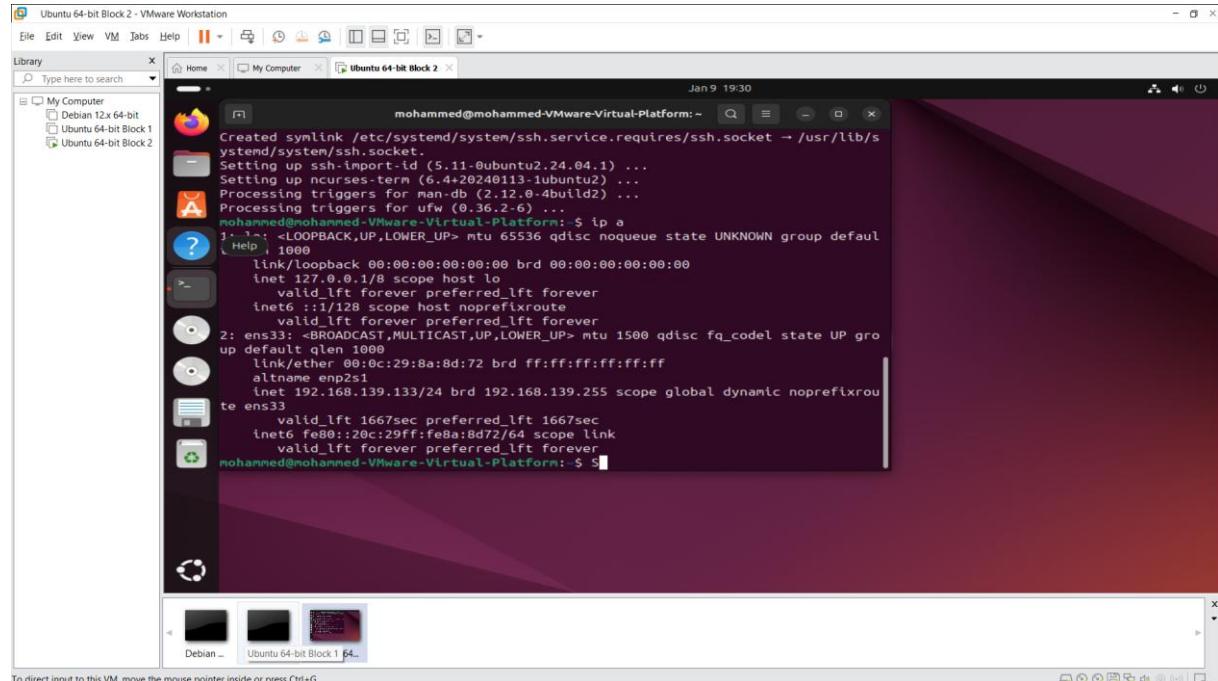


# Template Week 6 – Networking

Student number: 590173

## Assignment 6.1: Working from home

Screenshot installation openssh-server:



```
The authenticity of host '192.168.139.133 (192.168.139.133)' can't be established.
ED25519 key fingerprint is SHA256:qGeDRFTo4F1pR0lMNLuXY3cWv01CSJ5E6dA5k12tRiA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.139.133' (ED25519) to the list of known hosts.
