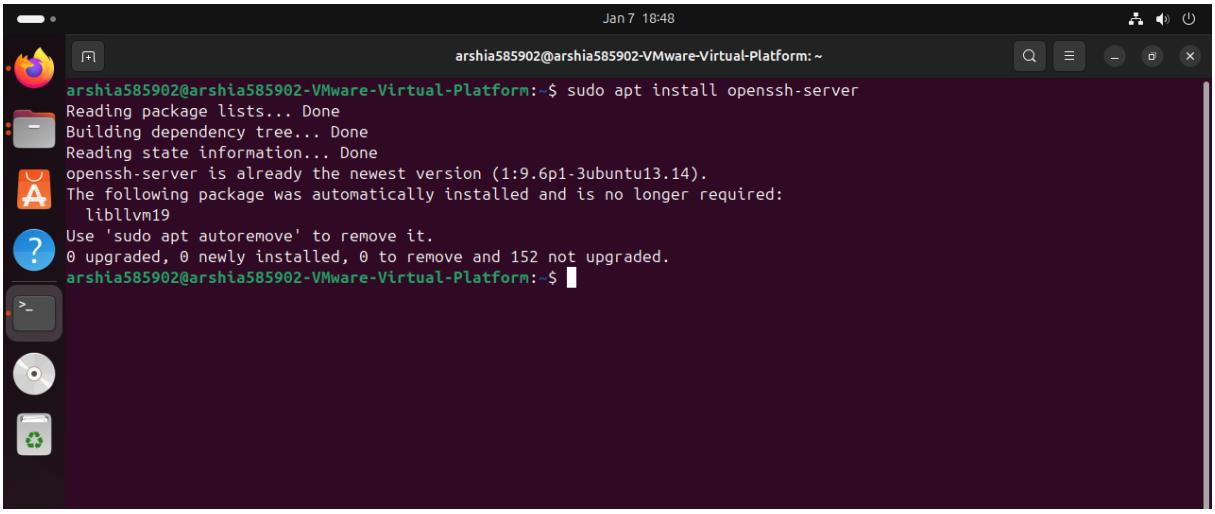
Template Week 6 – Networking

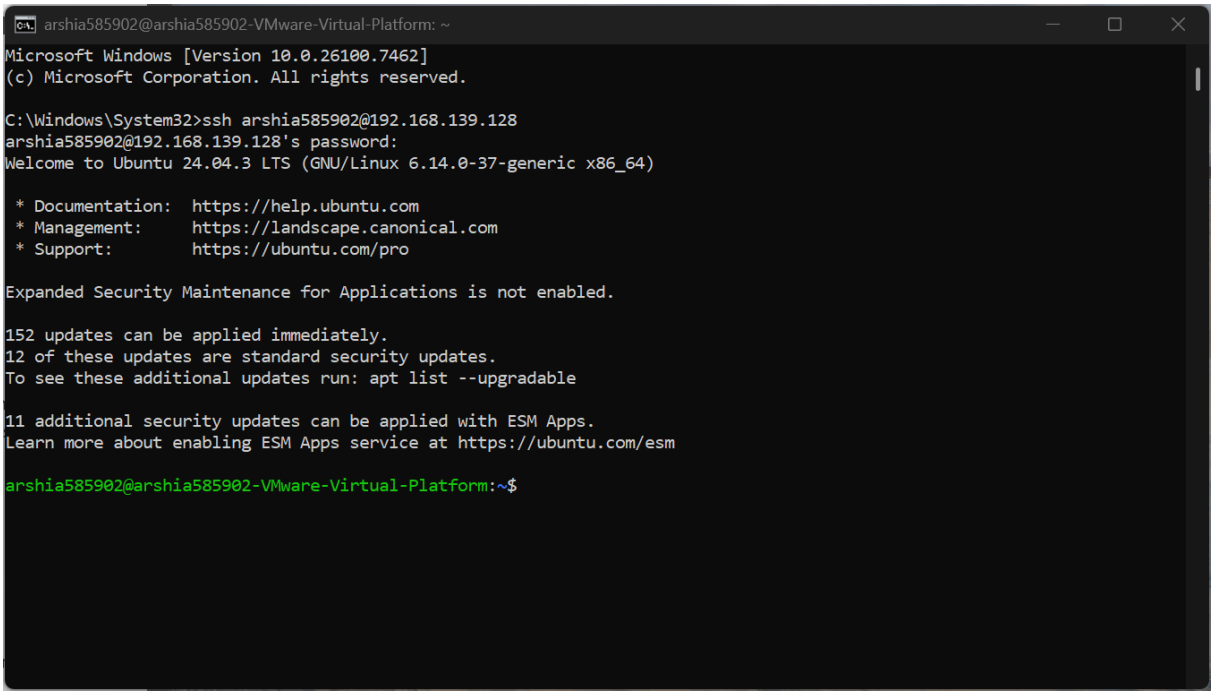
Student number: 585902

Assignment 6.1: Working from home

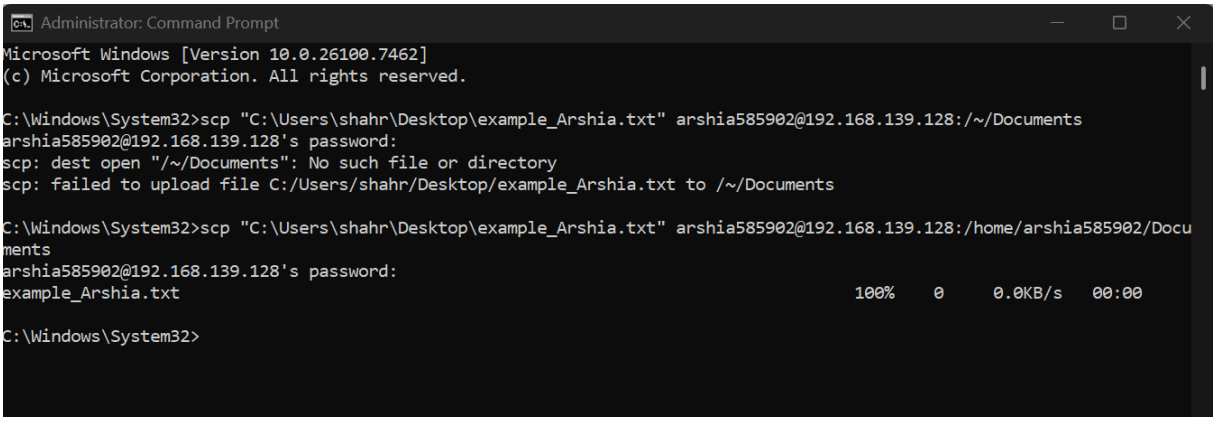
Screenshot installation openssh-server:



