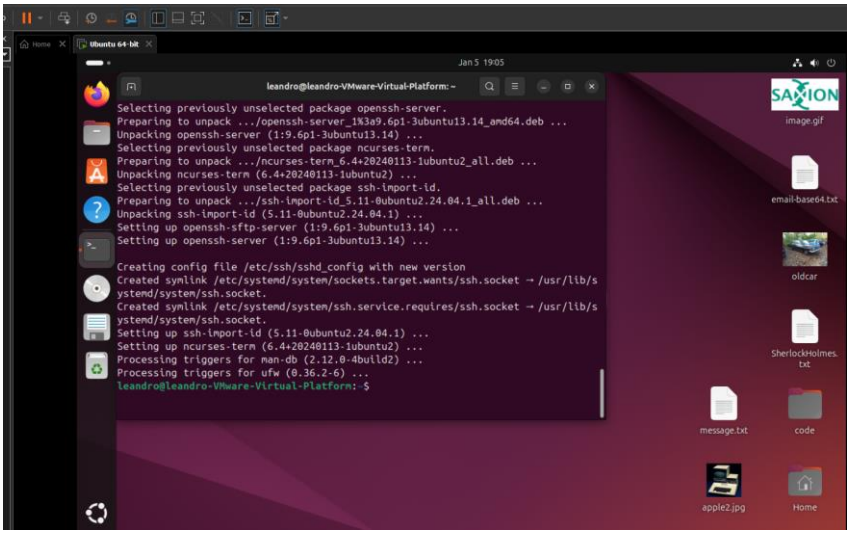
Week 6 – Networking

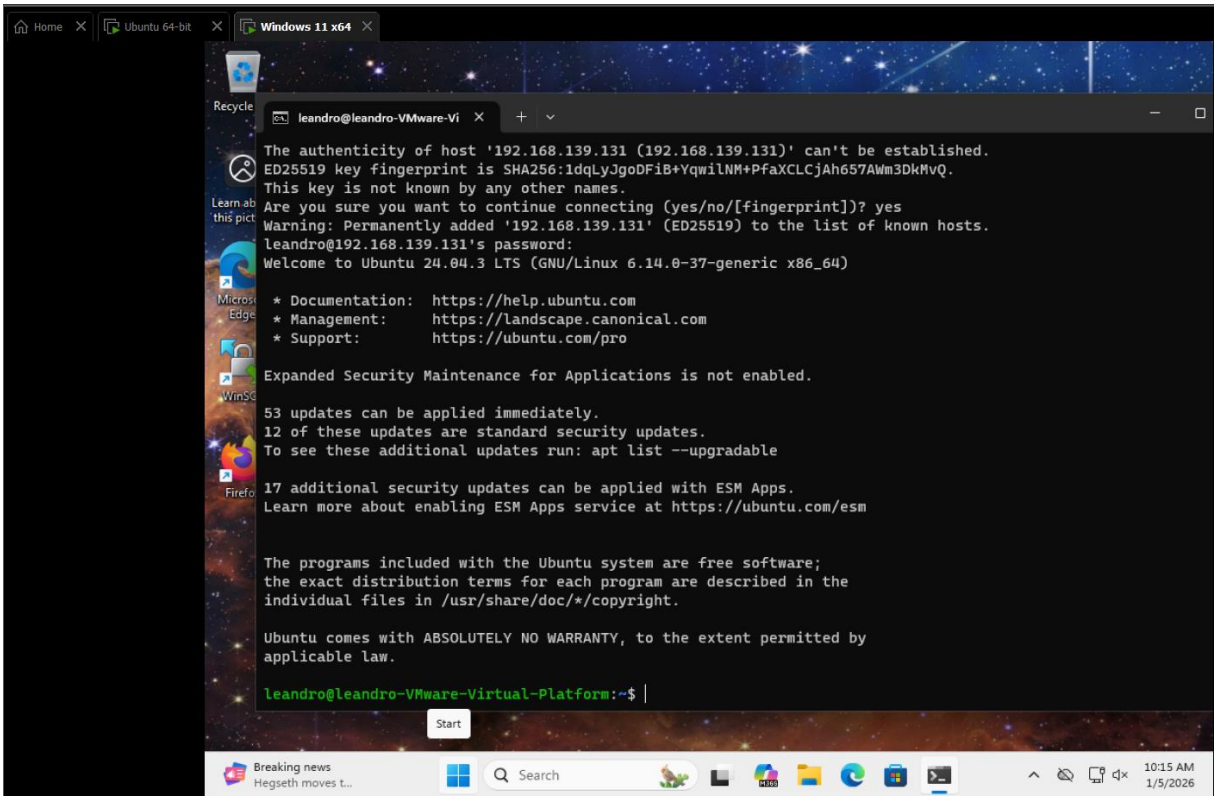
Student number: 576255