mohammed@192.168.139.133'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.

102 updates can be applied immediately.
28 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.

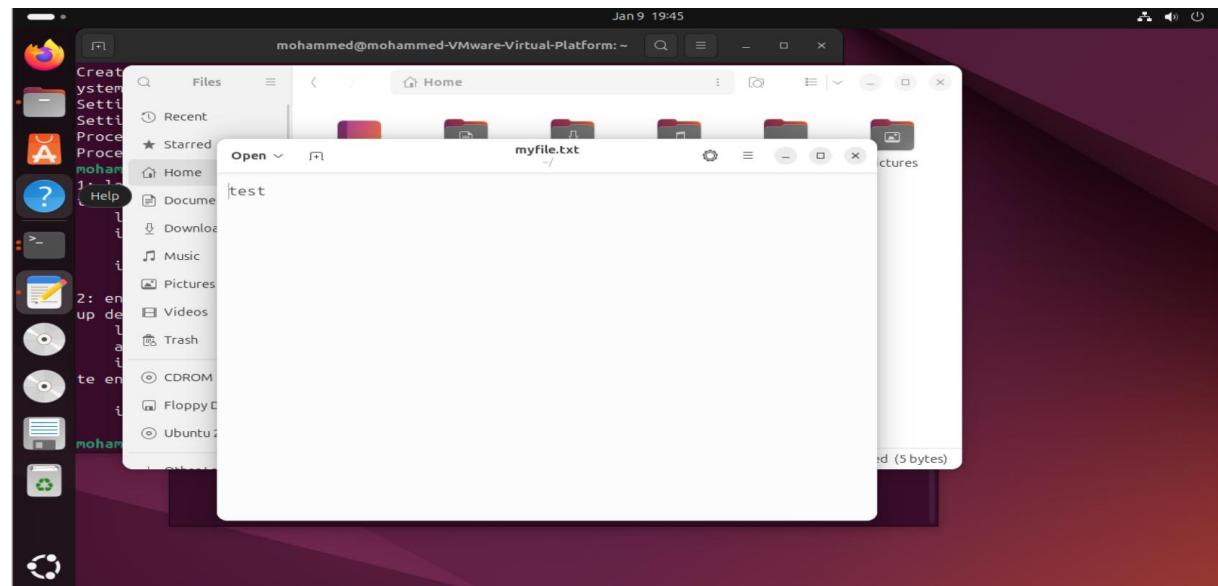
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

mohammed@mohammed-VMware-Virtual-Platform:~$ |
```

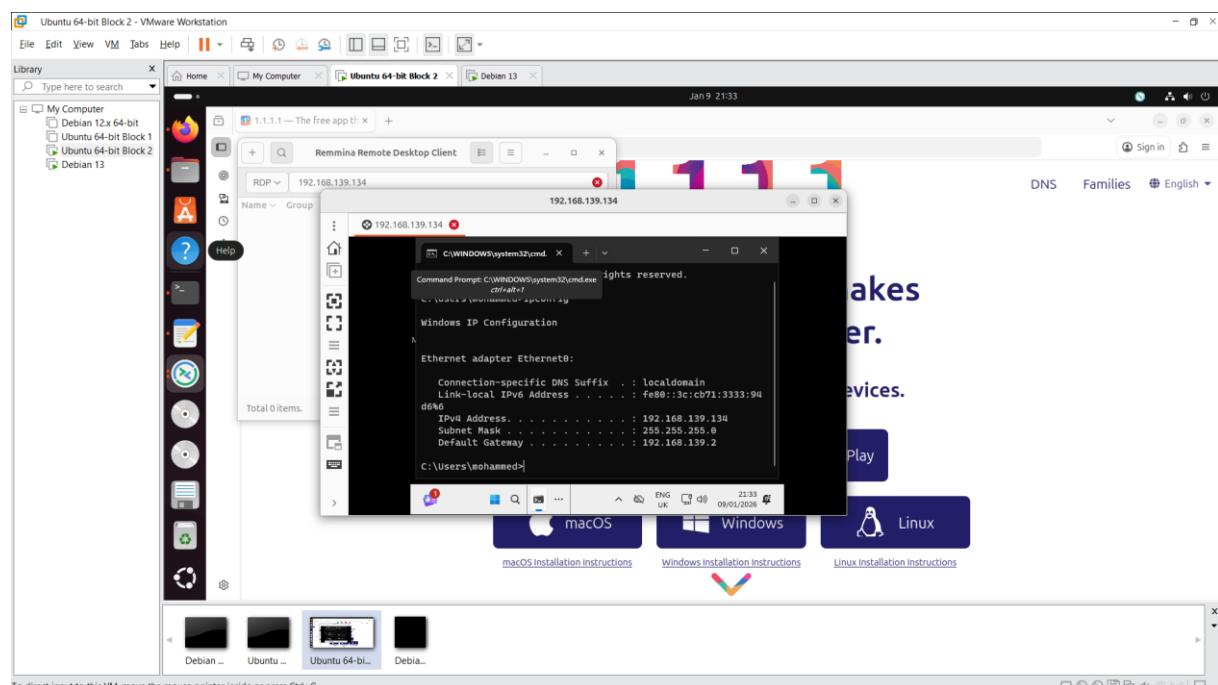
Screenshot successful SSH command execution:

Screenshot successful execution SCP command:

```
mohammed@mohammed-VMware-Virtual-Platform:~$ echo "test" > myfile.txt
mohammed@mohammed-VMware-Virtual-Platform:~$ scp myfile.txt mohammed@192.168.139.133:/home/mohammed
The authenticity of host '192.168.139.133 (192.168.139.133)' can't be established.
ED25519 key fingerprint is SHA256:qGeDRFTo4F1pR0lMNLuXY3cWv01CSJ5E6dA5k12tRiA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.139.133' (ED25519) to the list of known hosts.
mohammed@m192.168.139.133's password:
myfile.txt
mohammed@mohammed-VMware-Virtual-Platform:~$ |
```

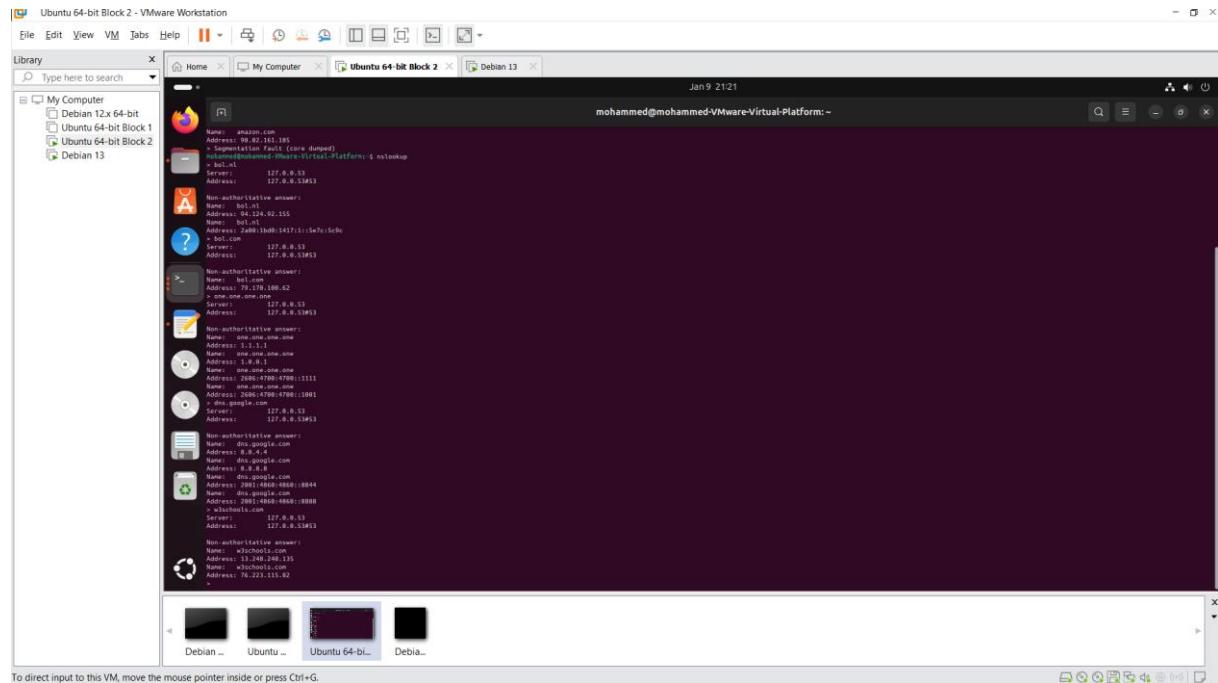


Screenshot remmina:



## Assignment 6.2: IP addresses websites

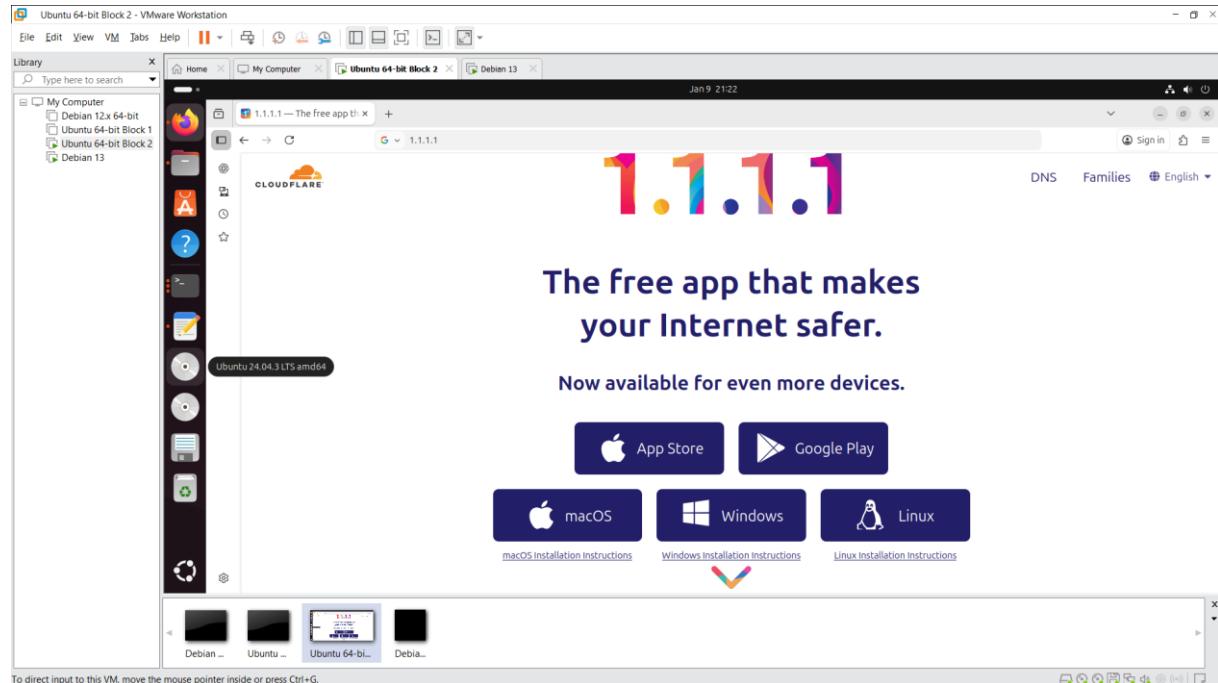
Relevant screenshots nslookup command:



The screenshot shows a VMware Workstation interface with multiple virtual machines listed in the library. The active window is a terminal session titled 'Ubuntu 64-bit Block 2 - VMware Workstation'. The command 'nslookup' is run, and the output shows the resolution of several domain names to their IP addresses. The results include entries for amazon.com, bol.com, helcom.no, one.one.one.one, dns.google.com, and wschools.com.

```
moshamed@moshamed-VMware-Virtual-Platform:~$ nslookup
Non-authoritative answer:
Name:  amazon.com
Address: 98.182.181.195
Name:  bol.com
Address: 94.242.92.155
Name:  helcom.no
Address: 80.100.130.88[141]:51[5e7c5e8v]
Name:  bol.com
Address: 94.242.92.155
Name:  one.one.one.one
Address: 208.67.99.79[1:111]
Name:  dns.google.com
Address: 208.67.99.79[1:1991]
Server:  127.0.0.53
Address: 127.0.0.53
Non-authoritative answer:
Name:  dns.google.com
Address: 8.8.4.4
Name:  dns.google.com
Address: 8.8.8.4
Name:  dns.google.com
Address: 46.180.149.184[1:884]
Name:  dns.google.com
Address: 46.180.149.185[1:885]
Name:  dns.google.com
Address: 76.233.115.82
Name:  wschools.com
Address: 192.168.110.128
```

