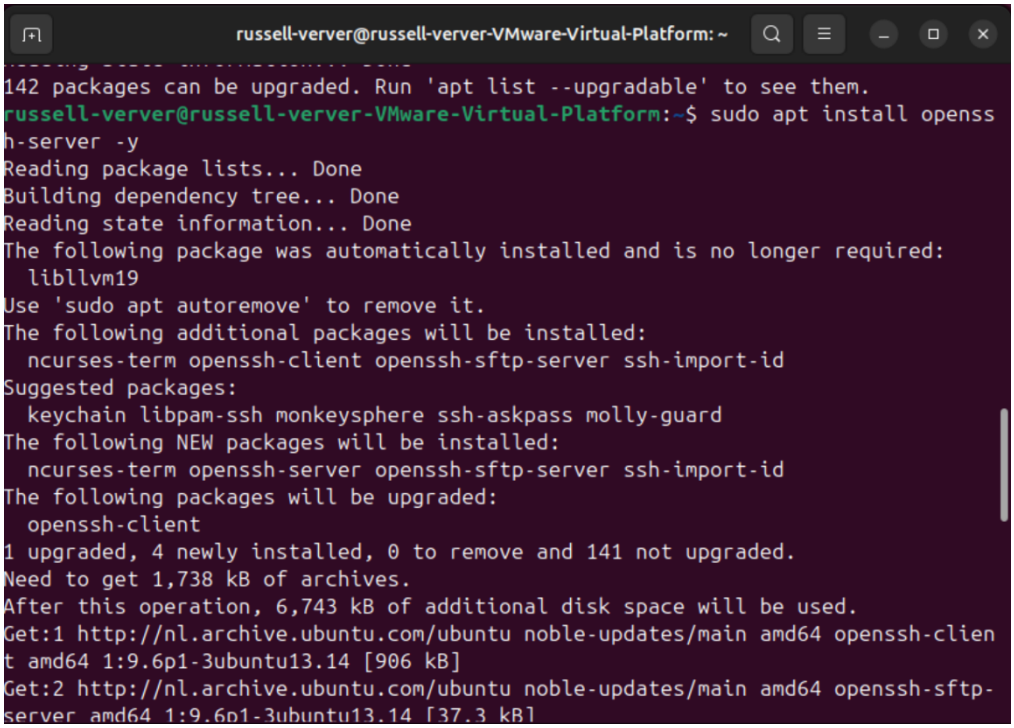
Template Week 6 – Networking

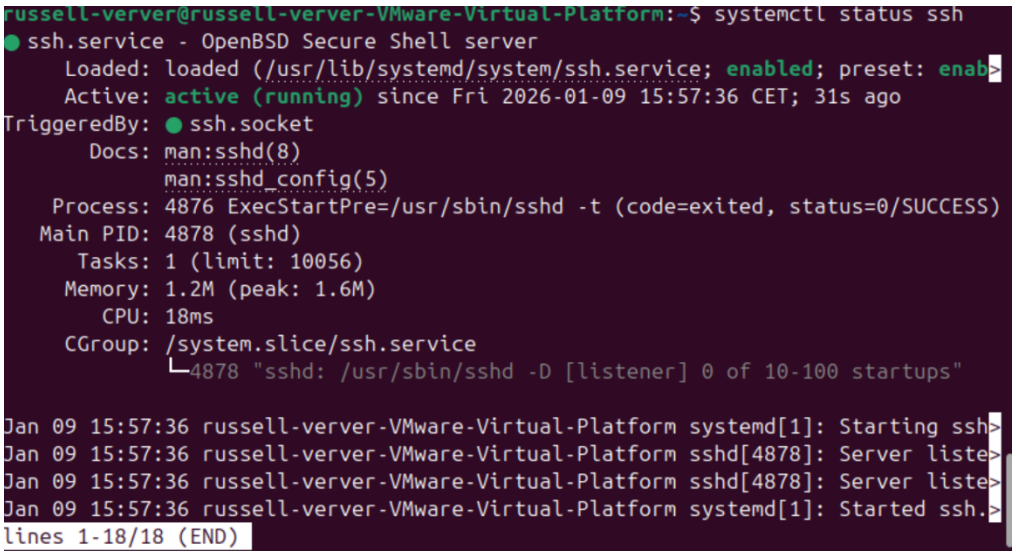
Student number:589768

Assignment 6.1: Working from home

