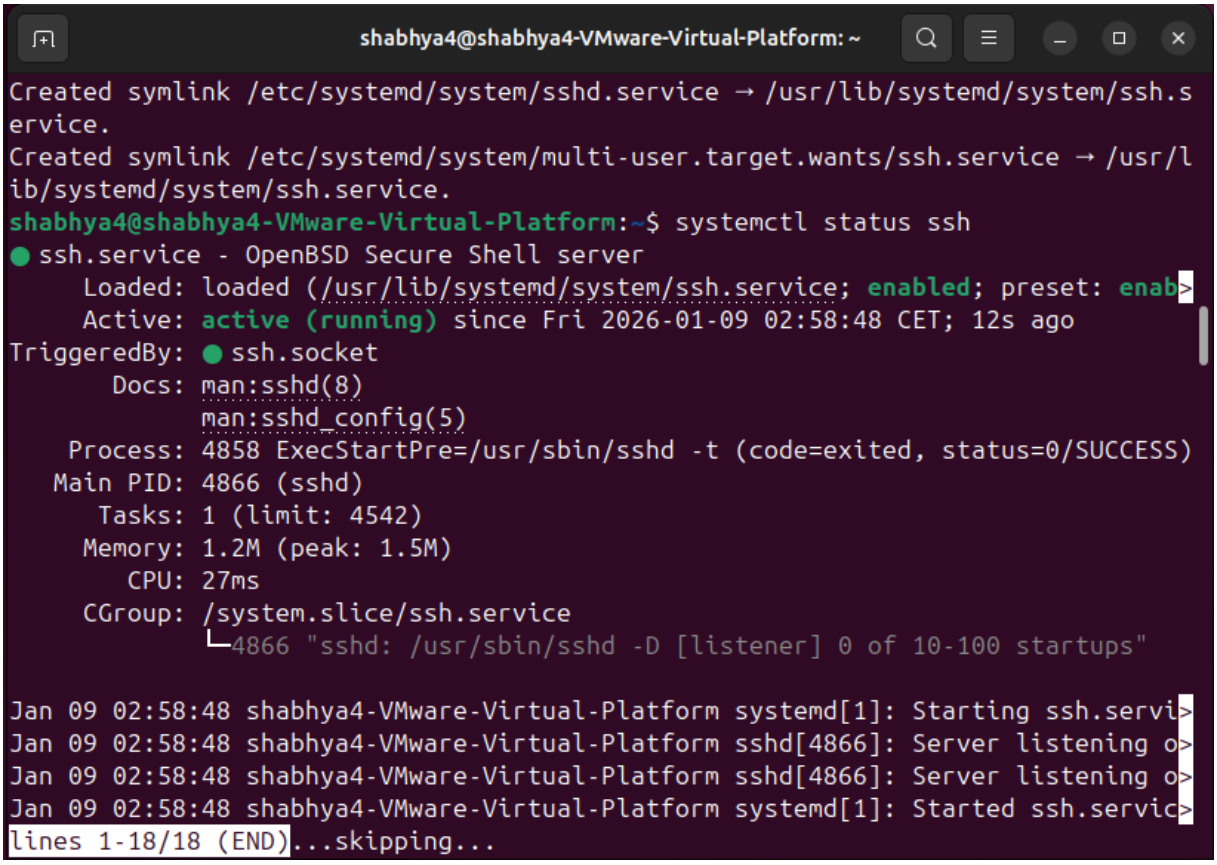
Week 6 – Networking

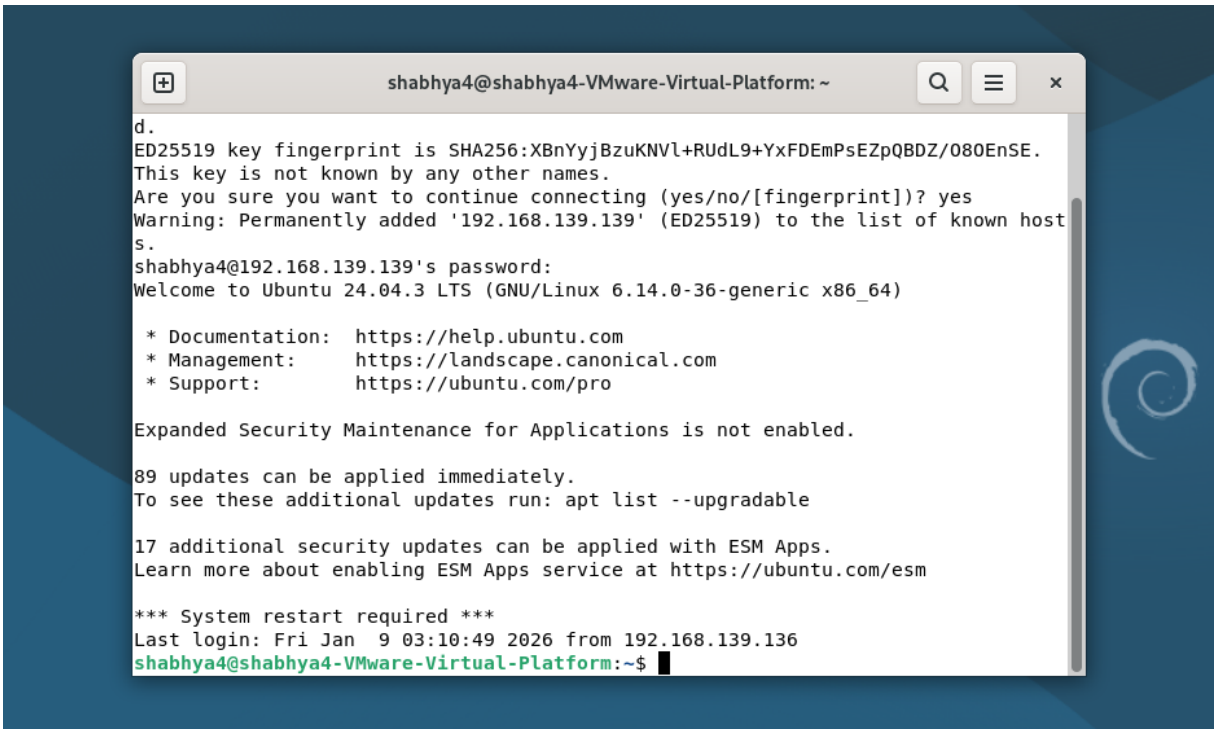
Student number: 578438