Screenshot website visit via IP address:



## Assignment 6.3: subnetting

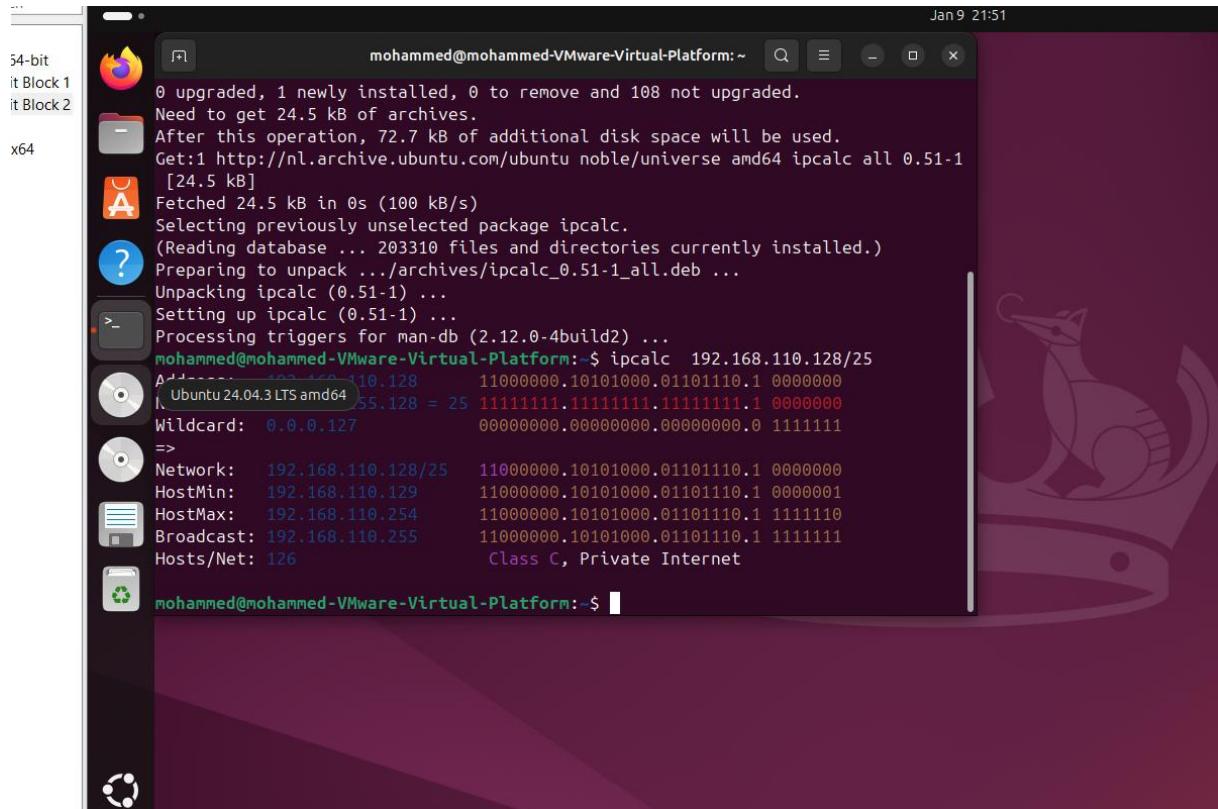
How many IP addresses are in this network configuration 192.168.110.128/25?