Assignment 6.1: Working from home

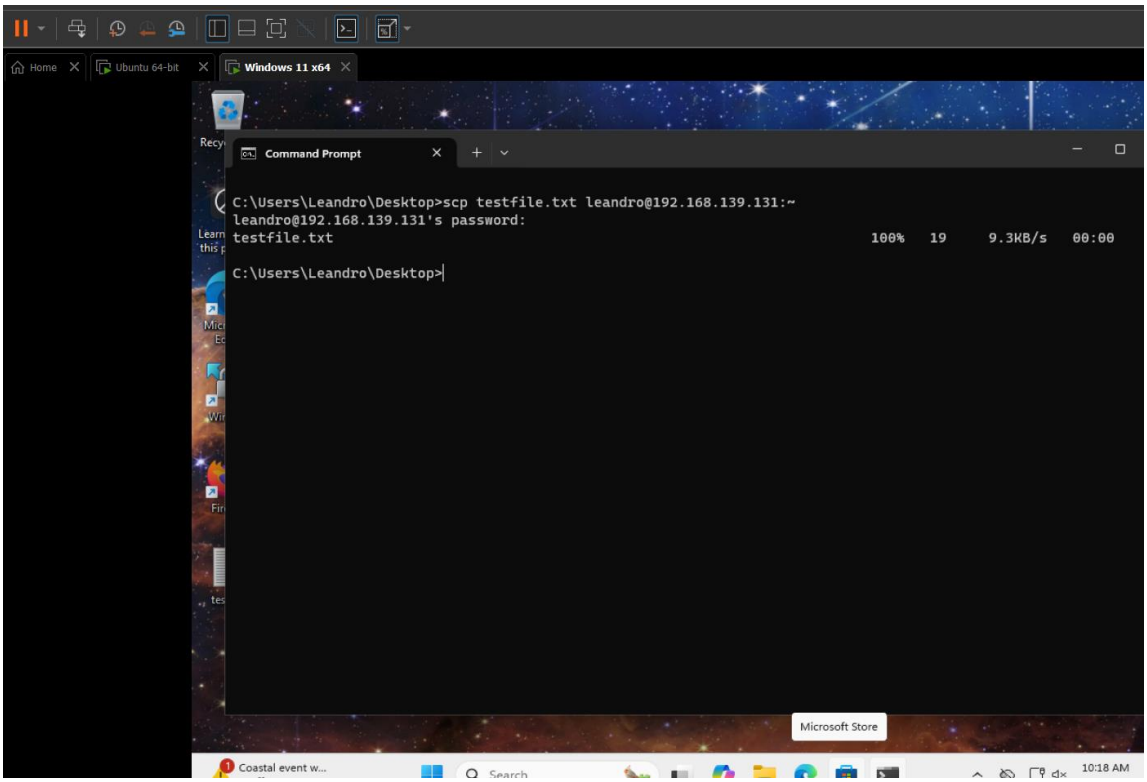
Screenshot installation openssh-server:



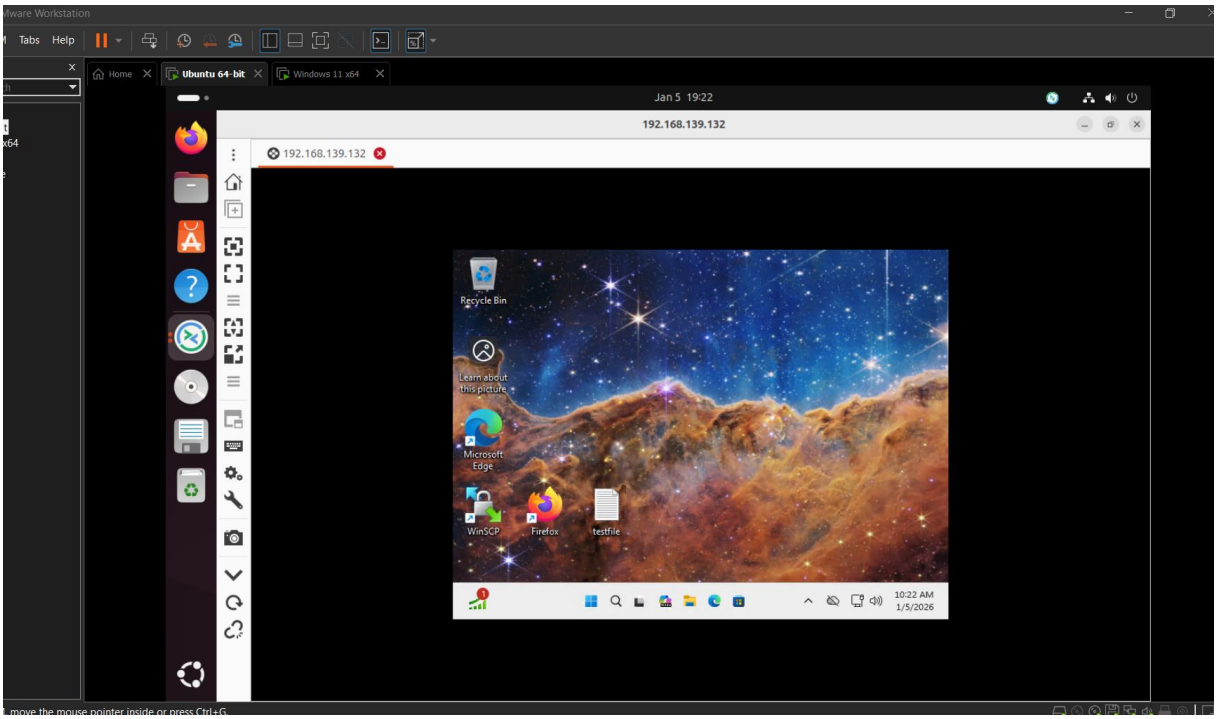
Screenshot successful SSH command execution:



Screenshot successful execution SCP command:

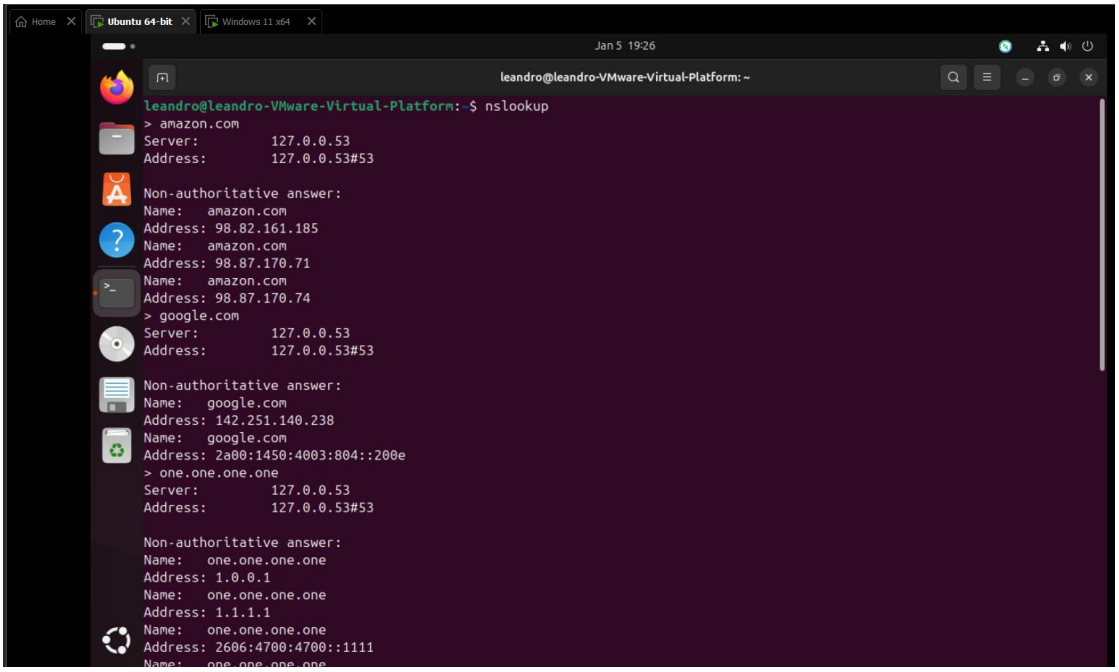


Screenshot remmina:

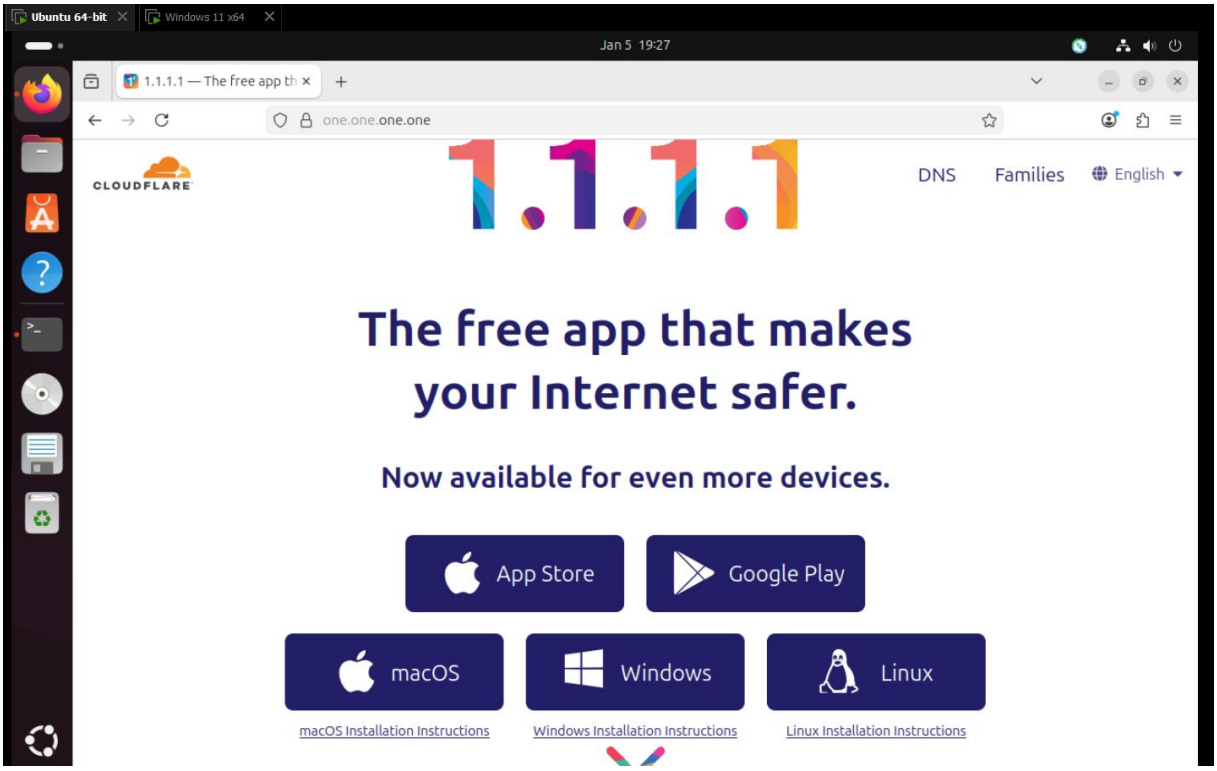


Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:



Screenshot website visit via IP address:



What does nslookup do?