Assignment 6.1: Working from home

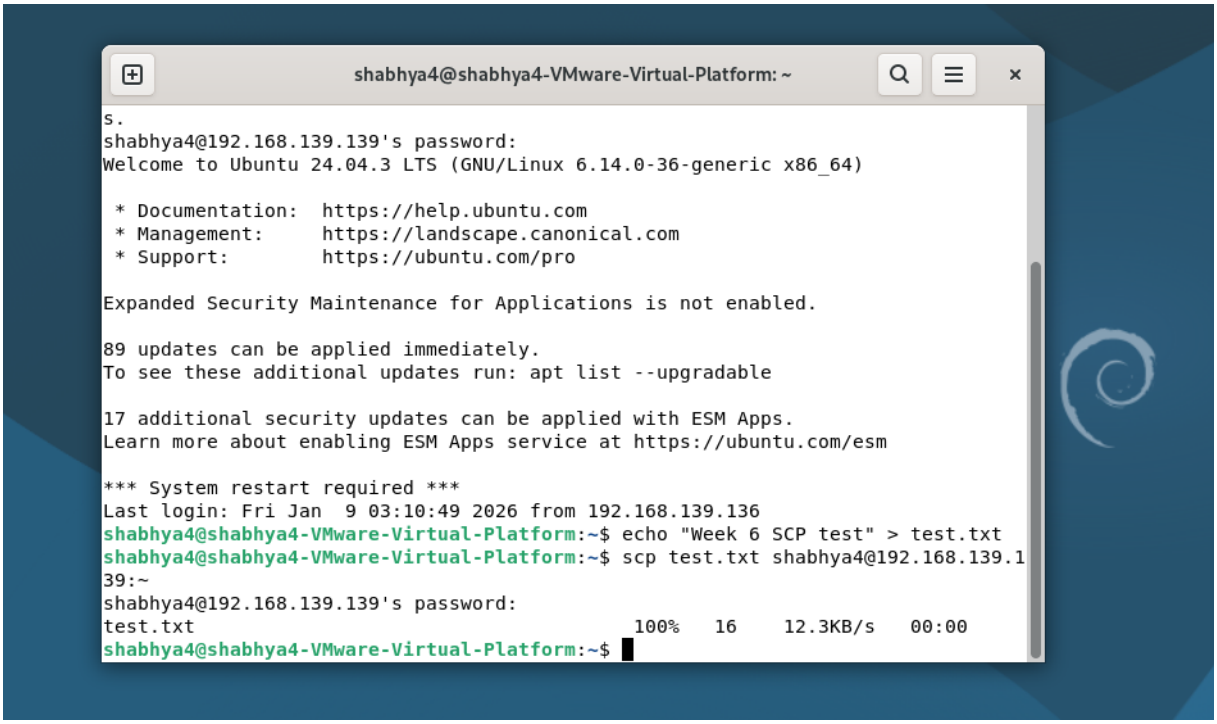
Screenshot installation openssh-server:



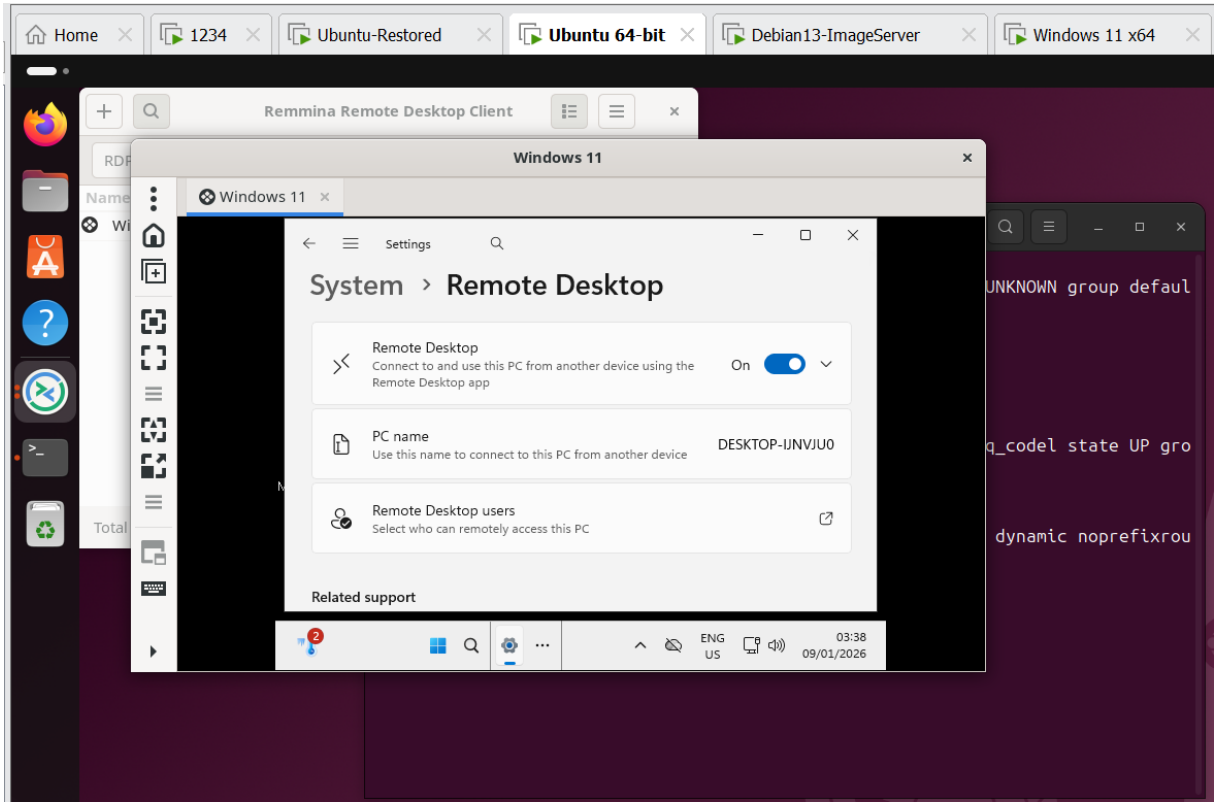
Screenshot successful SSH command execution:



Screenshot successful execution SCP command:

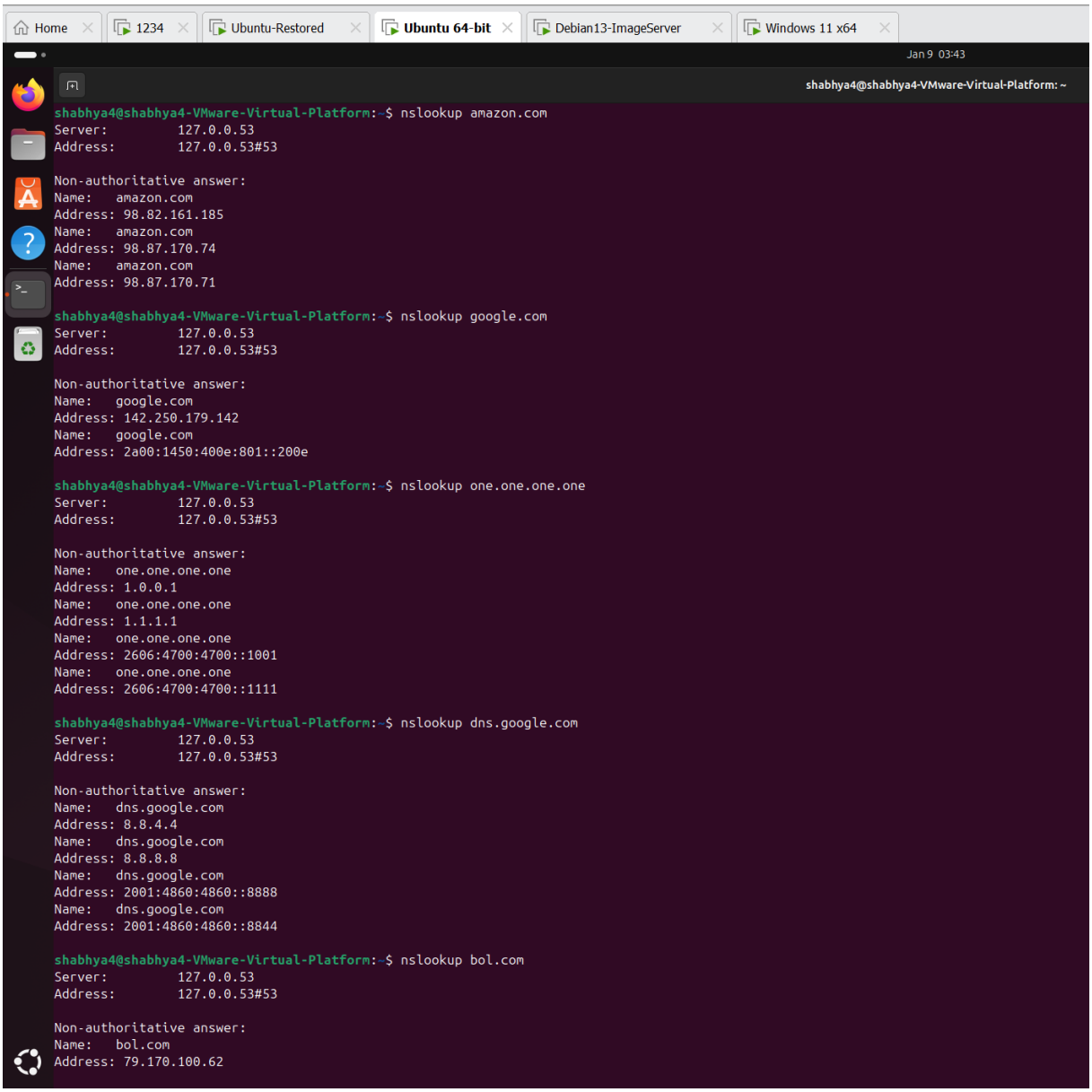


Screenshot remmina:



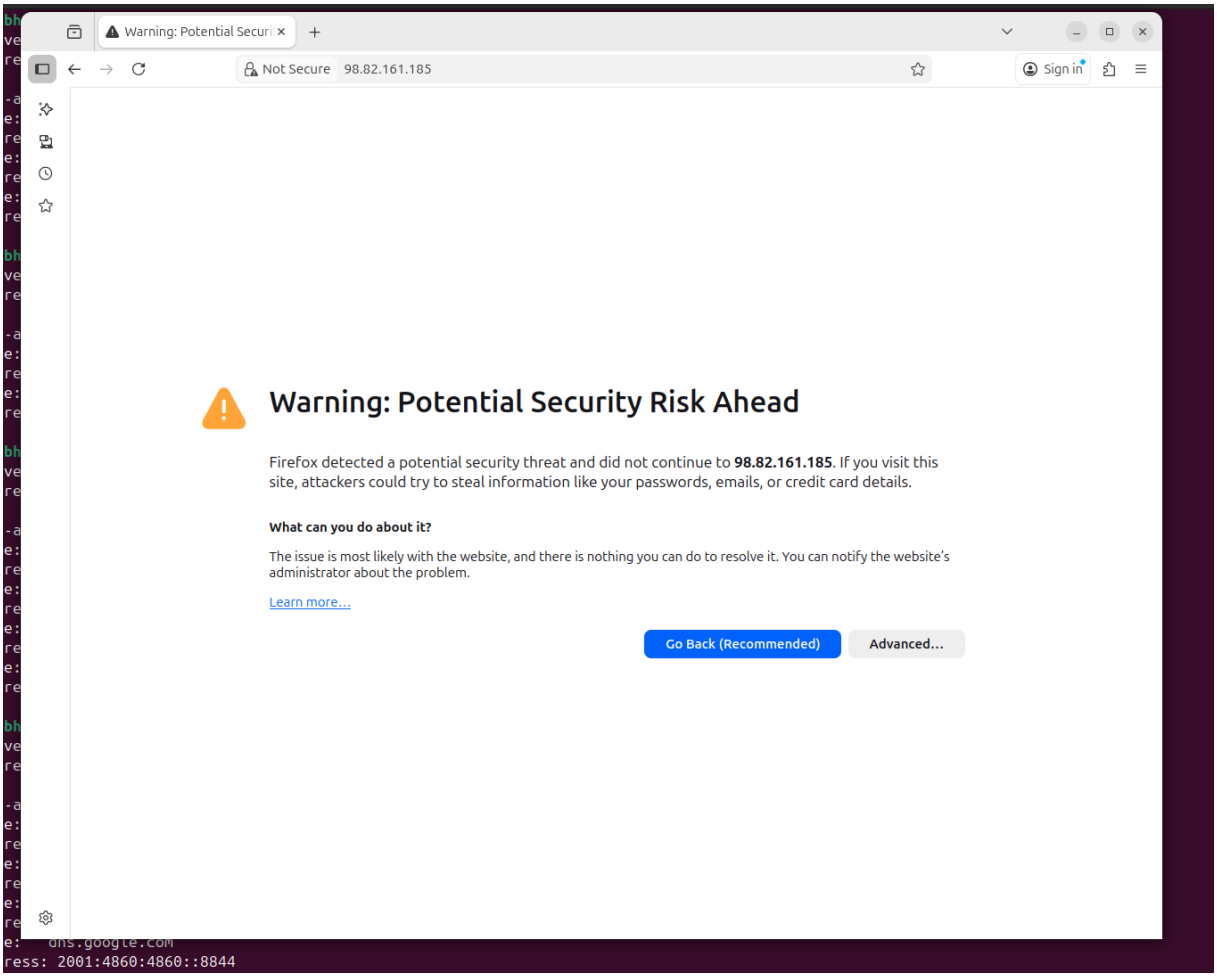
Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:



```
shabhya4@shabhya4-VMware-Virtual-Platform:~$ nslookup bol.com
Server:         127.0.0.53
Address:        127.0.0.53#53

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

shabhya4@shabhya4-VMware-Virtual-Platform:~$ nslookup w3schools.com
Server:         127.0.0.53
Address:        127.0.0.53#53

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

shabhya4@shabhya4-VMware-Virtual-Platform:~$
```

Screenshot website visit via IP address:



Assignment 6.3: subnetting

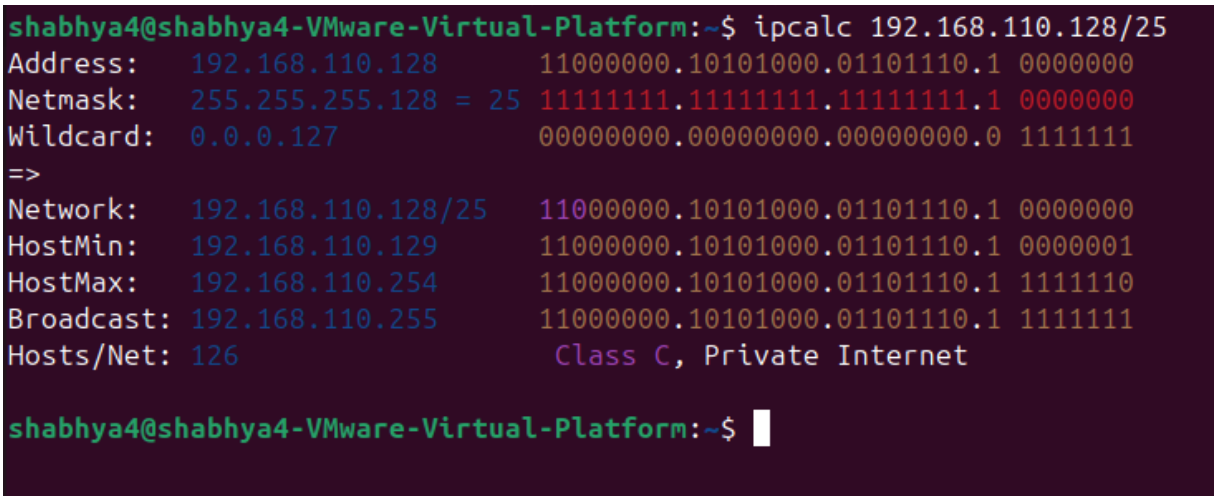
How many IP addresses are in this network configuration 192.168.110.128/25?

→ There are 128 IP addresses in the network 192.168.110.128/25.

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

→ The usable host range is 192.168.110.129 up to 192.168.110.254, which gives 126 usable IP addresses.

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



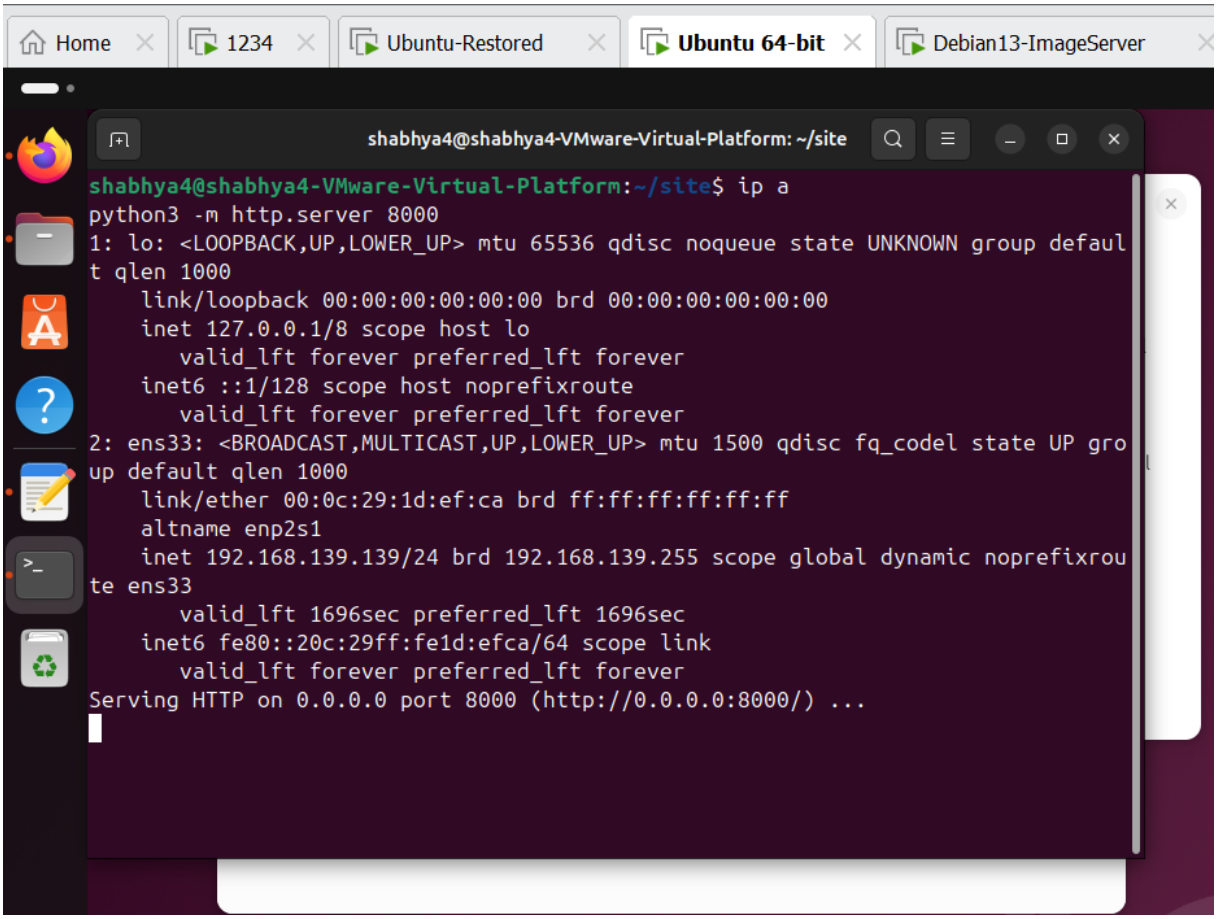
Running ipcalc 192.168.110.128/25 confirms my answers: it shows Network 192.168.110.128, HostMin 192.168.110.129, HostMax 192.168.110.254 and Hosts/Net 126

Explain the above calculation in your own words.