**There are 128 total IP addresses because a /25 prefix leaves 7 bits for hosts ( $2^7 = 128$ ).**

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

**The usable range is 192.168.110.129 through 192.168.110.254, excluding the network and broadcast addresses**

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



A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "mohammed@mohammed-VMware-Virtual-Platform:~". The terminal content shows the output of the command `ipcalc 192.168.110.128/25`. The output includes:

```
mohammed@mohammed-VMware-Virtual-Platform:~$ ipcalc 192.168.110.128/25
Address: 192.168.110.128      11000000.10101000.01101110.1 00000000
Mask:   255.255.255.128     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    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.

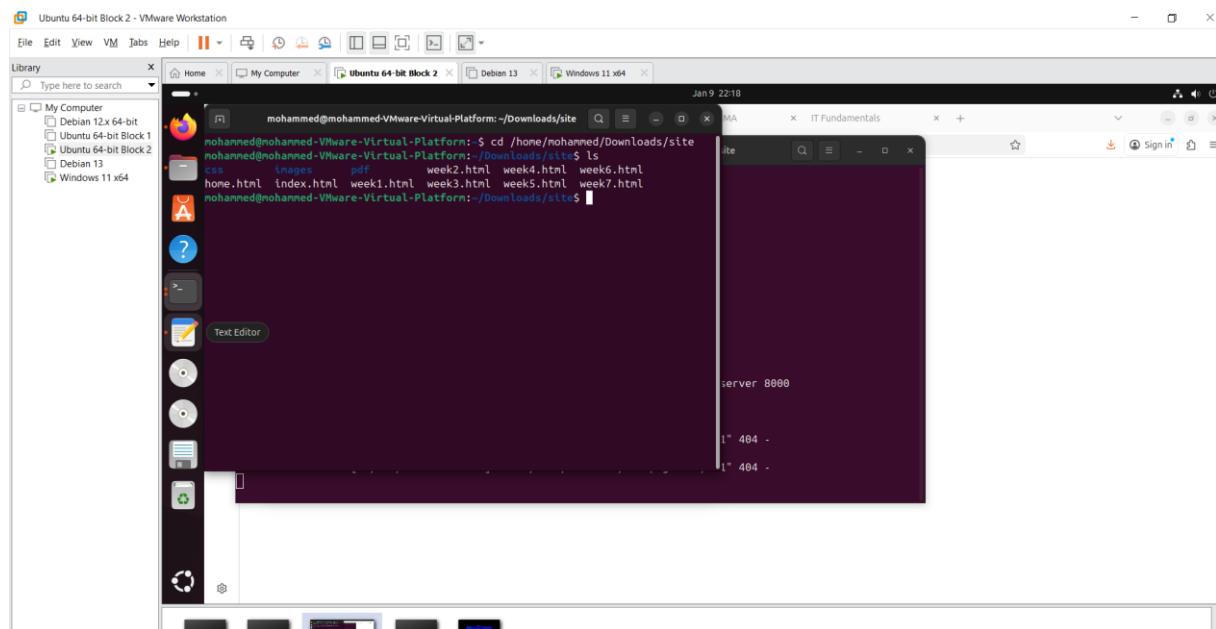
**The /25 mask locks the first 25 bits (shown in red in your terminal), leaving 7 bits to create 128 total addresses; we just set aside the first and last ones for network rules.**

#### Assignment 6.4: HTML

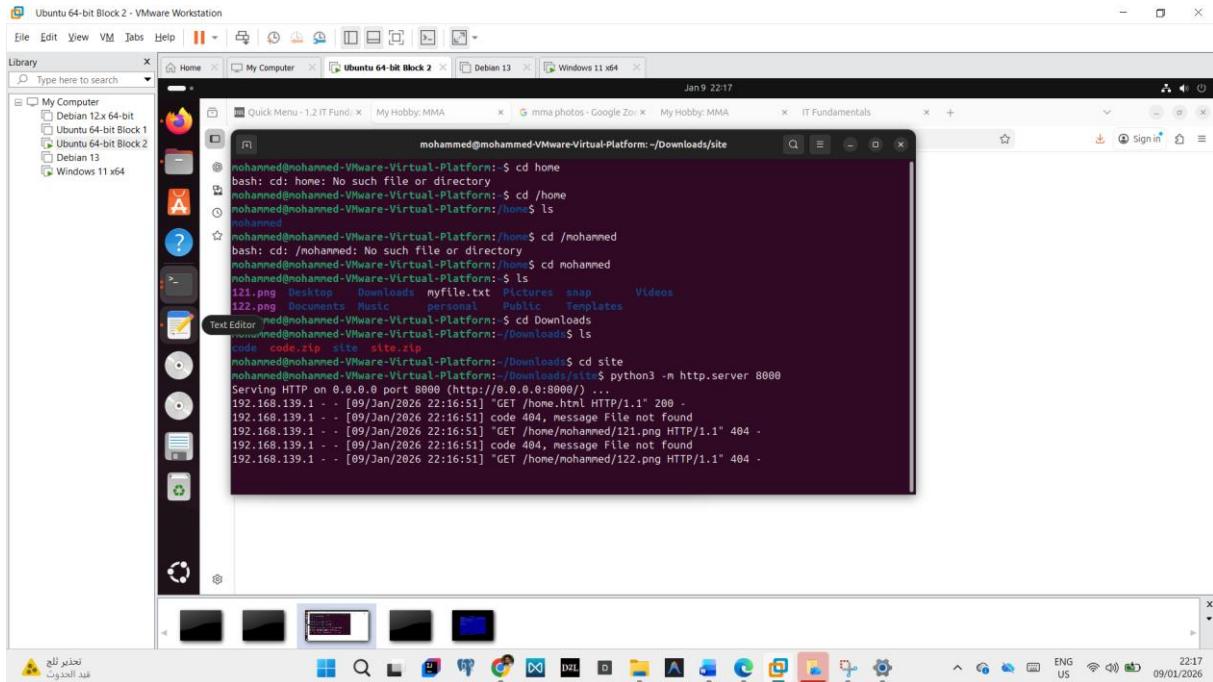
Screenshot IP address Ubuntu VM:

```
mohammed@mohammed-VMware-Virtual-Platform:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    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
    link/ether 00:0c:29:8a:8d:72 brd ff:ff:ff:ff:ff:ff
    altnet enp2s1
    inet 192.168.139.133/24 brd 192.168.139.255 scope global dynamic noprefixroute
        valid_lft 1516sec preferred_lft 1516sec
        inet6 fe80::20c:29ff:fe8a:8d72/64 scope link
            valid_lft forever preferred_lft forever
mohammed@mohammed-VMware-Virtual-Platform:~$ ^C
mohammed@mohammed-VMware-Virtual-Platform:~$ ^C
```

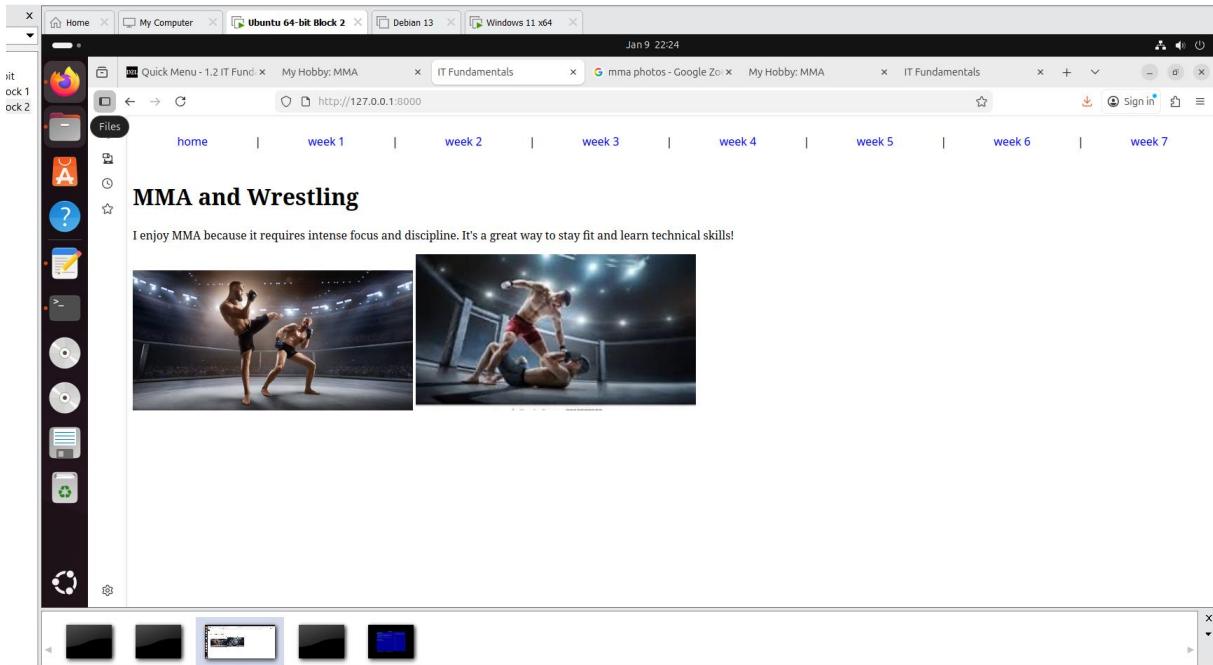
Screenshot of Site directory contents:



Screenshot python3 webserver command:



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.