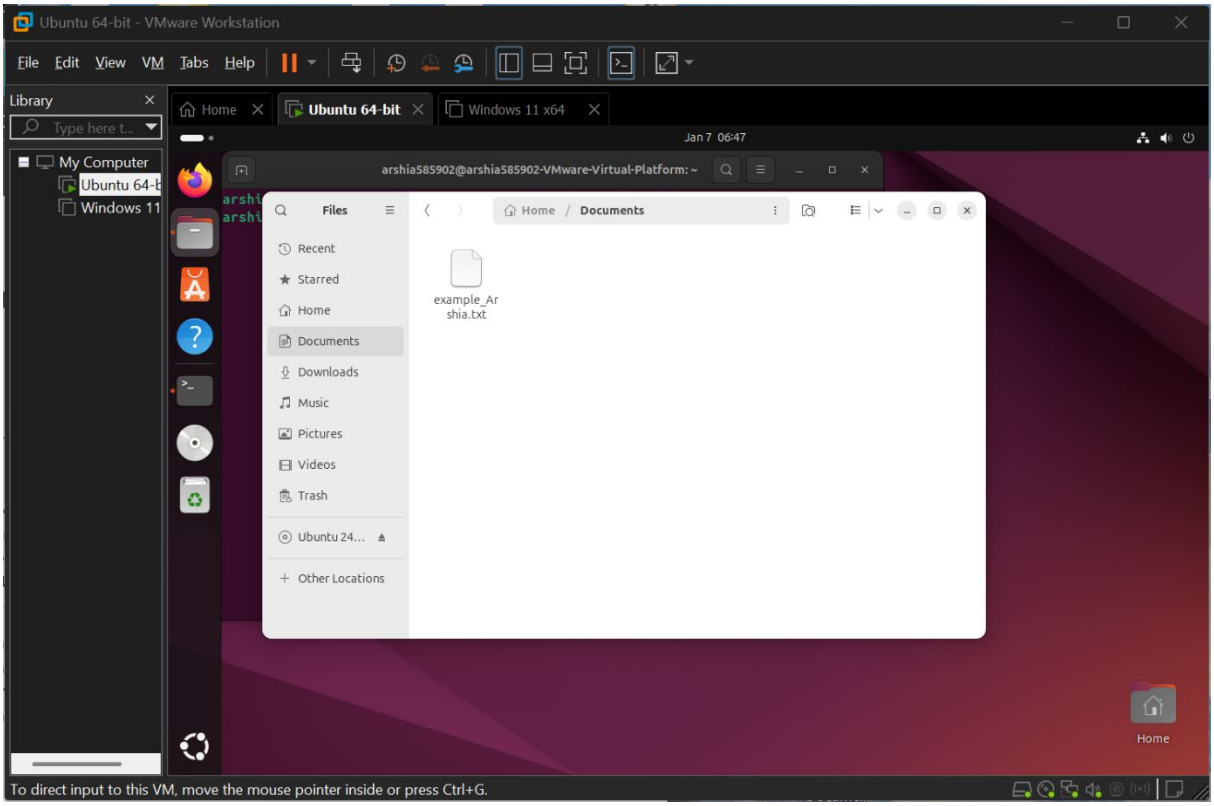
Screenshot successful SSH command execution:



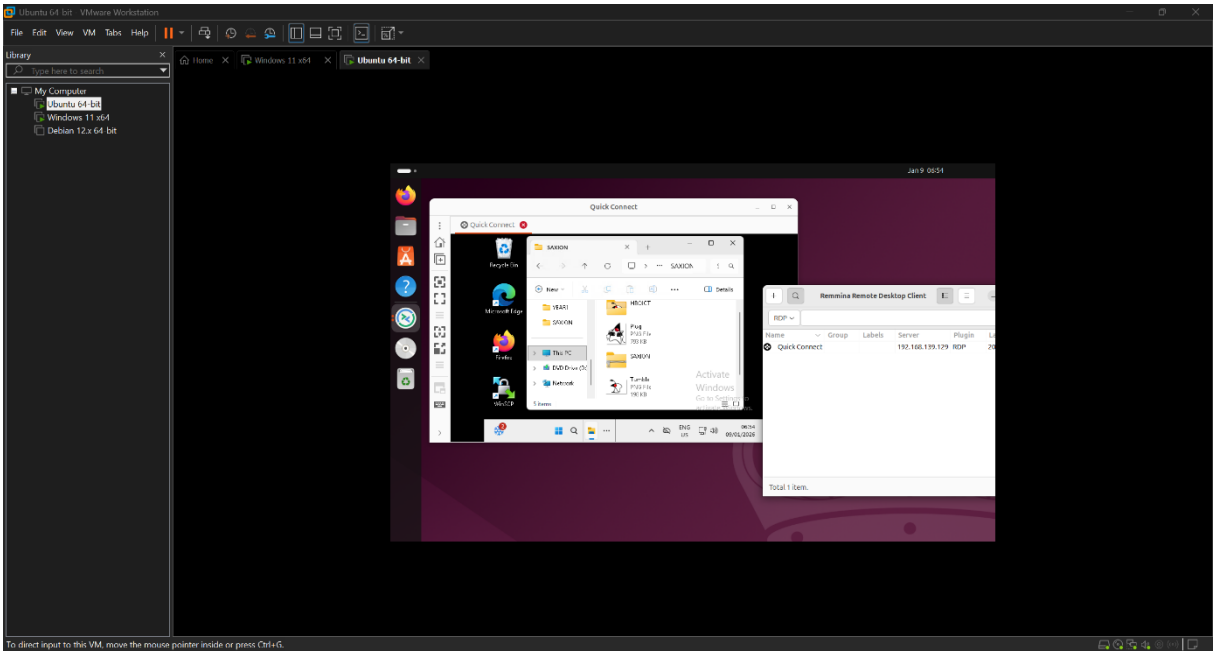
Screenshot successful execution SCP command:



Proof that the file was transferred successfully:



Screenshot remmina:



With some research I found out that these steps should be followed In order of a successful connection:

On windows I went to settings > system > remote desktop and enabled that which resulted in RDP server being enabled on Windows

In the same section there was a place where I could add accounts that I wanted to grant access but since im using admin on Ubuntu, I didn't need to add anything in there; because admins are allowed by default

Then I went to the command prompt on windows and inquired its IP address using "ipconfig" command; which gave : 192.168.139.129

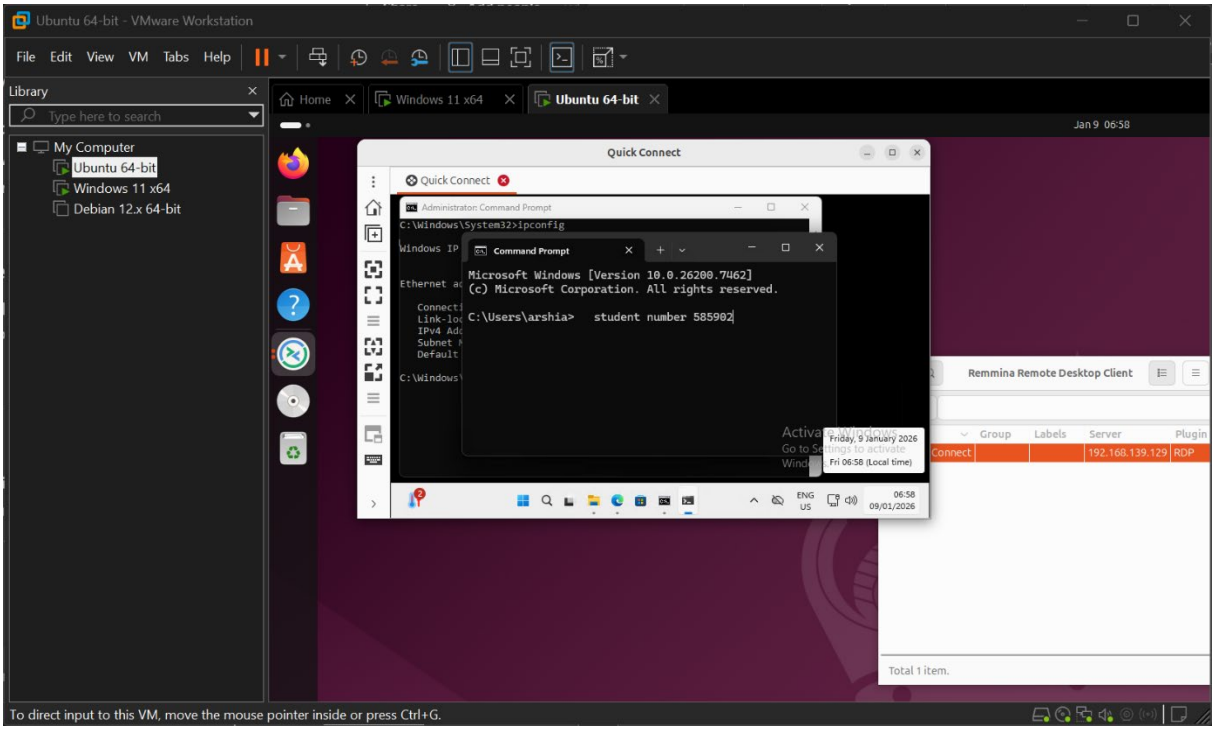
I then proceeded to enter that IP address in Remmina On Ubuntu where it applied