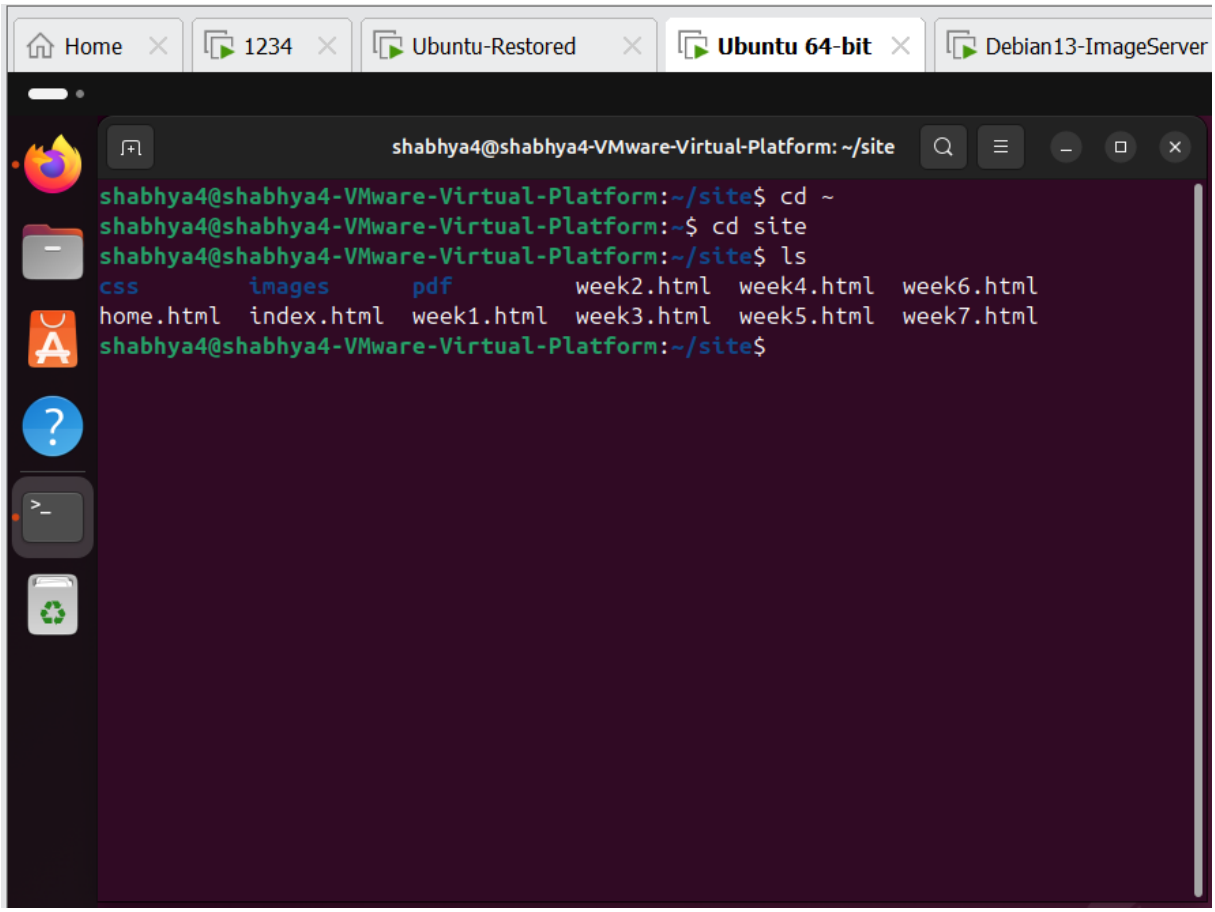
→ *The "/25" means 25 bits are used for the network part and the remaining 7 bits are for hosts. With 7 host bits there are $2^7 = 128$ total addresses in this subnet. The subnet range is from 192.168.110.128 to 192.168.110.255. The first address (192.168.110.128) is the network address, and the last address (192.168.110.255) is the broadcast address, so they cannot be assigned to devices. All addresses in between (192.168.110.129–192.168.110.254) are usable for hosts, which gives 128 − 2 = 126 usable host addresses*