```
public class Main {  
  
    public static void main(String[] args) {  
  
        // Step 1: Input the Strings  
  
        String ip = "192.168.1.100";  
        String subnet = "255.255.255.224";  
  
        // Step 2: Split them into four parts (octets)  
        String[] ipParts = ip.split("\\.");  
        String[] subnetParts = subnet.split("\\.");  
  
        System.out.println("--- Assignment 6.5: Network Segment Calculation ---");  
        System.out.print("Network Address: ");  
  
        // Step 3: Loop through all 4 parts to calculate the segment  
        String networkAddress = "";  
        for (int i = 0; i < 4; i++) {  
            // Convert text to numbers  
            int ipOctet = Integer.parseInt(ipParts[i]);  
            int subnetOctet = Integer.parseInt(subnetParts[i]);  
    }  
}
```

```
// Perform the Bitwise AND (&) operation
int resultOctet = ipOctet & subnetOctet;

// Build the final string
networkAddress += resultOctet;
if (i < 3) networkAddress += ".";
}

// Step 4: Output the results
System.out.println(networkAddress); // This will show 192.168.1.96

// Bonus: Calculate the range for a /27 (32 addresses)
System.out.println("Segment Range: 192.168.1.96 to 192.168.1.127");
System.out.println("-----");
}

}
```

The screenshot shows a Java code editor and a terminal window. The code editor displays `Main.java` with the following content:

```
1- public class Main {
2-     public static void main(String[] args) {
3-         // Step 1: Input the Strings
4-         String ip = "192.168.1.100";
5-         String subnet = "255.255.255.224";
6-
7-         // Step 2: Split them into four parts (octets)
8-         String[] ipParts = ip.split("\\.");
9-         String[] subnetParts = subnet.split("\\.");
10-
11        System.out.println("--- Assignment 6.5: Network Segment Calculation ---");
12        System.out.print("Network Address: ");
13-
14        // Step 3: Loop through all 4 parts to calculate the segment
15        String networkAddress = "";
16        for (int i = 0; i < 4; i++) {
17            // Convert text to numbers
18            int ipOctet = Integer.parseInt(ipParts[i]);
19            int subnetOctet = Integer.parseInt(subnetParts[i]);
20-
21            // Perform the Bitwise AND (&) operation
22            int resultOctet = ipOctet & subnetOctet;
23-
24            // Build the final string
25            networkAddress += resultOctet;
26            if (i < 3) networkAddress += ".";
27        }
28-
29        // Step 4: Output the results
30        System.out.println(networkAddress); // This will show 192.168.1.96
31-
32        // Bonus: Calculate the range for a /27 (32 addresses)
33        System.out.println("Segment Range: 192.168.1.96 to 192.168.1.127");
34        System.out.println("-----");
35    }
36 }
```

The terminal window below shows the execution of the program:

```
--- Assignment 6.5: Network Segment Calculation ---
Network Address: 192.168.1.96
Segment Range: 192.168.1.96 to 192.168.1.127
-----
...Program finished with exit code 0
Press ENTER to exit console.
```

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