Nslookup performs Name Server Lookups to resolve domain names to IP addresses and vice versa.

What does DNS do?

DNS (Domain Name Server) translates human-friendly domain names to machine-friendly IP addresses. As stated in the course material, this is what allows computers to connect to the appropriate servers.

Why did the website load when I entered the IP address directly?

Since entering the domain name resolves to in the previous step allowed the site to load, this shows how human-readable domain names are a convenience. However, underlying all communications via the Internet are these machine-readable IP addresses that allow for routing.

Assignment 6.3: subnetting

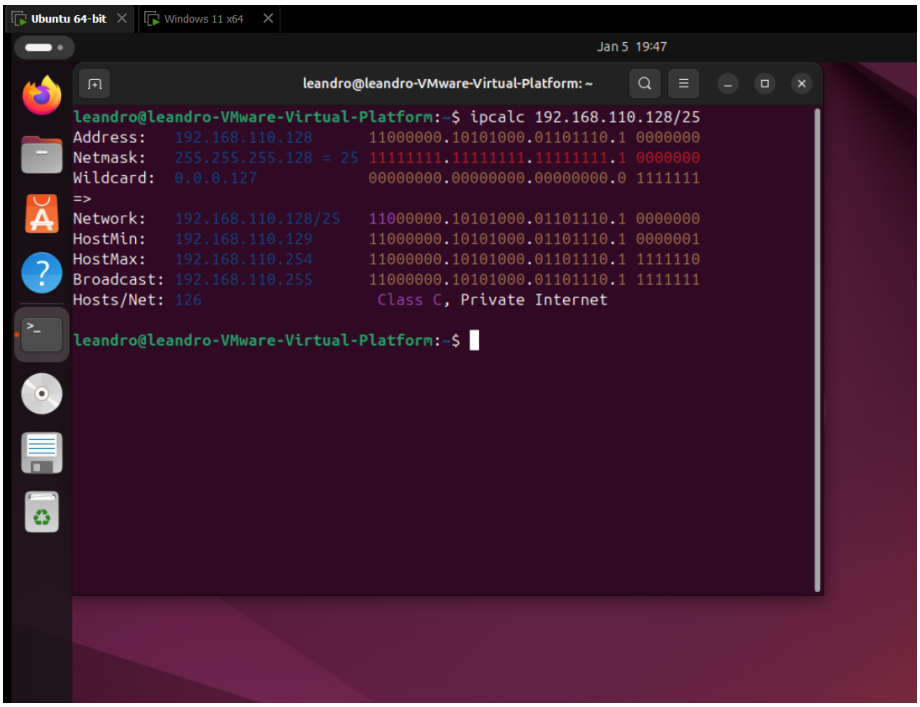
How many IP addresses are in this network configuration 192.168.110.128/25?

There are 128 total IP addresses. (Calculation: A /25 subnet leaves 7 bits for hosts. 2^7 = 128)

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

192.168.110.129 to 192.168.110.254 (The very first IP .128 is the Network ID, and the very last IP .255 is the Broadcast address, so they cannot be used).

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



Explain the above calculation in your own words.

Subnet Calculation: /25 means that 25 bits are accounted for the network and there are 7 bits (32 total - 25 = 7) left for hosts

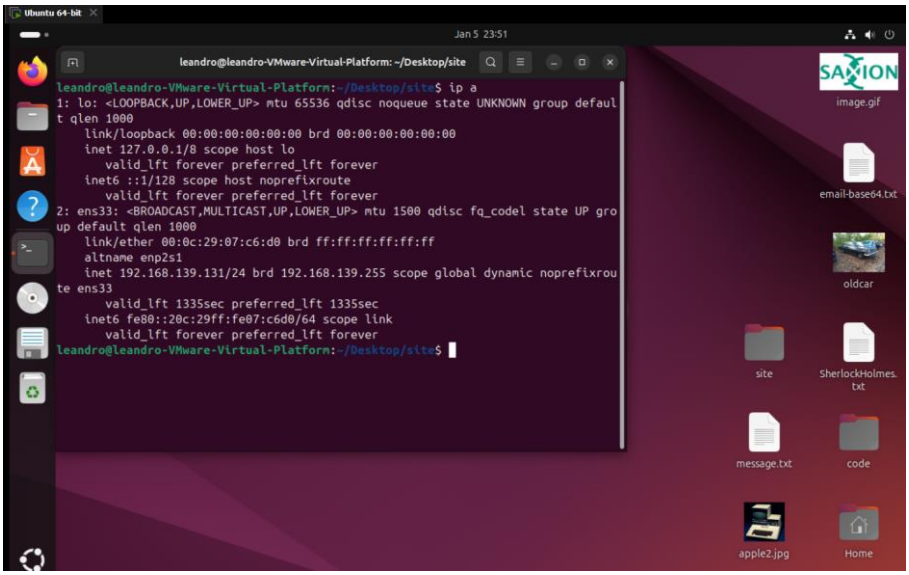
Total Addresses: Using the formula 2^7 it would be 128