Assignment 6.4: HTML

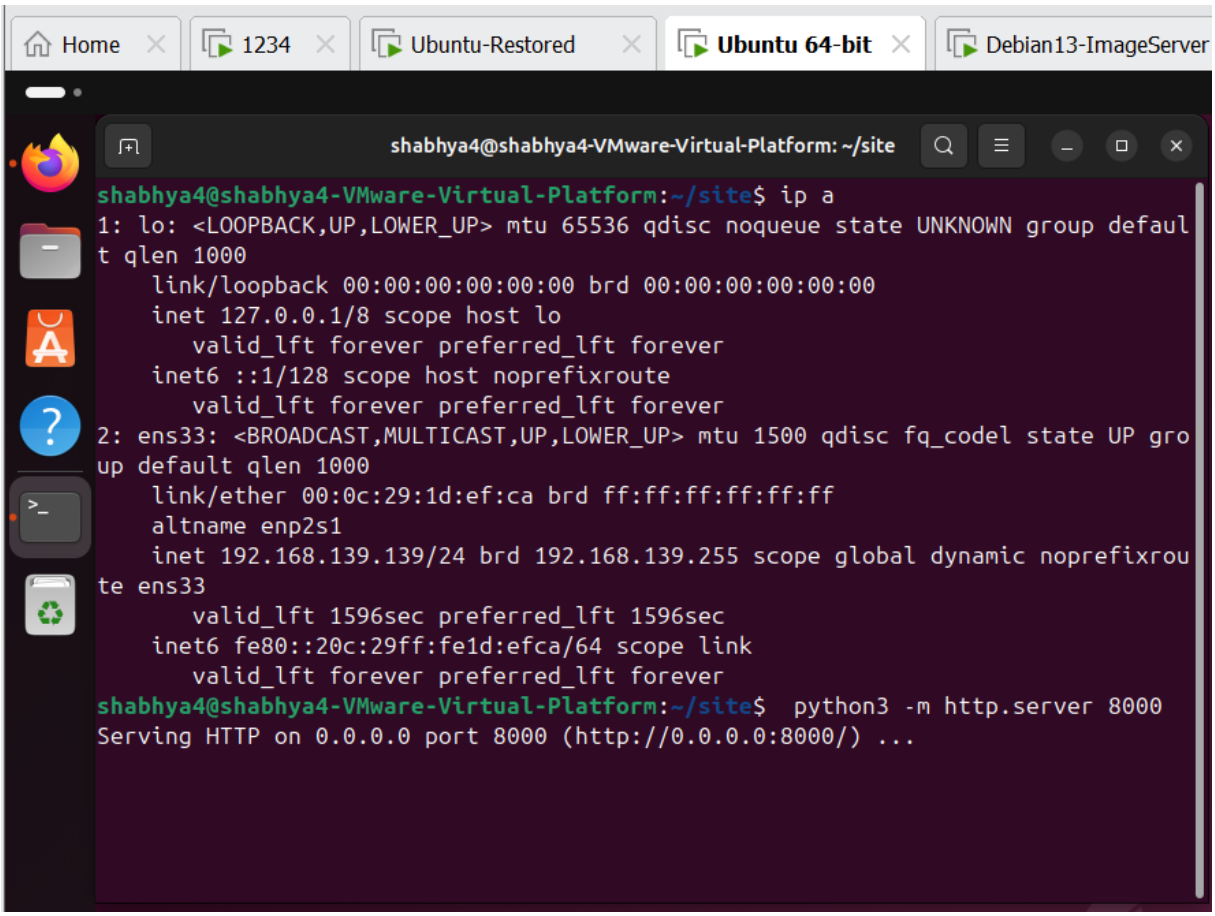
Screenshot IP address Ubuntu VM:



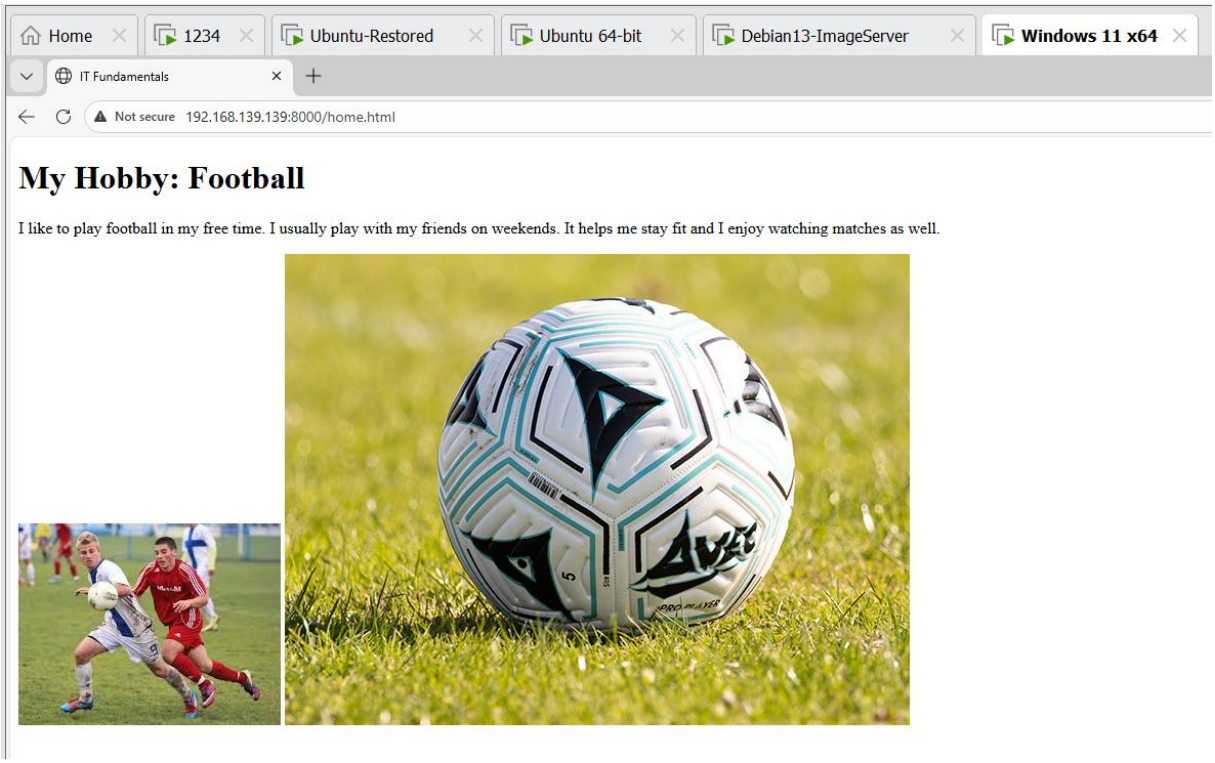
Screenshot of Site directory contents:



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.