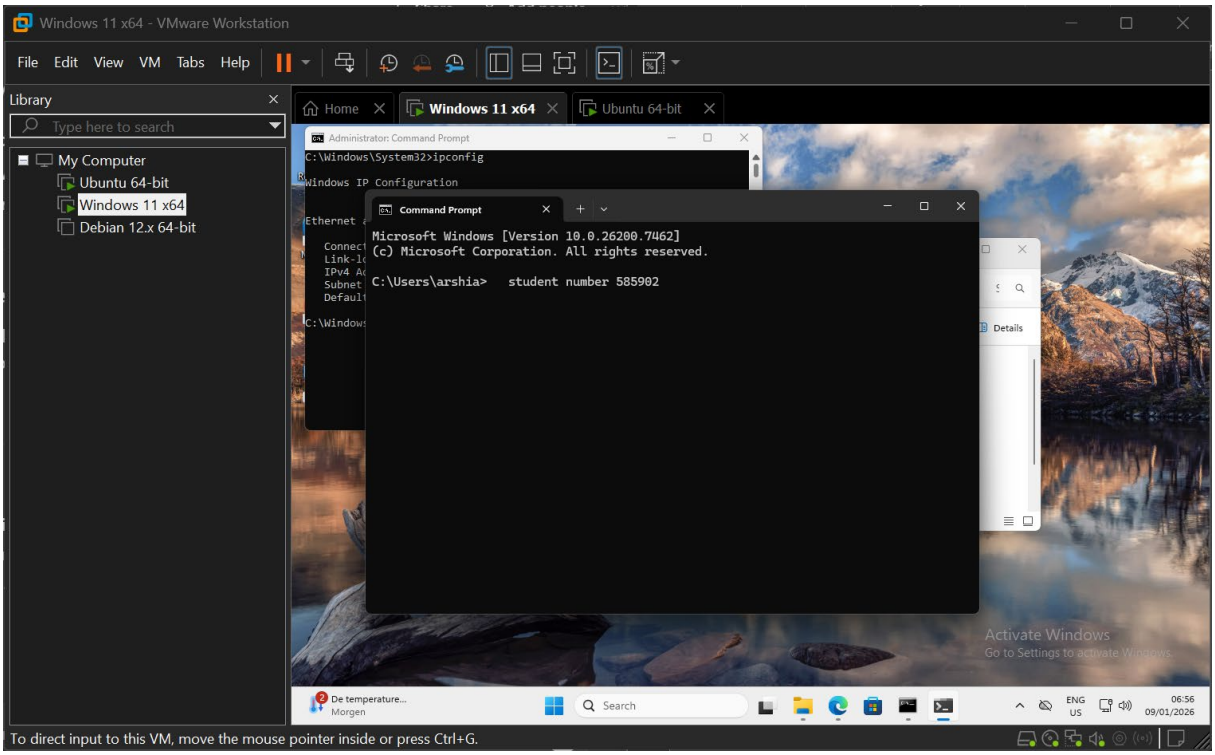
In the configurations of Remmina I entered my Windows password and username(which was the same as ubuntu)

As primary proof, I took screen shots of the configurations I set both on Ubuntu and windows, The setup that resulted in successfully accessing the Windows VM

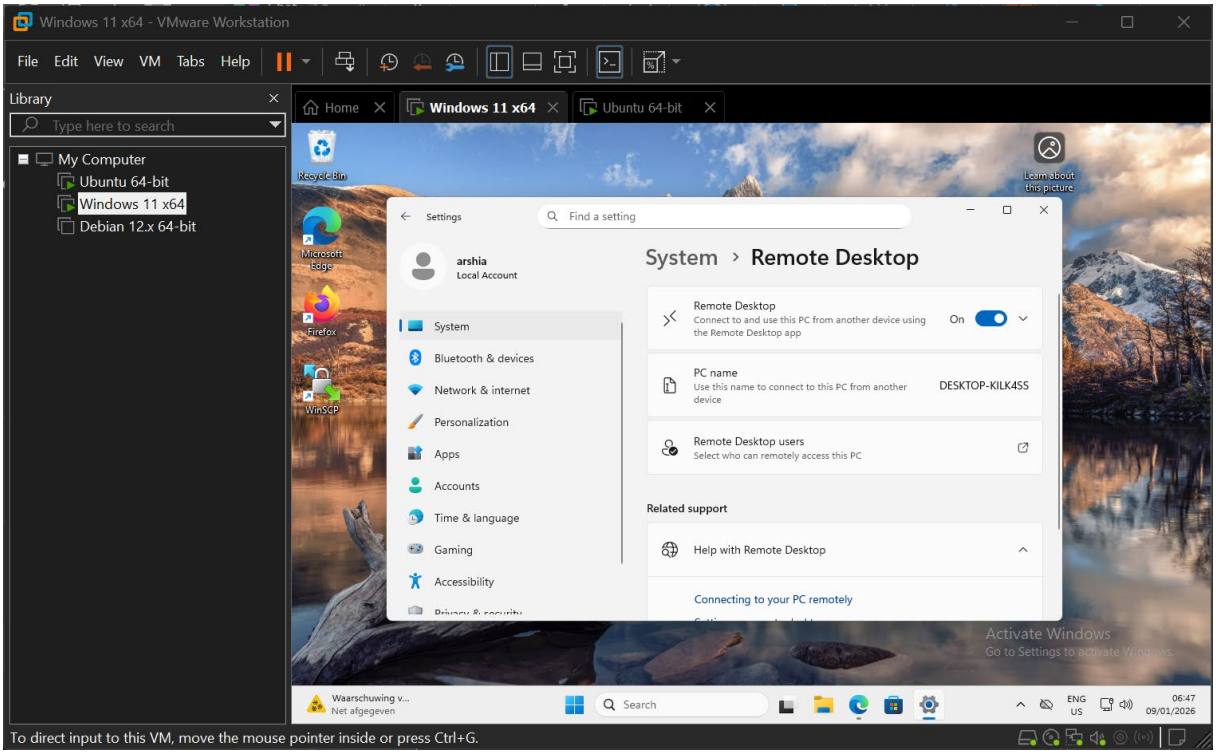
As more proof I tried showing common contents both on the windows VM and ubuntu VM (when connected to the windows). For example, I wrote down my student number in the command prompt on Windows and then showed the same thing on ubuntu. Additionally, I showed the contents of SAXION folder on windows (made in week 5) accessed through ubuntu, using remmina