Usable Range: In any subnet, the first is the network address (192.168.110.128) and the last is the broadcast address (192.168.110.255), thus we deduct those two from total giving us usable 126 computers

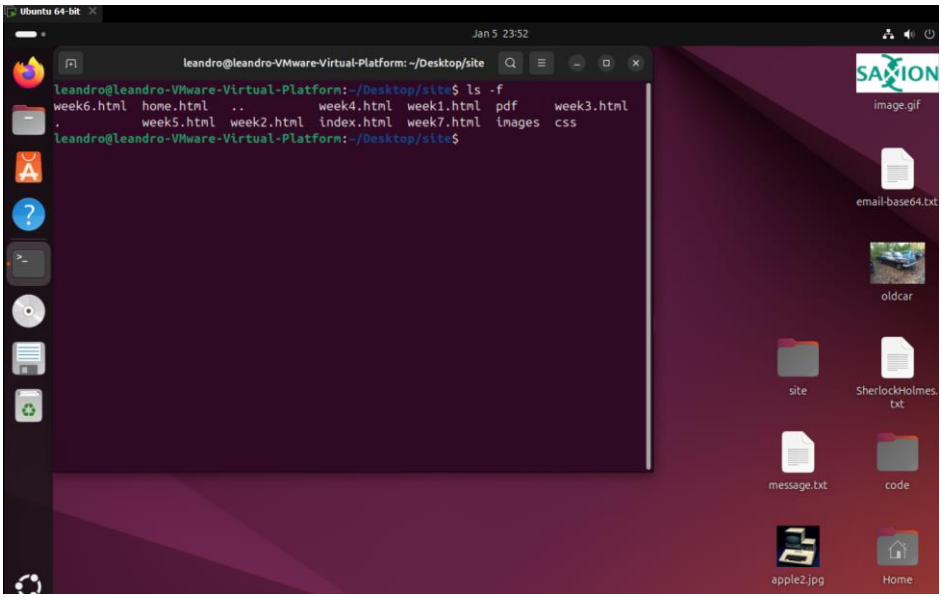
The Range: This network goes from .128; the first usable host is .129, the last is .254

Assignment 6.4: HTML

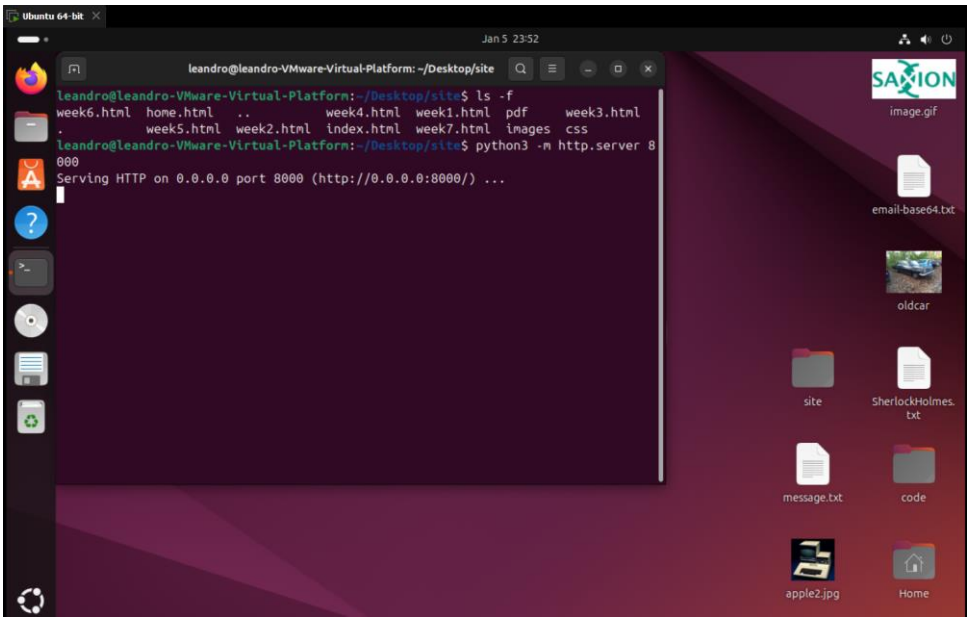
Screenshot IP address Ubuntu VM:



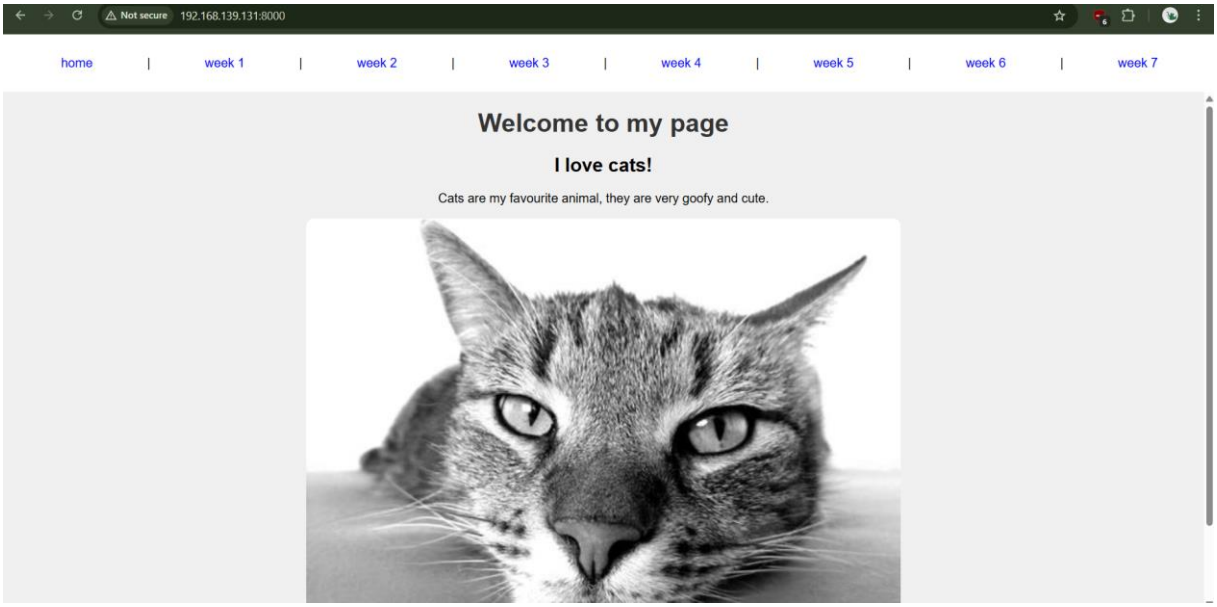
Screenshot of Site directory contents:



Screenshot python3 webserver command:



Screenshot web browser visits your site



Assignment 6.5: Network segment

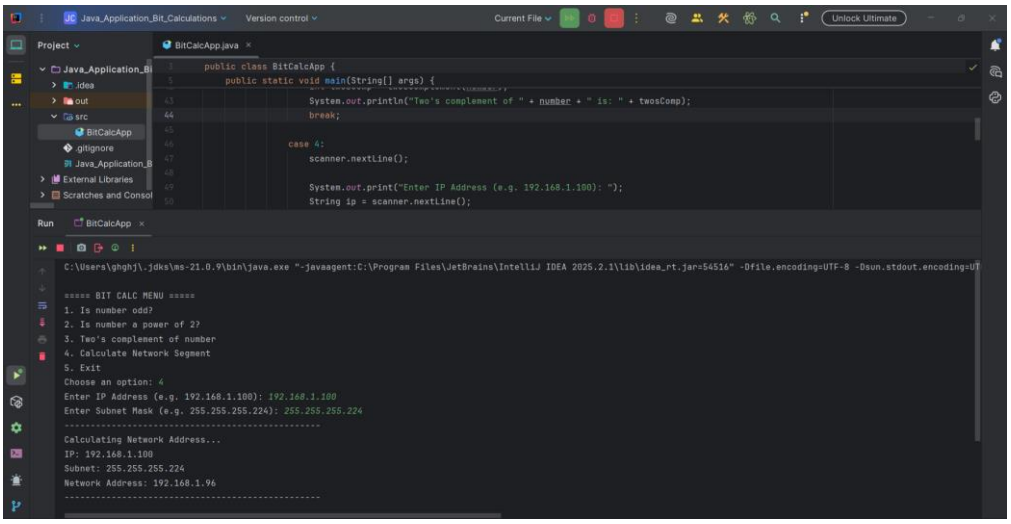
Remember that bitwise java application you've made in week 2? Expand that application so that you can also calculate a network segment as explained in the PowerPoint slides of week 6. Use the bitwise & AND operator. You need to be able to input two Strings. An IP address and a subnet.

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.



Source code:

```
import java.util.Scanner;

public class BitCalcApp {

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

      do {
         System.out.println("\n===== BIT CALC 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");
         System.out.println("5. Exit");
         System.out.print("Choose an option: ");
         choice = scanner.nextInt();

         switch (choice) {
            case 1:
               System.out.print("Enter a number: ");
               number = scanner.nextInt();
               if (isOdd(number))
                  System.out.println(number + " is odd.");
               else
                  System.out.println(number + " is even.");
               break;

            case 2:
               System.out.print("Enter a number: ");
               number = scanner.nextInt();
               if (isPowerOfTwo(number))
                  System.out.println(number + " is a power of 2.");
               else
                  System.out.println(number + " is NOT a power of 2.");
               break;

            case 3:
               System.out.print("Enter a number: ");
               number = scanner.nextInt();
               int twosComp = twosComplement(number);
               System.out.println("Two's complement of " + number + " is: " + twosComp);
               break;

            case 4:
               scanner.nextLine();
```

```
            System.out.print("Enter IP Address (e.g. 192.168.1.100): ");
            String ip = scanner.nextLine();

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

            calculateNetwork(ip, mask);
            break;

        case 5:
            System.out.println("Exiting application...");
            break;

        default:
            System.out.println("Invalid choice. Try again.");
      }

   } while (choice != 5);

   scanner.close();
}

public static boolean isOdd(int n) {
   return (n & 1) == 1;
}

public static boolean isPowerOfTwo(int n) {
   return n > 0 && (n & (n - 1)) == 0;
}

public static int twosComplement(int n) {
   return ~n + 1;
}

public static void calculateNetwork(String ip, String mask) {
   String[] ipParts = ip.split("\\.");
   String[] maskParts = mask.split("\\.");

   if (ipParts.length != 4 || maskParts.length != 4) {
      System.out.println("Invalid format! Please use x.x.x.x");
      return;
   }

   int[] networkParts = new int[4];

   for (int i = 0; i < 4; i++) {
      int ipOctet = Integer.parseInt(ipParts[i]);
      int maskOctet = Integer.parseInt(maskParts[i]);
```

```
            networkParts[i] = ipOctet & maskOctet;
        }

        System.out.println("-----------------------------------------------");
        System.out.println("Calculating Network Address...");
        System.out.println("IP: " + ip);
        System.out.println("Subnet: " + mask);
        System.out.println("Network Address: " + networkParts[0] + "." + networkParts[1] + "." +
networkParts[2] + "." + networkParts[3]);
        System.out.println("-----------------------------------------------");
    }
}
```