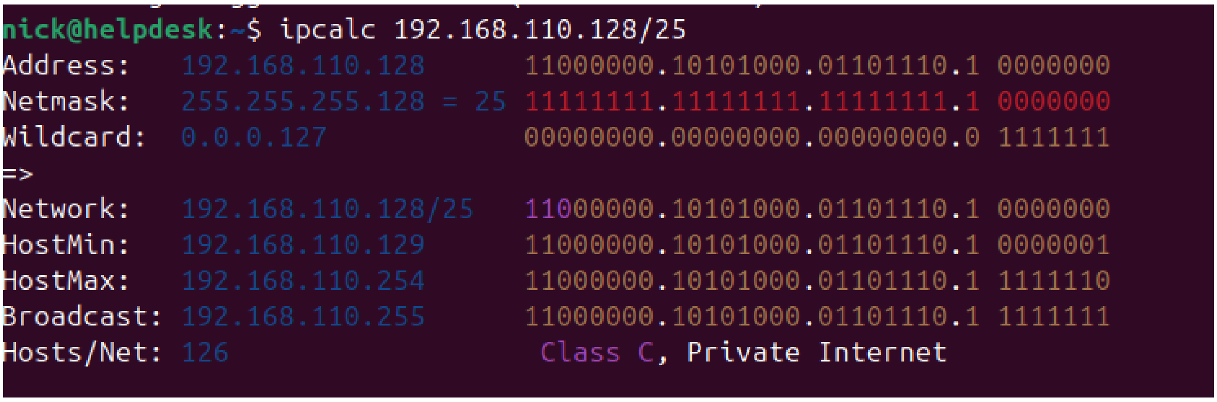
How many IP addresses are in this network configuration 192.168.110.128/25?

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`



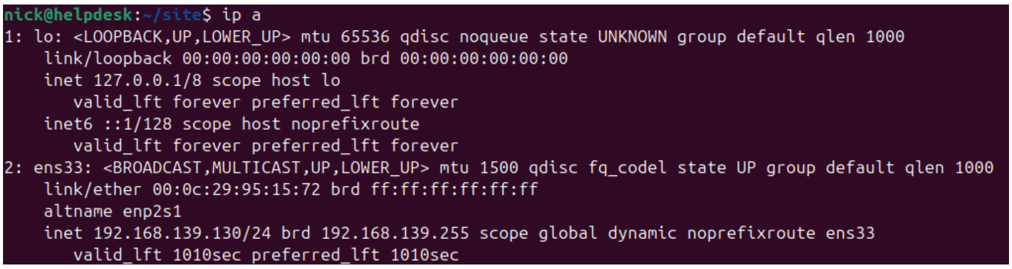
```
nick@helpdesk:~$ ipcalc 192.168.110.128/25
Address:   192.168.110.128      11000000.10101000.01101110.1 0000000
Netmask:   255.255.255.128 = 25 11111111.11111111.11111111.1 0000000
Wildcard:  0.0.0.127            00000000.00000000.00000000.0 1111111
=>
Network:   192.168.110.128/25   11000000.10101000.01101110.1 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
```

Explain the above calculation in your own words.

De subnetmask van 25 zorgt ervoor dat er 25 bits worden gebruikt voor netwerk en 7 bits voor host, en 2^7 = 128, dus 128 IP-adressen. Er worden 2 adressen gereserveerd: de laagste (128) en de hoogste (255). Omdat die 2 worden gebruikt zijn er 126 ip-adressen waarmee je kan verbinden

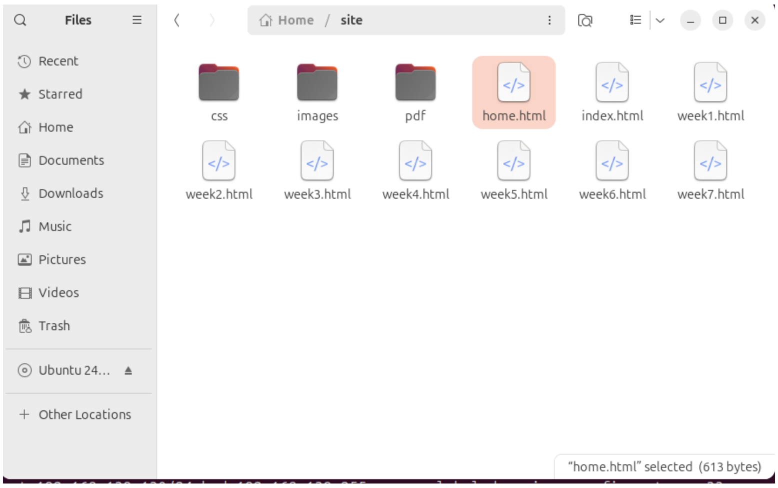
Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

