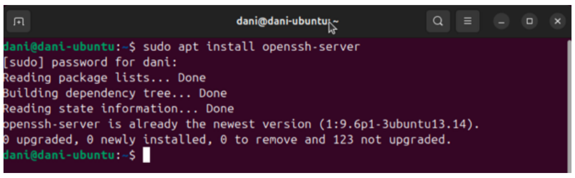
Template Week 6 – Networking

Student number: 592964

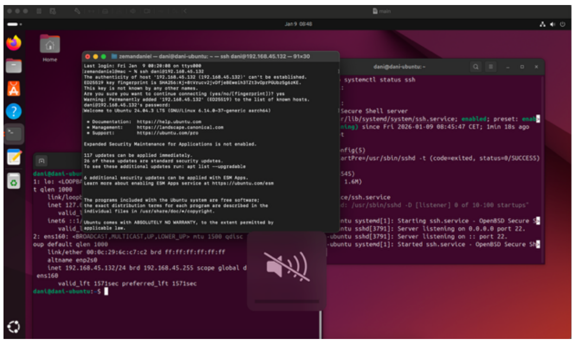
Assignment 6.1: Working from home

Screenshot installation openssh-server:



```
dani@dani-ubuntu:~$ sudo apt install openssh-server
[sudo] password for dani:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:9.6p1-3ubuntu13.14).
0 upgraded, 0 newly installed, 0 to remove and 123 not upgraded.
dani@dani-ubuntu:~$ 
```

Screenshot successful SSH command execution:

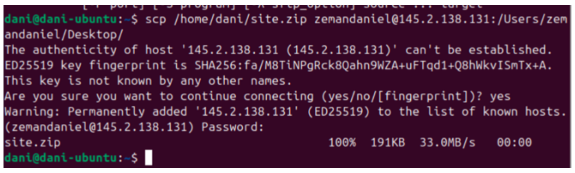


```
Last login: Fri Jan  9 08:20:08 on ttys000
zemandaniel@mac ~ % ssh dani@192.168.45.132
The authenticity of host '192.168.45.132 (192.168.45.132)' can't be established.
ED25519 key fingerprint is SHA256:Kj+8tVzucv2jvOfjeBEweih3TZt3vOprPGUbz5g6zKE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.45.132' (ED25519) to the list of known hosts.
dani@192.168.45.132'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.

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

6 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.
```

```
 systemctl status ssh
Secure Shell server
r/lib/systemd/system/ssh.service; enabled; preset: enab
ning) since Fri 2026-01-09 08:45:47 CET; 1min 18s ago
nfig(5)
artPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
545)
 1.6M)
ce/ssh.service
d: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
ntu systemd[1]: Starting ssh.service - OpenBSD Secure S
ntu sshd[3791]: Server listening on 0.0.0.0 port 22.
-ubuntu sshd[3791]: Server listening on :: port 22.
-ubuntu systemd[1]: Started ssh.service - OpenBSD Secure Sh
```

```
dani@dani-ubun
1: lo: <LOOPBA
t qlen 1000
    link/loopb
    inet 127.0
       valid_l
    inet6 ::1/
       valid_l
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
oup default qlen 1000
    link/ether 00:0c:29:6c:c7:c2 brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    inet 192.168.45.132/24 brd 192.168.45.255 scope global d
 ens160
       valid_lft 1571sec preferred_lft 1571sec
dani@dani-ubuntu:~$ 
```

Screenshot successful execution SCP command:

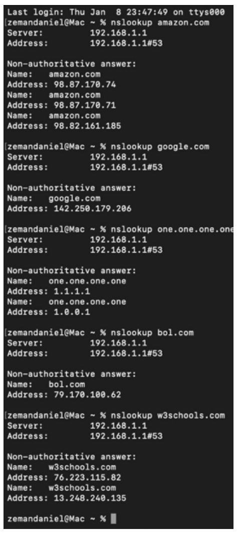


```
dani@dani-ubuntu:~$ scp /home/dani/site.zip zemandaniel@145.2.138.131:/Users/zem
andaniel/Desktop/
The authenticity of host '145.2.138.131 (145.2.138.131)' can't be established.
ED25519 key fingerprint is SHA256:fa/M8TiNPgRck8Qahn9WZA+uFTqd1+Q8hWkvISmTx+A.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '145.2.138.131' (ED25519) to the list of known hosts.
(zemandaniel@145.2.138.131) Password:
site.zip                                      100%  191KB  33.0MB/s   00:00
dani@dani-ubuntu:~$ 
```