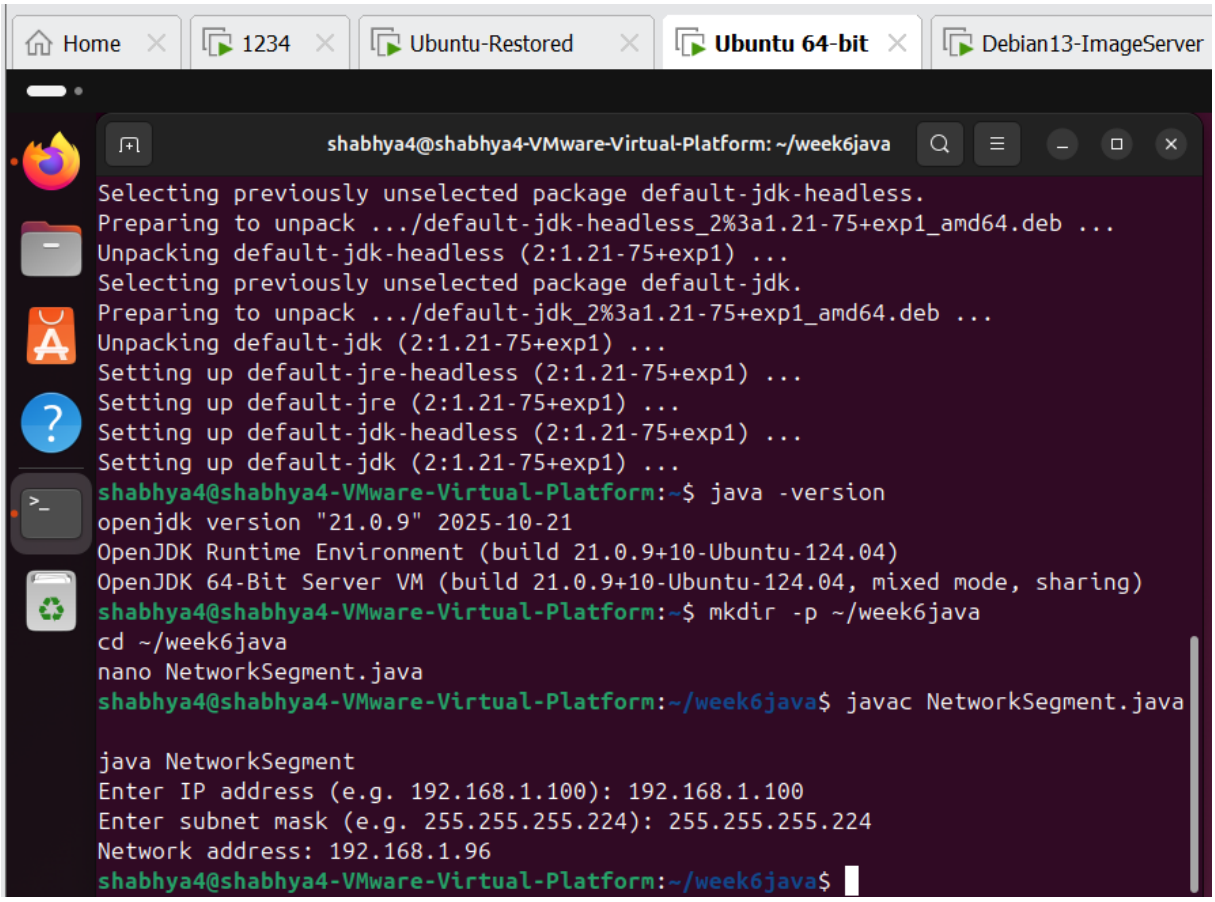
IP: 192.168.1.100 and subnet: 255.255.255.224 for /27

```
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 (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.

```
import java.util.Scanner;

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

      System.out.print("Enter IP address (e.g. 192.168.1.100): ");
      String ipStr = sc.nextLine().trim();

      System.out.print("Enter subnet mask (e.g. 255.255.255.224): ");
      String maskStr = sc.nextLine().trim();

      int[] ip = parseIp(ipStr);
      int[] mask = parseIp(maskStr);
      int[] net = new int[4];

      // bitwise AND each octet
      for (int i = 0; i < 4; i++) {
         net[i] = ip[i] & mask[i];
      }

      System.out.println("Network address: "
            + net[0] + "." + net[1] + "." + net[2] + "." + net[3]);
   }

   // Convert "192.168.1.100" to {192, 168, 1, 100}
   private static int[] parseIp(String s) {
      String[] parts = s.split("\\.");
      if (parts.length != 4) {
```

```
            throw new IllegalArgumentException("Invalid IP format: " + s);
        }
        int[] result = new int[4];
        for (int i = 0; i < 4; i++) {
            result[i] = Integer.parseInt(parts[i]);
        }
        return result;
    }
}
```



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