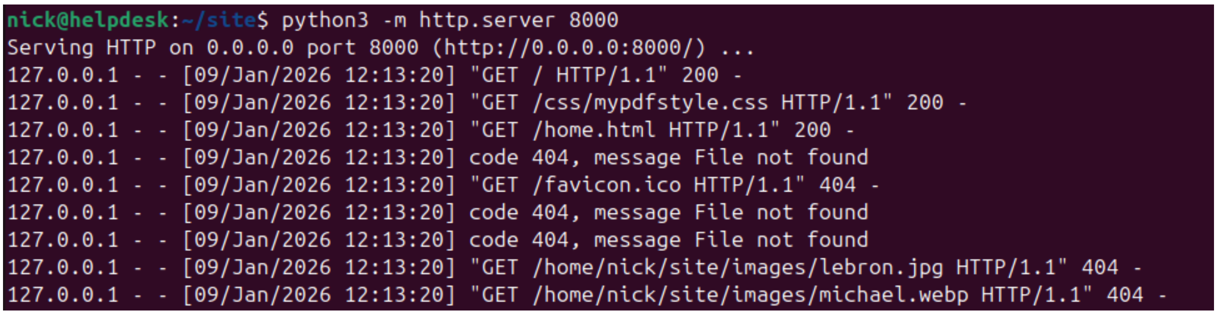


```
nick@helpdesk:~/site$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:95:15:72 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.139.130/24 brd 192.168.139.255 scope global dynamic noprefixroute ens33
       valid_lft 1010sec preferred_lft 1010sec
```

Screenshot of Site directory contents:

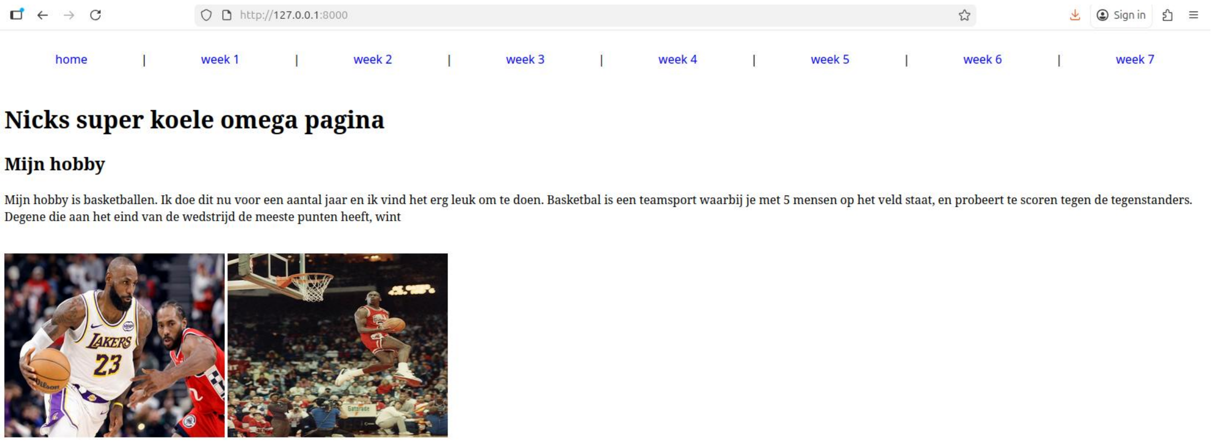


Screenshot python3 webserver command:



```
nick@helpdesk:~/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 - - [09/Jan/2026 12:13:20] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Jan/2026 12:13:20] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
127.0.0.1 - - [09/Jan/2026 12:13:20] "GET /home.html HTTP/1.1" 200 -
127.0.0.1 - - [09/Jan/2026 12:13:20] code 404, message File not found
127.0.0.1 - - [09/Jan/2026 12:13:20] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [09/Jan/2026 12:13:20] code 404, message File not found
127.0.0.1 - - [09/Jan/2026 12:13:20] code 404, message File not found
127.0.0.1 - - [09/Jan/2026 12:13:20] "GET /home/nick/site/images/lebron.jpg HTTP/1.1" 404 -
127.0.0.1 - - [09/Jan/2026 12:13:20] "GET /home/nick/site/images/michael.webp HTTP/1.1" 404 -
```