Screenshot remmina:

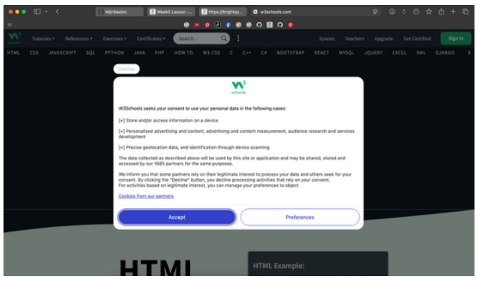
Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:



```
Last login: Thu Jan  8 23:47:49 on ttys000
zemandaniel@Mac ~ % nslookup amazon.com
Server:         192.168.1.1
Address:        192.168.1.1#53

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

zemandaniel@Mac ~ % nslookup google.com
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.179.206

zemandaniel@Mac ~ % nslookup one.one.one.one
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
Name:   one.one.one.one
Address: 1.1.1.1
Name:   one.one.one.one
Address: 1.0.0.1

zemandaniel@Mac ~ % nslookup bol.com
Server:         192.168.1.1
Address:        192.168.1.1#53

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

zemandaniel@Mac ~ % nslookup w3schools.com
Server:         192.168.1.1
Address:        192.168.1.1#53

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

zemandaniel@Mac ~ %
```

Screenshot website visit via IP address:



Assignment 6.3: subnetting

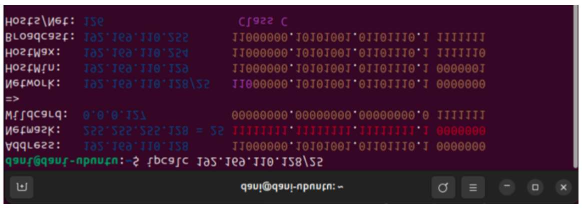
How many IP addresses are in this network configuration 192.168.110.128/25?

128 IP addresses

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

From .129 to .254 (that's 126 usable IP addresses)

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



```
deni@deni-ubuntu:~$ 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
```

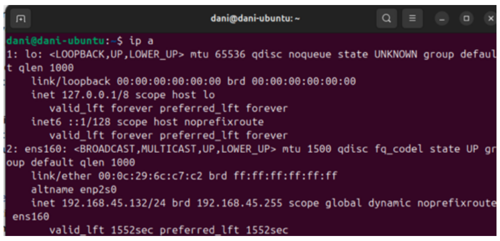
Explain the above calculation in your own words.

IPv4 has 32bits, and in example there is 25 bits -> 32-25= 7 host bits. **2^7 = 128 IP adresses**

.128 is network address and .255 is broadcast address, the rest is in range we can hand out.

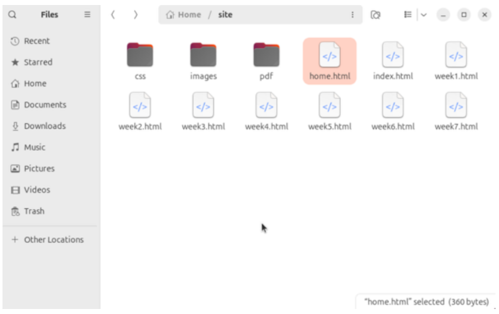
Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

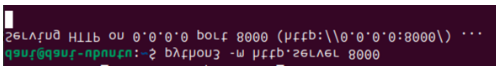


```
dani@dani-ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 00:0c:29:6c:c7:c2 brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    inet 192.168.45.132/24 brd 192.168.45.255 scope global dynamic noprefixroute
 ens160
       valid_lft 1552sec preferred_lft 1552sec
```

Screenshot of Site directory contents:

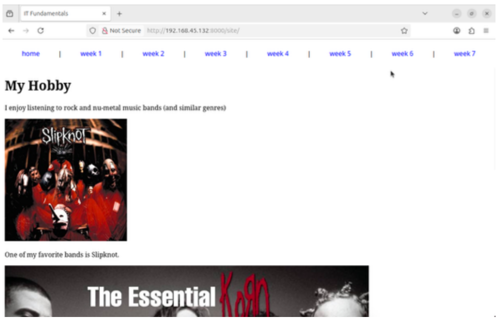


Screenshot python3 webserver command:



```
dani@dani-ubuntu:~$ 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
http://192.168.45.132:8000/site/
home | week 1 | week 2 | week 3 | week 4 | week 5 | week 6 | week 7

My Hobby

I enjoy listening to rock and nu-metal music bands (and similar genres)

One of my favorite bands is Slipknot.

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.

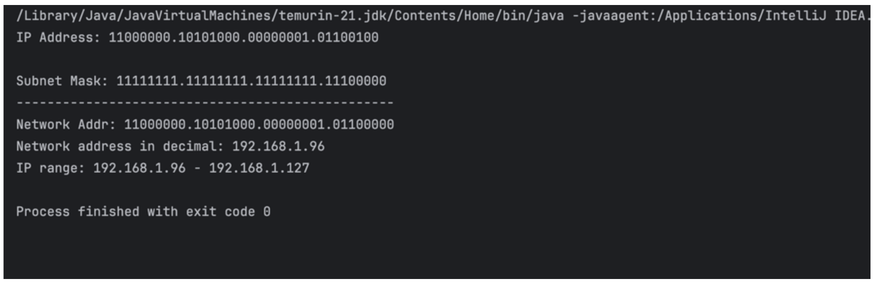
IP: 192.168.1.100 and subnet: 255.255.255.224 for /27

```
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) {
        exercise1();
        exercise2();
        exercise3();
        exercise4();
        exercise5();
        exercise6();
        exercise7();

        String ip = "192.168.1.100"; // input string
        String subnet = "255.255.255.224"; // input subnet
        calculateNetworkSegment(ip, subnet);
    }

    public static void exercise1() { }
    public static void exercise2() { }
    public static void exercise3() { }
    public static void exercise4() { }
    public static void exercise5() { }
    public static void exercise6() { }
    public static void exercise7() { }

    // New method for 6.5
    public static void calculateNetworkSegment(String ip, String subnet) {
        String[] ipParts = ip.split("\\."); // this represents dot -> \\.
        String[] subnetParts = subnet.split("\\.");
        int[] networkParts = new int[4];

        System.out.print("IP Address: ");
        for (int i = 0; i < 4; i++) {
            networkParts[i] = Integer.parseInt(ipParts[i]) &
Integer.parseInt(subnetParts[i]);

            String binary =
```

```java
Integer.toBinaryString(Integer.parseInt(ipParts[i]));

            // if the length is less than 8 fill with 0
            while (binary.length() < 8) {
                binary = "0" + binary;
            }

            System.out.print(binary);
            if (i < 3) System.out.print(".");
        }
        System.out.println();
        System.out.println();

        System.out.print("Subnet Mask: ");
        for (int i = 0; i < 4; i++) {
            String binary =
Integer.toBinaryString(Integer.parseInt(subnetParts[i]));
            while (binary.length() < 8) {
                binary = "0" + binary;
            }
            System.out.print(binary);
            if (i < 3) System.out.print(".");
        }
        System.out.println();

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

        System.out.print("Network Addr: ");
        for (int i = 0; i < 4; i++) {
            String binary = Integer.toBinaryString(networkParts[i]);
            while (binary.length() < 8) {
                binary = "0" + binary;
            }
            System.out.print(binary);
            if (i < 3) System.out.print(".");
        }
        System.out.println();

        String networkAddress = networkParts[0] + "." + networkParts[1] +
"." + networkParts[2] + "." + networkParts[3];
        System.out.println("Network address in decimal: " +
networkAddress);

        int hostBits = 32;
        for (String s : subnetParts) {
            hostBits -= Integer.bitCount(Integer.parseInt(s));
        }
        int numberOfHosts = (int) Math.pow(2, hostBits);

        int[] broadcastParts = networkParts.clone();
        broadcastParts[3] += numberOfHosts - 1;

        String broadcastAddress = broadcastParts[0] + "." +
broadcastParts[1] + "." + broadcastParts[2] + "." + broadcastParts[3];
        System.out.println("IP range: " + networkAddress + " - " +
broadcastAddress);
    }
}
```



```
/Library/Java/JavaVirtualMachines/temurin-21.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.
IP Address: 11000000.10101000.00000001.01100100

Subnet Mask: 11111111.11111111.11111111.11100000
------------------------------------------------
Network Addr: 11000000.10101000.00000001.01100000
Network address in decimal: 192.168.1.96
IP range: 192.168.1.96 - 192.168.1.127

Process finished with exit code 0
```

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