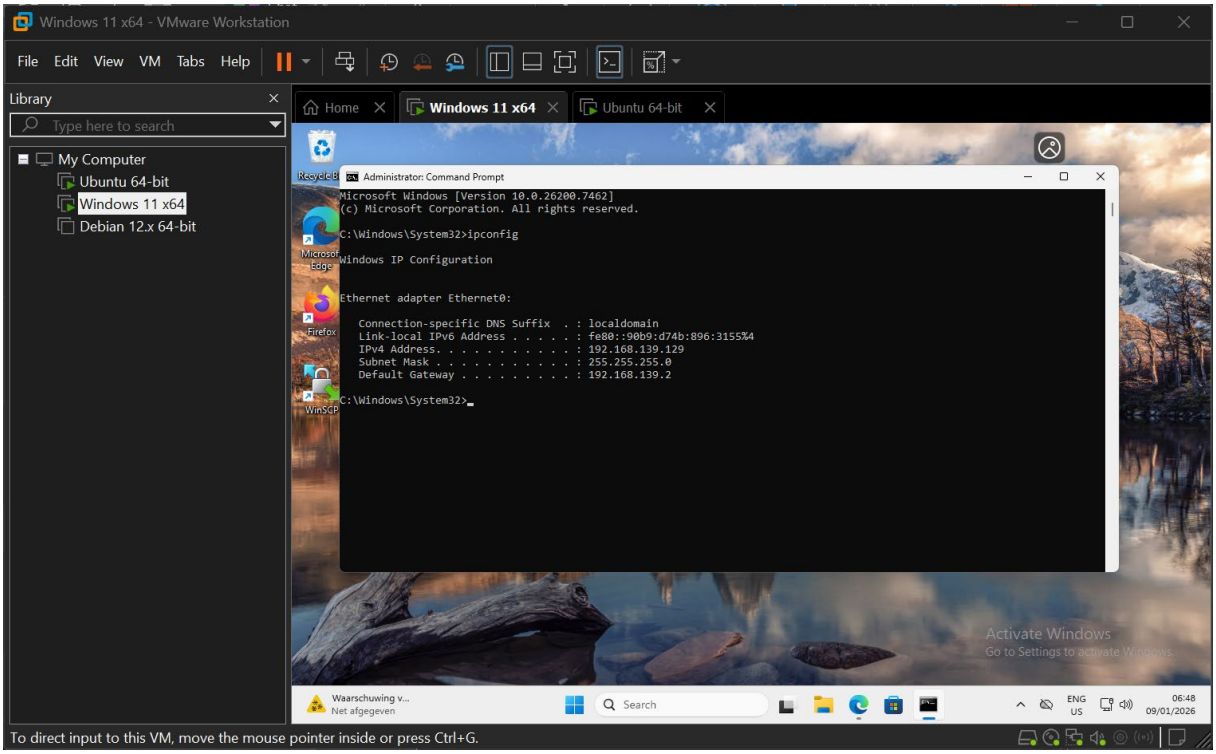
Generally the concept is as follows: remote desktop must be enabled on Windows and both VMs should be connected to the same network. Remmina must connect with RDP protocol, using the IP address of the Windows VM


Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Ubuntu 64-bit
Windows 11 x64
Debian 12.x 64-bit
Home
Windows 11 x64
Ubuntu 64-bit
Jan 9 06:58
Quick Connect
Administrator: Command Prompt
C:\Windows\System32>ipconfig
Command Prompt
Microsoft Windows [Version 10.0.26200.7462]
(c) Microsoft Corporation. All rights reserved.
C:\Users\arshia> student number 585902
Remmina Remote Desktop Client
Group
Labels
Server
Plugin
Connect
192.168.139.129
RDP
Total 1 item.
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

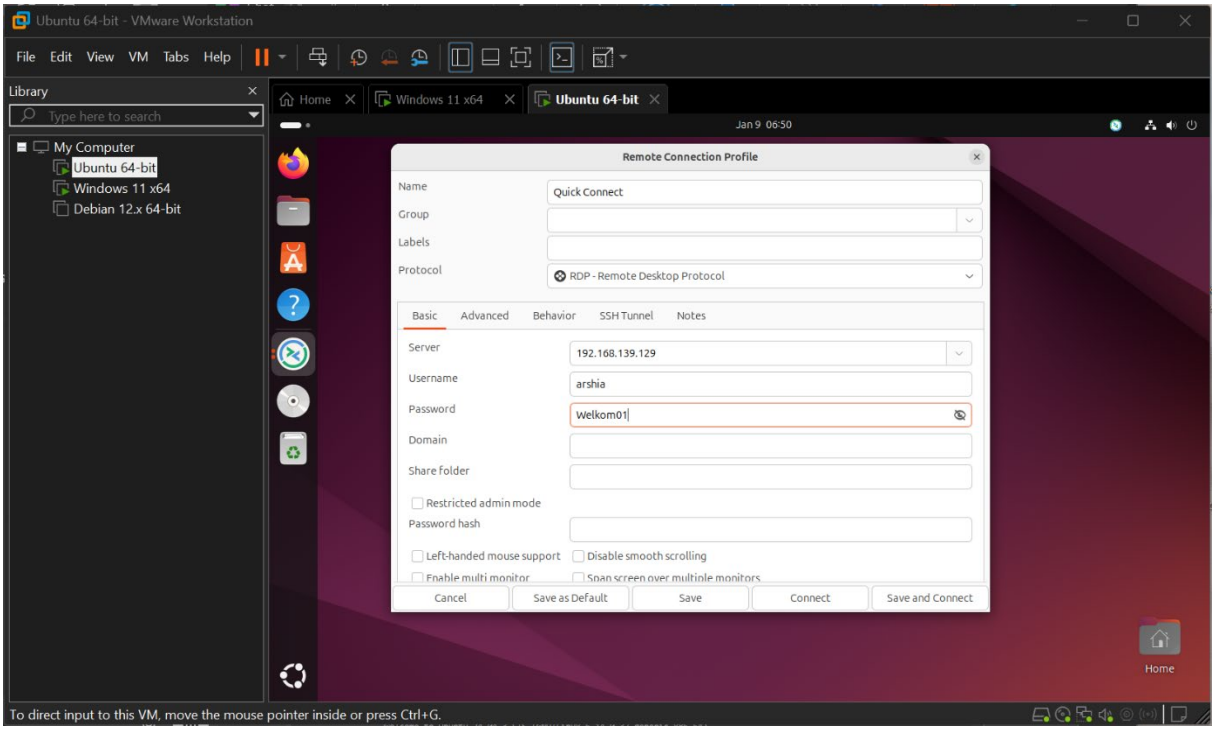

Windows 11 x64 - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Ubuntu 64-bit
Windows 11 x64
Debian 12.x 64-bit
Home
Windows 11 x64
Ubuntu 64-bit
Administrator: Command Prompt
C:\Windows\System32>ipconfig
Windows IP Configuration
Command Prompt
Microsoft Windows [Version 10.0.26200.7462]
(c) Microsoft Corporation. All rights reserved.
C:\Users\arshia> student number 585902
Details
Activate Windows
Go to Settings to activate Windows.
Search
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