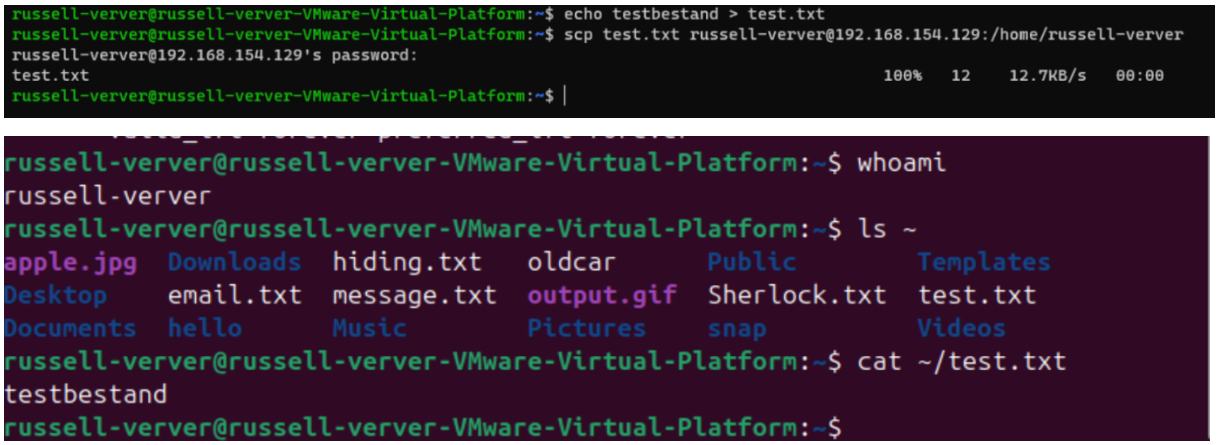
Screenshot installation openssh-server:



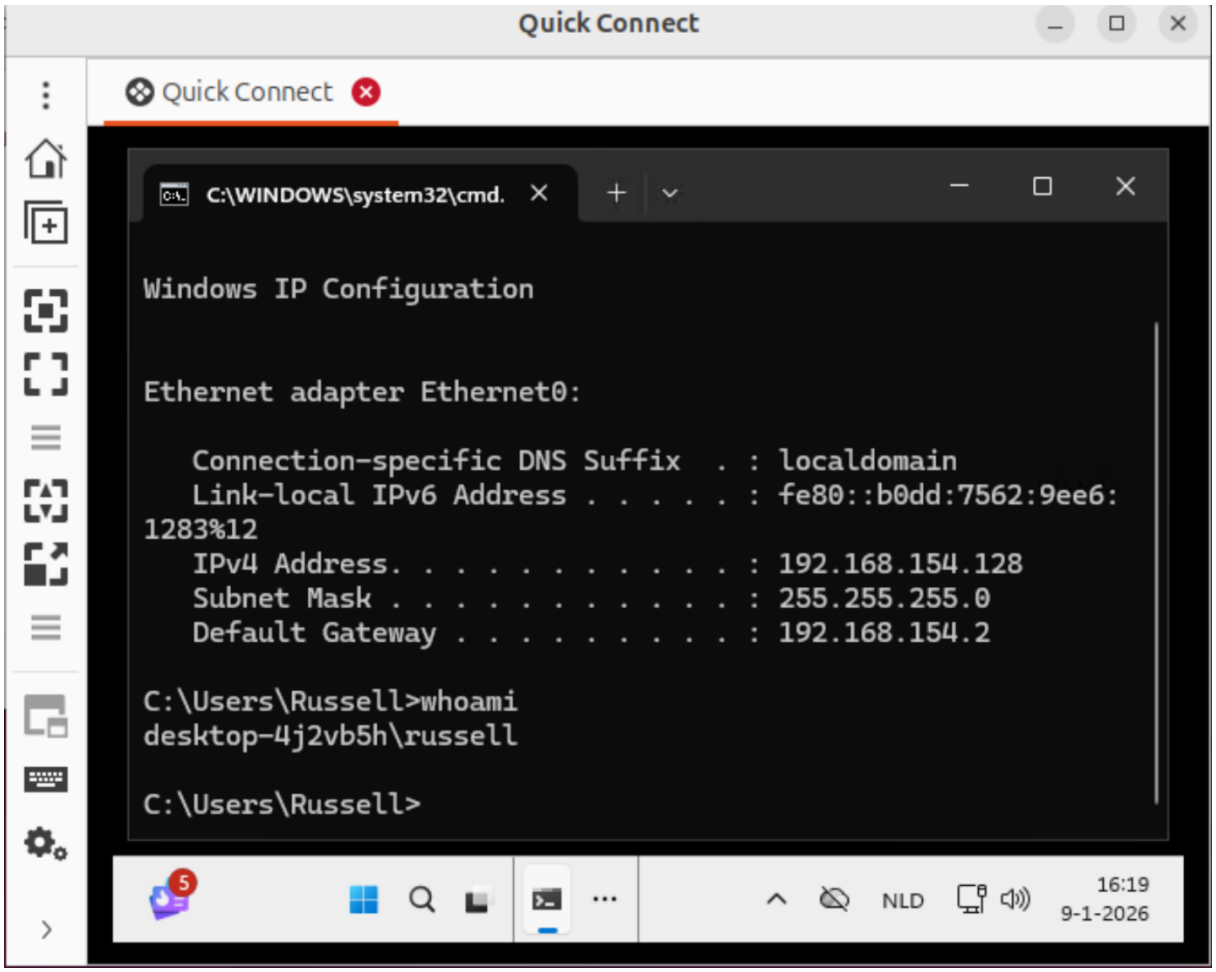
Screenshot successful SSH command execution:



Screenshot successful execution SCP command:



Screenshot remmina:

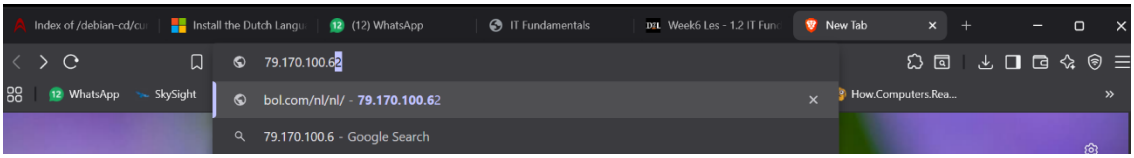


Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

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

Screenshot website visit via IP address:



Assignment 6.3: subnetting

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

Het Antwoord is 128,want het ip adress beststaat uit 25 bits eengewoon ip4 ip adress bestaat uit 32 bits als je 32-25 hou je 7bit over als je dit dan berekend door 2 tot de macht 7 te doen kom je op 128 ip-adressen.

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

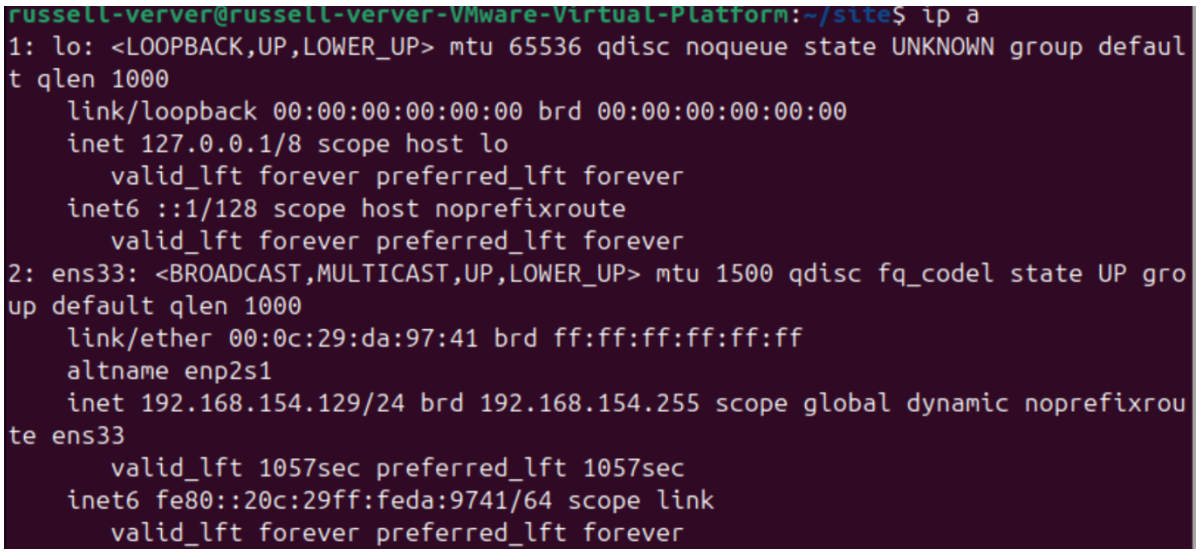
als je een ip-adess hebt, bijvoorbeeld het ip-adress van hier boven 192.168.110.128 dan zit de range tussen 192.168.110.255 en 192.110.128 als je deze van elkaar aftrekt kom je op 128 ip adressen 2 van deze ip adressen zijn niet bruikbaar dus 126 ip adressen

Check your two previous answers with this Linux command:

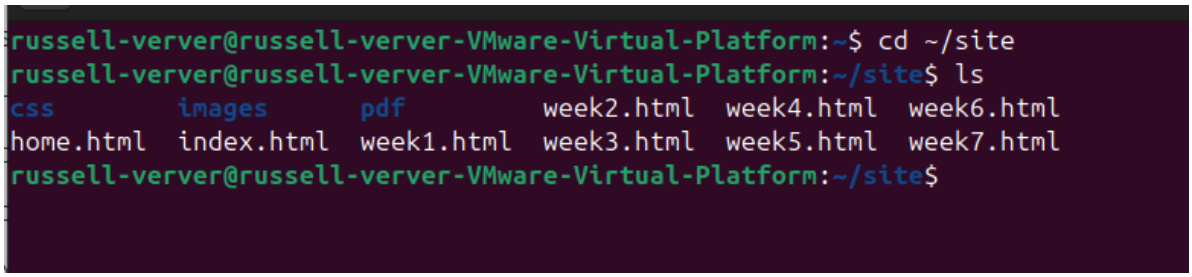
```
russell-verver@russell-verver-VMware-Virtual-Platform:~$  ipcalc 192.168.110.128
/25
Address:   192.168.110.128      11000000.10101000.01101110.1 0000000
Netmask:   255.255.255.128 = 25 11111111.11111111.11111111.1 0000000
Wildcard:  0.0.0.127            00000000.00000000.00000000.0 1111111
=>
Network:   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, Private Internet

russell-verver@russell-verver-VMware-Virtual-Platform:~$
```

Assignment 6.4: HTML

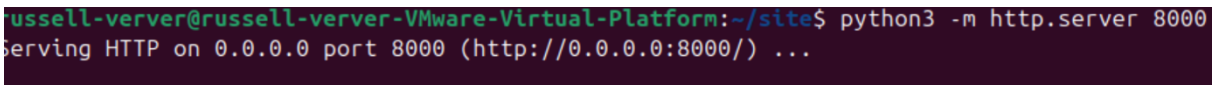
Screenshot IP address Ubuntu VM:



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 (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.

```java
import java.util.Scanner;

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

        System.out.println("=== Network Segment Calculator ===");
        System.out.print("Enter IP address (e.g., 192.168.1.100): ");
        String ipAddress = scanner.nextLine();

        System.out.print("Enter subnet mask (e.g., 255.255.255.224): ");
        String subnetMask = scanner.nextLine();

        String[] ip = ipAddress.split("\\.");
        String[] mask = subnetMask.split("\\.");

        int[] ipOctet = new int[4];
        int[] maskOctet = new int[4];
        int[] network = new int[4];
        int[] broadcast = new int[4];

        for (int i = 0; i < 4; i++) {
            ipOctet[i] = Integer.parseInt(ip[i]);
            maskOctet[i] = Integer.parseInt(mask[i]);
            network[i] = ipOctet[i] & maskOctet[i];
            broadcast[i] = network[i] | (~maskOctet[i] & 0xFF);
        }

        System.out.println("\nCalculate the network segment");
        System.out.print("IP Address:   ");
        for (int i = 0; i < 4; i++) {
            System.out.print(String.format("%8s",
Integer.toBinaryString(ipOctet[i])).replace(' ', '0'));
            if (i < 3) System.out.print(".");
        }
```

```
        System.out.println();

        System.out.print("Subnet Mask:   ");
        for (int i = 0; i < 4; i++) {
            System.out.print(String.format("%8s",
Integer.toBinaryString(maskOctet[i])).replace(' ', '0'));
            if (i < 3) System.out.print(".");
        }
        System.out.println();

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

        System.out.print("Network Addr: ");
        for (int i = 0; i < 4; i++) {
            System.out.print(String.format("%8s",
Integer.toBinaryString(network[i])).replace(' ', '0'));
            if (i < 3) System.out.print(".");
        }
        System.out.println();

        System.out.print("\nThis gives ");
        for (int i = 0; i < 4; i++) {
            System.out.print(network[i]);
            if (i < 3) System.out.print(".");
        }
        System.out.println(" in decimal as the network address.");

        int hostBits = 0;
        for (int i = 0; i < 4; i++) {
            for (int bit = 0; bit < 8; bit++) {
                if ((maskOctet[i] & (1 << bit)) == 0)
                    hostBits++;
            }
        }

        int totalHosts = (int) Math.pow(2, hostBits);
        System.out.println("For a /" + (32 - hostBits) + " subnet, each
segment has " + totalHosts + " IP addresses (2^" + hostBits + ").");

        System.out.print("The range of this network segment is from ");
        for (int i = 0; i < 4; i++) {
            System.out.print(network[i]);
            if (i < 3) System.out.print(".");
        }
        System.out.print(" to ");
        for (int i = 0; i < 4; i++) {
            System.out.print(broadcast[i]);
            if (i < 3) System.out.print(".");
        }
        System.out.println(".");

        scanner.close();
    }
}
```

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