Screenshot web browser visits your site



home | week 1 | week 2 | week 3 | week 4 | week 5 | week 6 | week 7

Nicks super koele omega pagina

Mijn hobby

Mijn hobby is basketballen. Ik doe dit nu voor een aantal jaar en ik vind het erg leuk om te doen. Basketbal is een teamsport waarbij je met 5 mensen op het veld staat, en probeert te scoren tegen de tegenstanders. Degene die aan het eind van de wedstrijd de meeste punten heeft, wint

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.

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
    showMenu();
    System.out.print("Choose your option (1-4): ");
    int choice = scanner.nextInt();

    switch (choice) {
        case 1:
            checkOdd(number);
            break;
        case 2:
            checkPowerOfTwo(number);
            break;
        case 3:
            showTwosComplement(number);
            break;
        case 4:
            calculateNetworkSegment(scanner);
            break;
        default:
            System.out.println("That is not a valid option");
    }

    scanner.close();
}

// Menu display
public static void showMenu() {
    System.out.println("\n=== MENU ===");
    System.out.println("1. Is number odd?");
    System.out.println("2. Is number a power of 2?");
    System.out.println("3. Two’s complement of number");
    System.out.println("4. Calculate network segment details");
}

// Option 1: Odd check
public static void checkOdd(int number) {
    if ((number & 1) != 1) {
        System.out.println(number + " is odd");
    } else {
        System.out.println(number + " is even");
    }
}

// Option 2: Power of 2 check
public static void checkPowerOfTwo(int number) {
    if ((number & (number - 1)) == 0 && number > 0) {
        System.out.println(number + " is a power of 2");
    } else {
        System.out.println(number + " is not a power of 2");
    }
}

// Option 3: Two’s complement
public static void showTwosComplement(int number) {
    int twosComplement = ~number + 1;
    System.out.println("Two’s complement of " + number + " is " + twosComplement);
}

public static void calculateNetworkSegment(Scanner scanner) {

    scanner.nextLine();

    System.out.print("Enter IP address (e.g. 192.168.1.100): ");
    String ipInput = scanner.nextLine();

    System.out.print("Enter subnet mask (e.g. 255.255.255.224): ");
    String maskInput = scanner.nextLine();

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

    int[] ip = new int[4];
    int[] mask = new int[4];
    int[] network = new int[4];

    for (int i = 0; i < 4; i++) {
        ip[i] = Integer.parseInt(ipParts[i]);
        mask[i] = Integer.parseInt(maskParts[i]);
        network[i] = ip[i] & mask[i];
    }

    System.out.println("\nCalculate the network segment\n");

    // IP in binary
    System.out.print("IP Address:    ");
    for (int i = 0; i < 4; i++) {
        System.out.print(toBinary(ip[i]));
        if (i < 3) System.out.print(".");
    }
    System.out.println();

    // Subnet in binary
    System.out.print("Subnet Mask:   ");
    for (int i = 0; i < 4; i++) {
        System.out.print(toBinary(mask[i]));
        if (i < 3) System.out.print(".");
    }
    System.out.println();

    System.out.println("------------------------------------------------");

    // Network in binary
    System.out.print("Network Addr:  ");
    for (int i = 0; i < 4; i++) {
        System.out.print(toBinary(network[i]));
        if (i < 3) System.out.print(".");
    }
    System.out.println();

    // Network in decimal
    System.out.println("\nThis gives " +
            network[0] + "." +
            network[1] + "." +
            network[2] + "." +
            network[3] +
            " in decimal as the network address."
    );

    // /27 facts
    int hostsPerSubnet = 32;
    int start = network[3];
    int end = start + hostsPerSubnet - 1;

    System.out.println(
            "For a /27 subnet, each segment (or subnet) has " +
                    hostsPerSubnet + " IP addresses (2^5)."
    );

    System.out.println(
            "The range of this network segment is from " +
                    network[0] + "." +
                    network[1] + "." +
                    network[2] + "." + start +
                    " to " +
                    network[0] + "." +
                    network[1] + "." +
                    network[2] + "." + end + "."
    );
}

public static String toBinary(int number) {
    return String.format("%8s", Integer.toBinaryString(number))
            .replace(' ', '0');
}
```

```
}
```

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