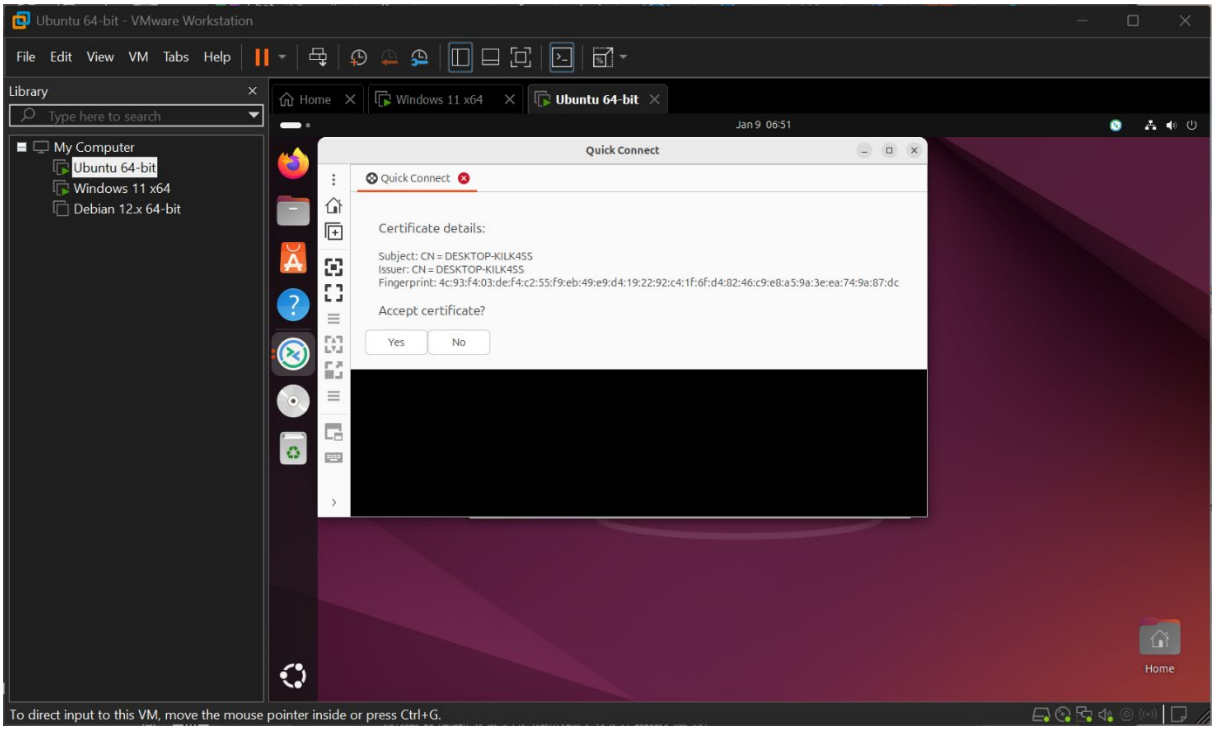
Applied settings :





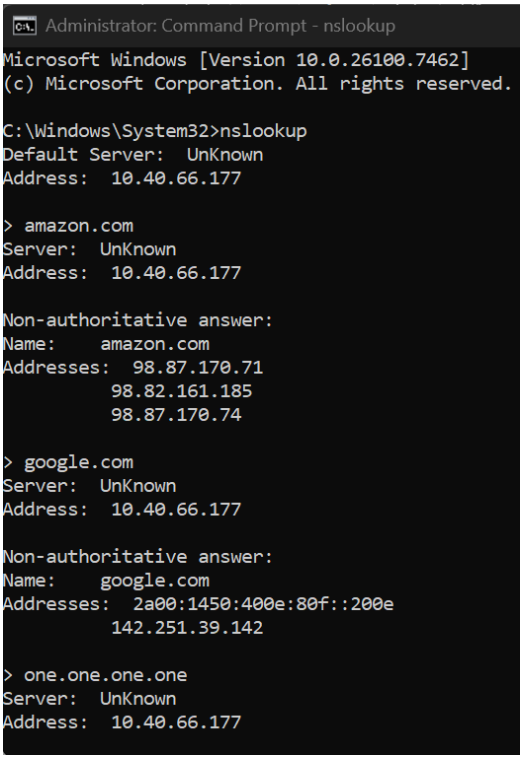
Remmina configuration:

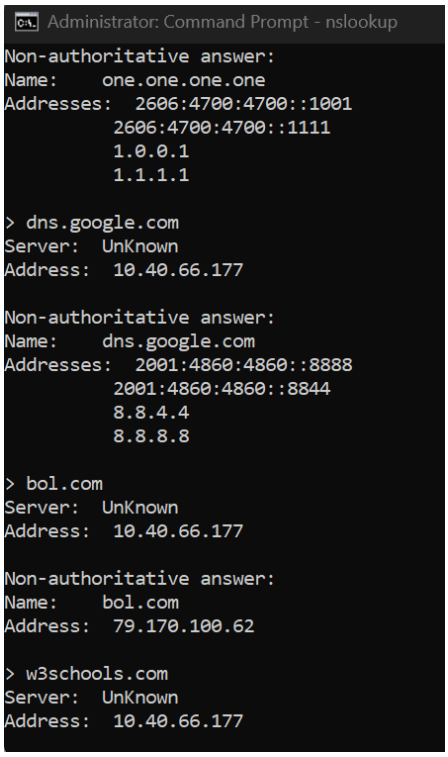




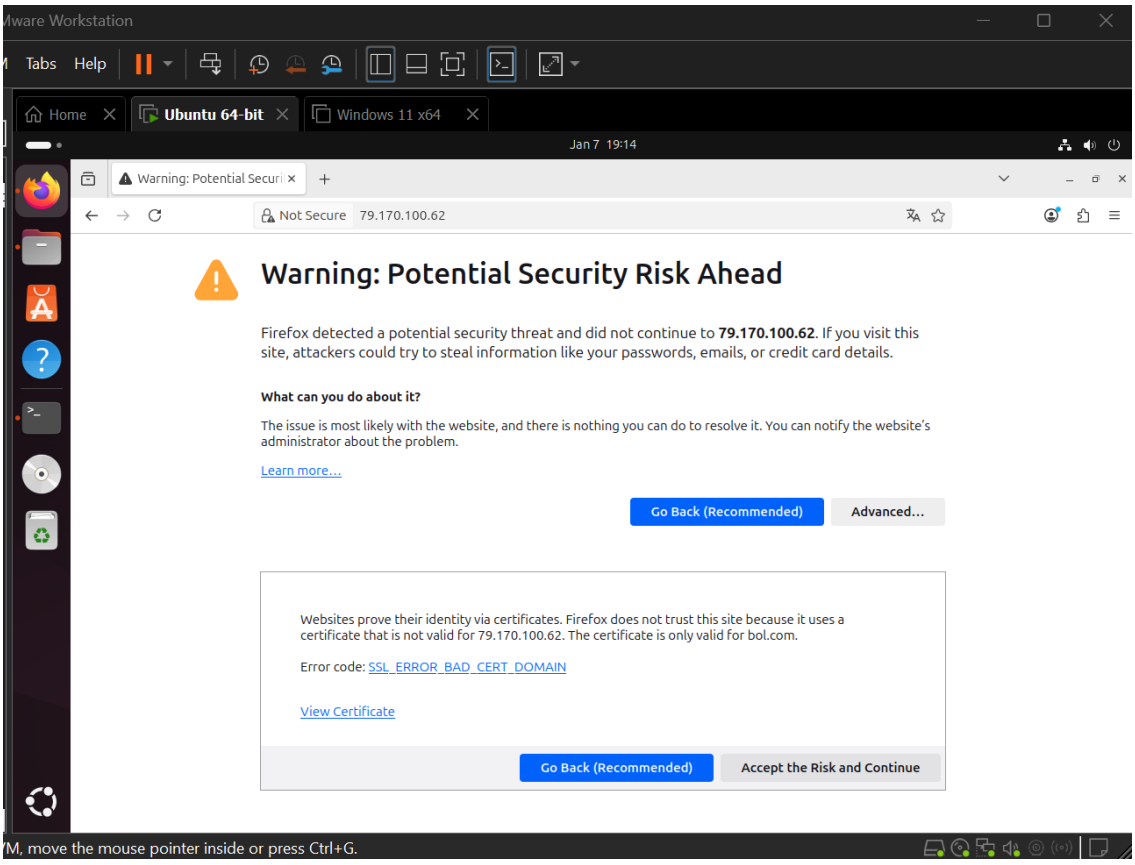
Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:





Screenshot website visit via IP address:



Assignment 6.3: subnetting

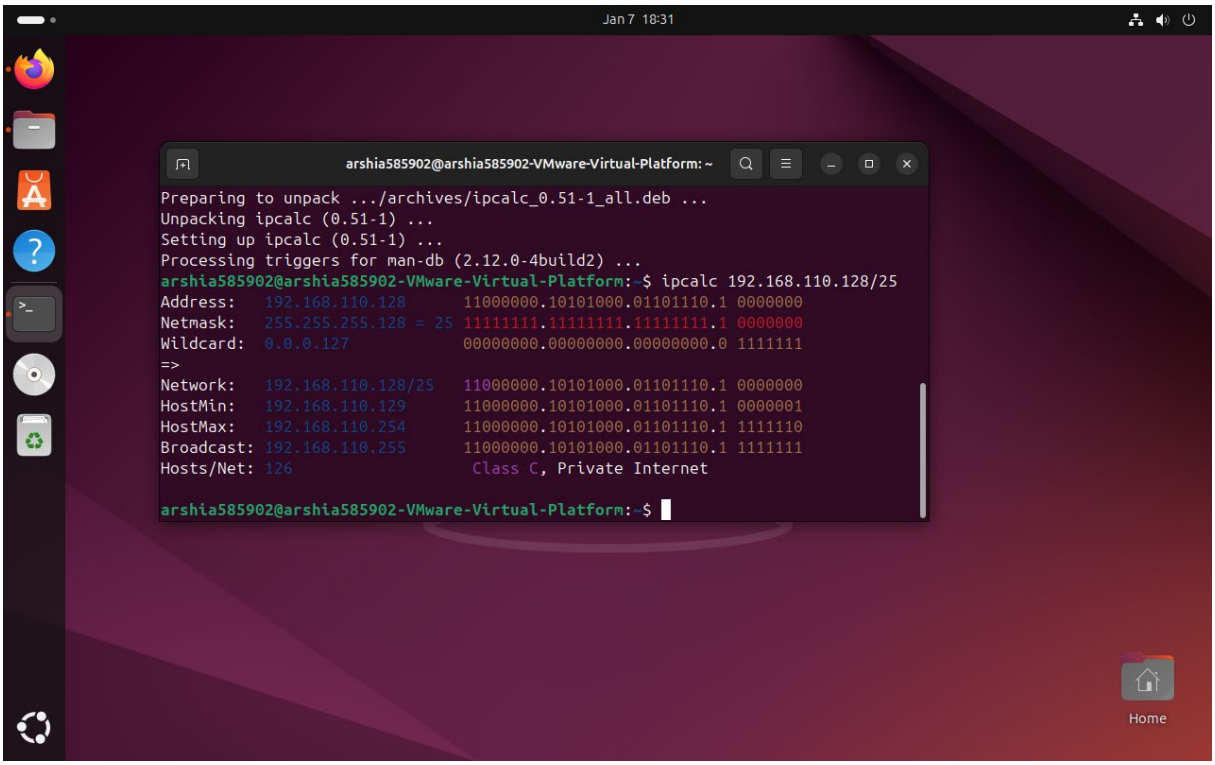
How many IP addresses are in this network configuration 192.168.110.128/25?

128 addresses, 126 of which usable for hosts because the first one (being network ip) and the last one (being broadcast ip) are not usable for that – [calculation explained bellow]

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

From 192.168.110.129/25 to 192.168.110.254/25

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

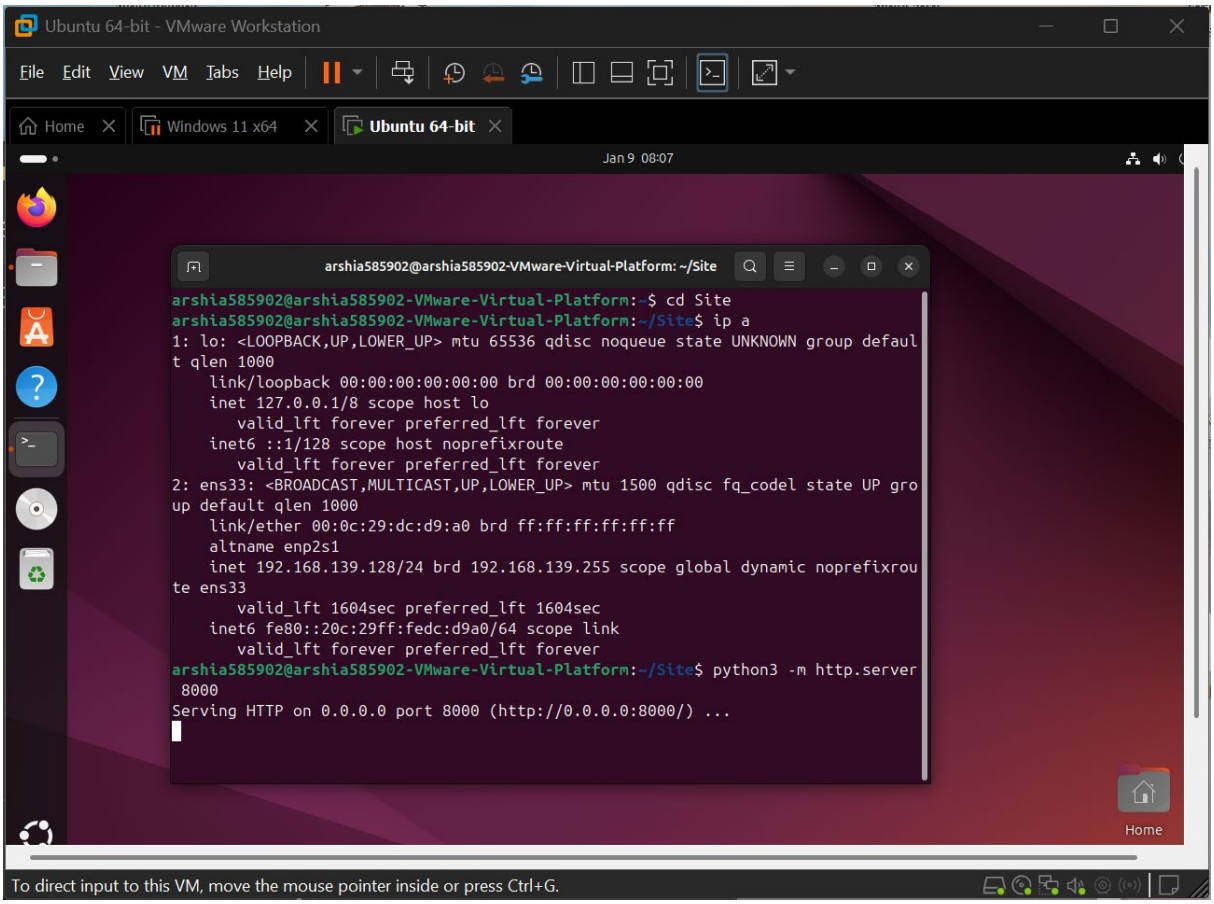


Explain the above calculation in your own words.

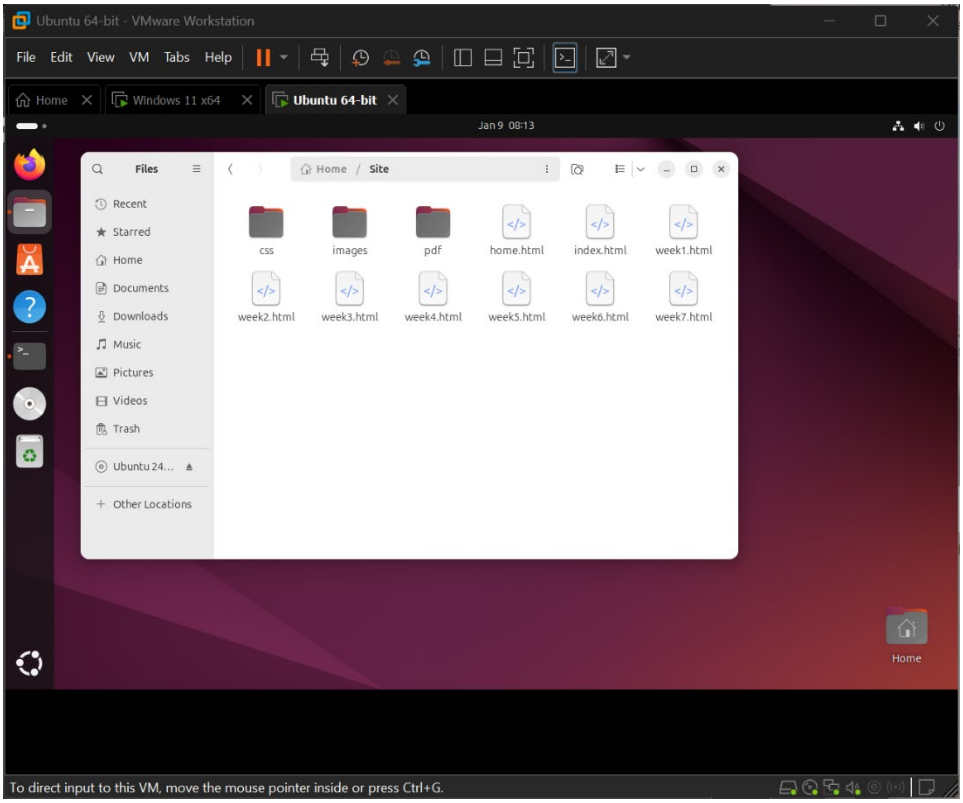
If it wasn't a '/25' subnet, there would have been four octets or groups of 8 bits which together would have formed 32 bits which would have been 2^{32} addresses. But now, 25 bit is subtracted from 32 since those 2^{25} addresses are reserved for the network and can't be given to hosts. that leaves us with 7 bits or 2^{7} addresses to be given to hosts; (two of which are reduced again, for network and broadcast). That '/25' subnet mask makes the block size 128 (or 2^{7}) and the boundaries of that subnet should be multiples of the block size which gives us two possible subnets per last octet, 0-127 and 128-255. the last octet determines the network address and that address is the beginning of the range. Since that says 128 then it means that the range should be from 128 to 255 but the first one and the last one cant be used; therefore what actually goes to hosts are from ~.129/25 to ~.254/25

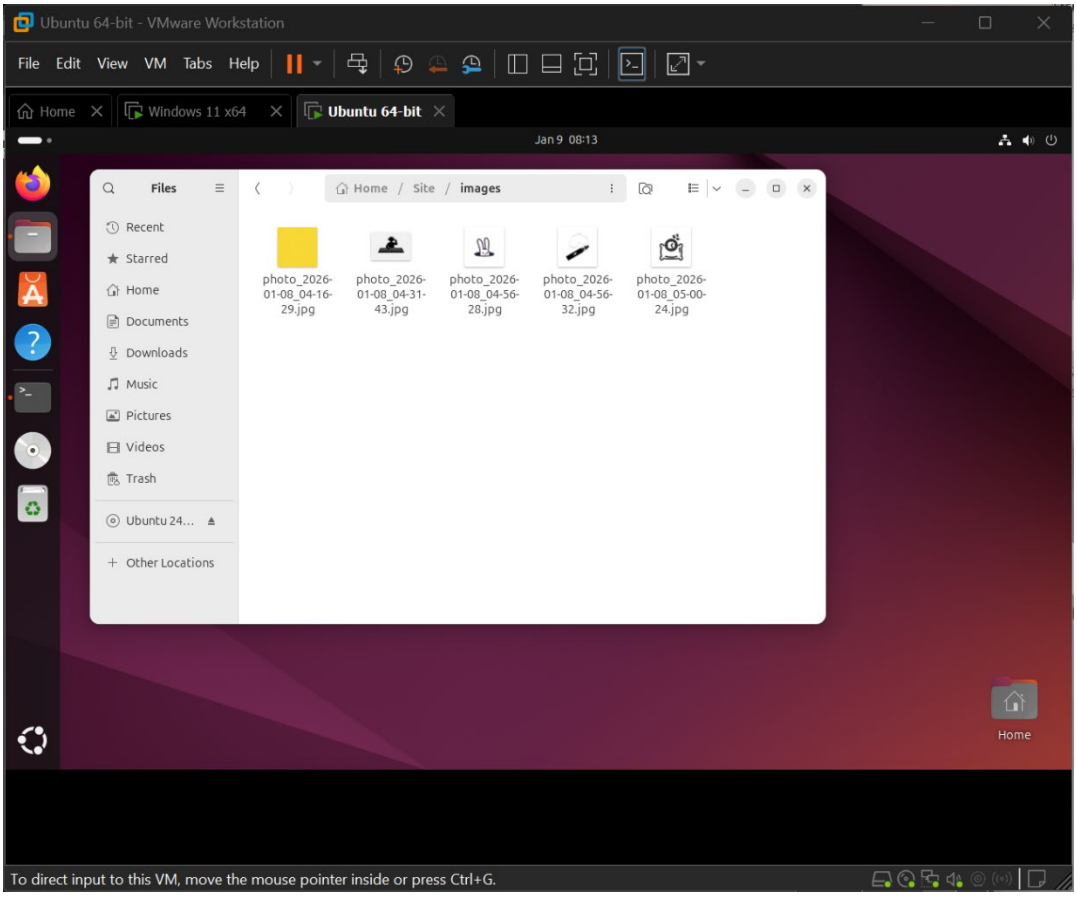
Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

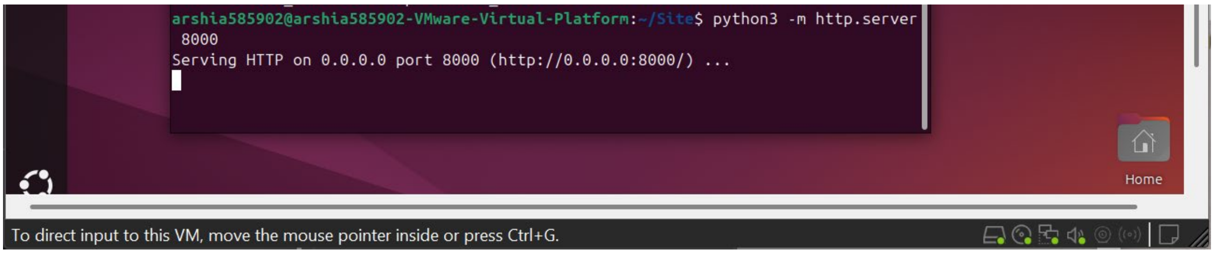


Screenshot of Site directory contents:

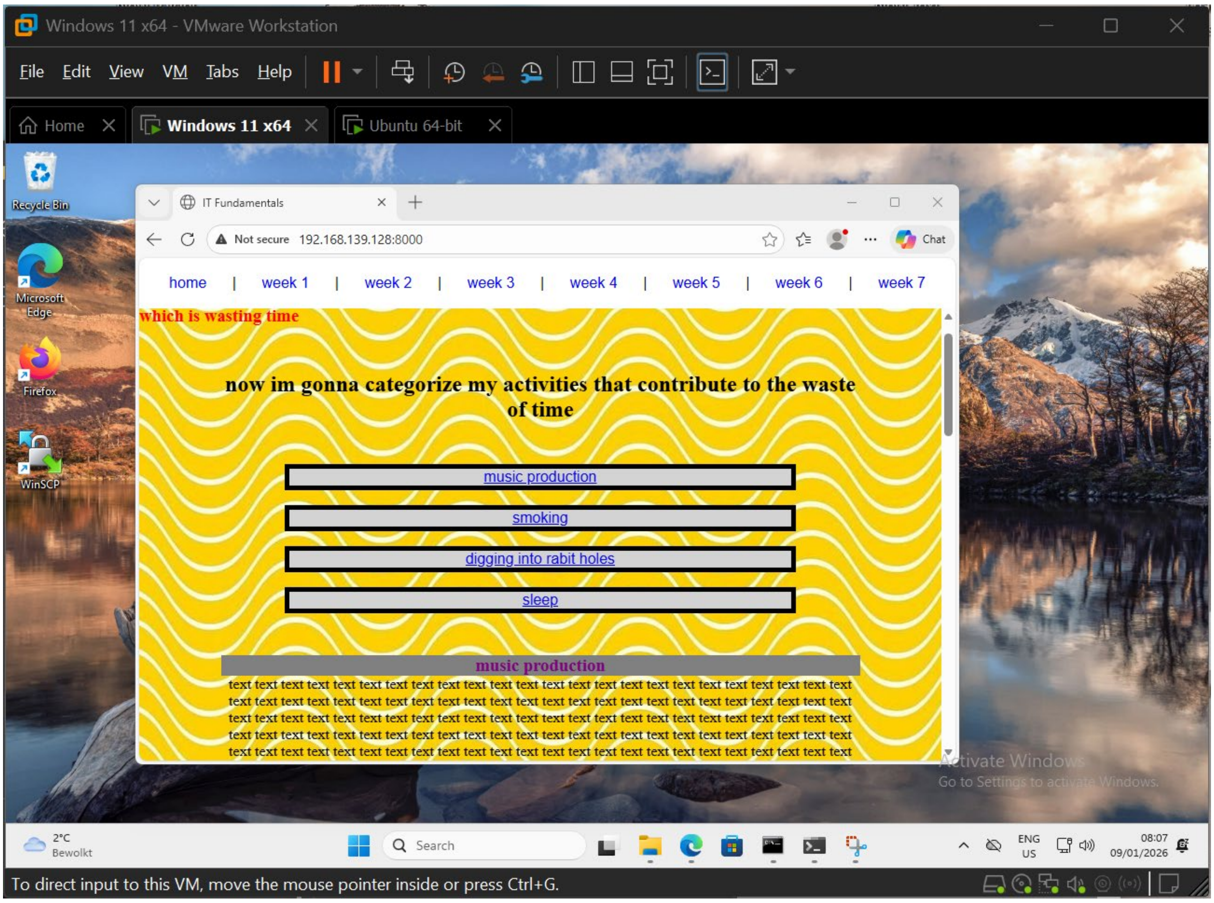




Screenshot python3 webserver command:



Screenshot web browser visits your site



Assignment 6.5: Network segment

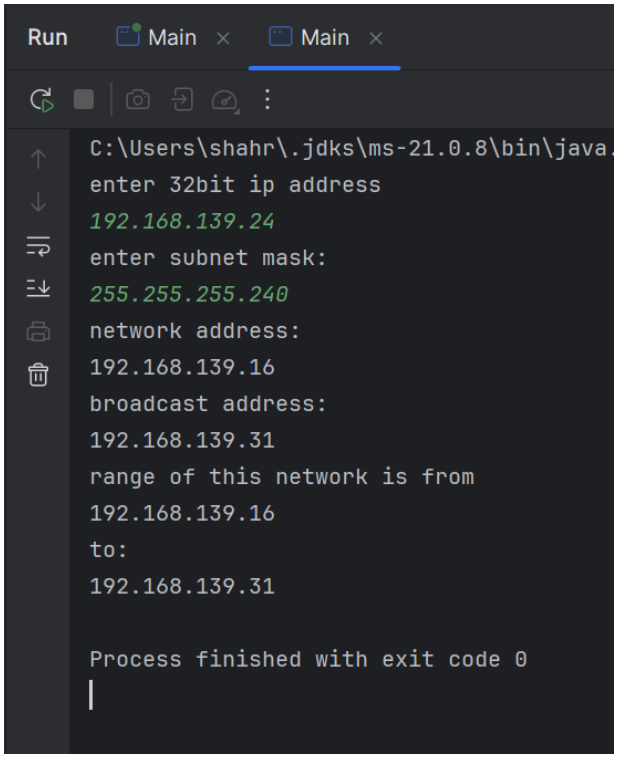
Remember that bitwise java application you've made in week 2? Expand that application so that you can also calculate a network segment as explained in the PowerPoint slides of week 6. Use the bitwise & AND operator. You need to be able to input two Strings. An IP address and a subnet.

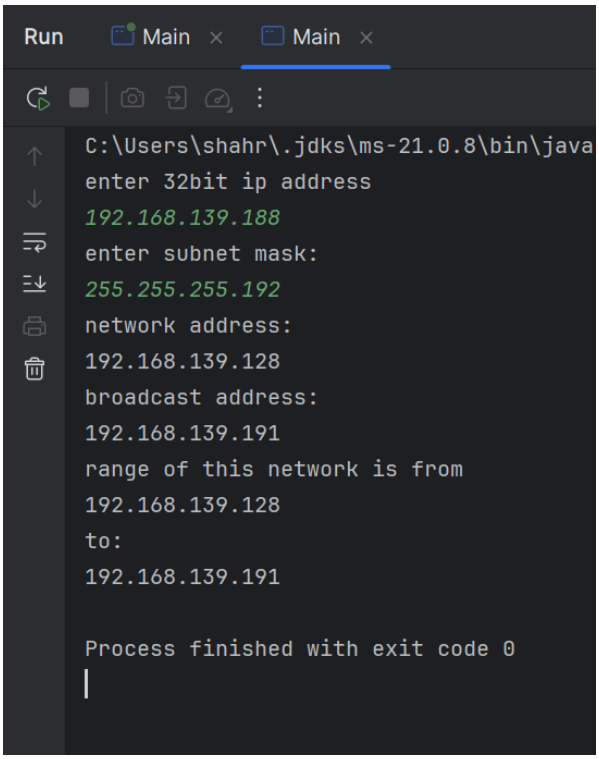
IP: 192.168.1.100 and subnet: 255.255.255.224 for /27

```
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.





```java
import java.util.ArrayList;
import  java.util.Scanner;
public class Main {
    static Scanner scanner = new Scanner(System.in);
    public static void main(String[] args){
        System.out.println("enter 32bit ip address");
        String ipString = scanner.nextLine().trim();
        int[] ip = addressToArray(ipString);
        System.out.println("enter subnet mask: ");
        String sMask = scanner.nextLine().trim();
        int[] subnetMask = addressToArray(sMask);
        int[] network = new int[4];
        int[] broadcast = new int[4];
        for(int i = 0; i<4; i++){
            network[i] = subnetMask[i] & ip[i];
            broadcast[i] = (~subnetMask[i] & 255) | network[i];
        }
        System.out.println("network address: ");
        System.out.println(arrayToAddress(network));
        System.out.println("broadcast address: ");
        System.out.println(arrayToAddress(broadcast));
        System.out.println("range of this network is from ");
        System.out.println(arrayToAddress(network));
        System.out.println("to: ");
        System.out.println(arrayToAddress(broadcast));
    }

    static int[] addressToArray(String addressString){
        int[] temp = new int[]{4,4,4,4};
        int j =0;
        int begin = 0;
        for (int i = 0; i < addressString.length(); i++) {
            if (addressString.charAt(i) == '.') {
                temp[j] =
Integer.parseInt(addressString.substring(begin, i));
                j++;
                begin = i+1;
            }
        }
        temp[j] = Integer.parseInt(addressString.substring(begin));
        return temp;
    }

    static String arrayToAddress(int[] address){
        String addressString = "";
        for(int i = 0; i < address.length; i++){
            addressString = addressString + address[i] + ".";
        }
        addressString = addressString.substring(0,
addressString.length()-1);
        return addressString;
    }
}
```

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