

Template Week 6 – Networking

Student number:

Assignment 6.1: Working from home

Screenshot installation openssh-server:

```
ubuntu@ubuntu2404:~/Documents$ sudo systemctl start ssh
ubuntu@ubuntu2404:~/Documents$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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 pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:dc:89:90 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.139.131/24 brd 192.168.139.255 scope global dynamic noprefixroute ens33
        valid_lft 1129sec preferred_lft 1129sec
ubuntu@ubuntu2404:~/Documents$ ssh ubuntu@192.168.139.131
The authenticity of host '192.168.139.131 (192.168.139.131)' can't be established.
ED25519 key fingerprint is SHA256:6u5vPdqrR9X2XtbR83HSI092UBrLPSN13Yx2o4SuaHc.
This key is not known by any other names.
```

Screenshot successful SSH command execution:

```
ubuntu@ubuntu2404:~/Documents$ ssh ubuntu@192.168.139.131
The authenticity of host '192.168.139.131 (192.168.139.131)' can't be established.
ED25519 key fingerprint is SHA256:6u5vPdqrR9X2XtbR83HSI092UBrLPSN13Yx2o4SuaHc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.139.131' (ED25519) to the list of known hosts.
ubuntu@192.168.139.131's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-31-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.

519 updates can be applied immediately.
254 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
```

Screenshot successful execution SCP command:

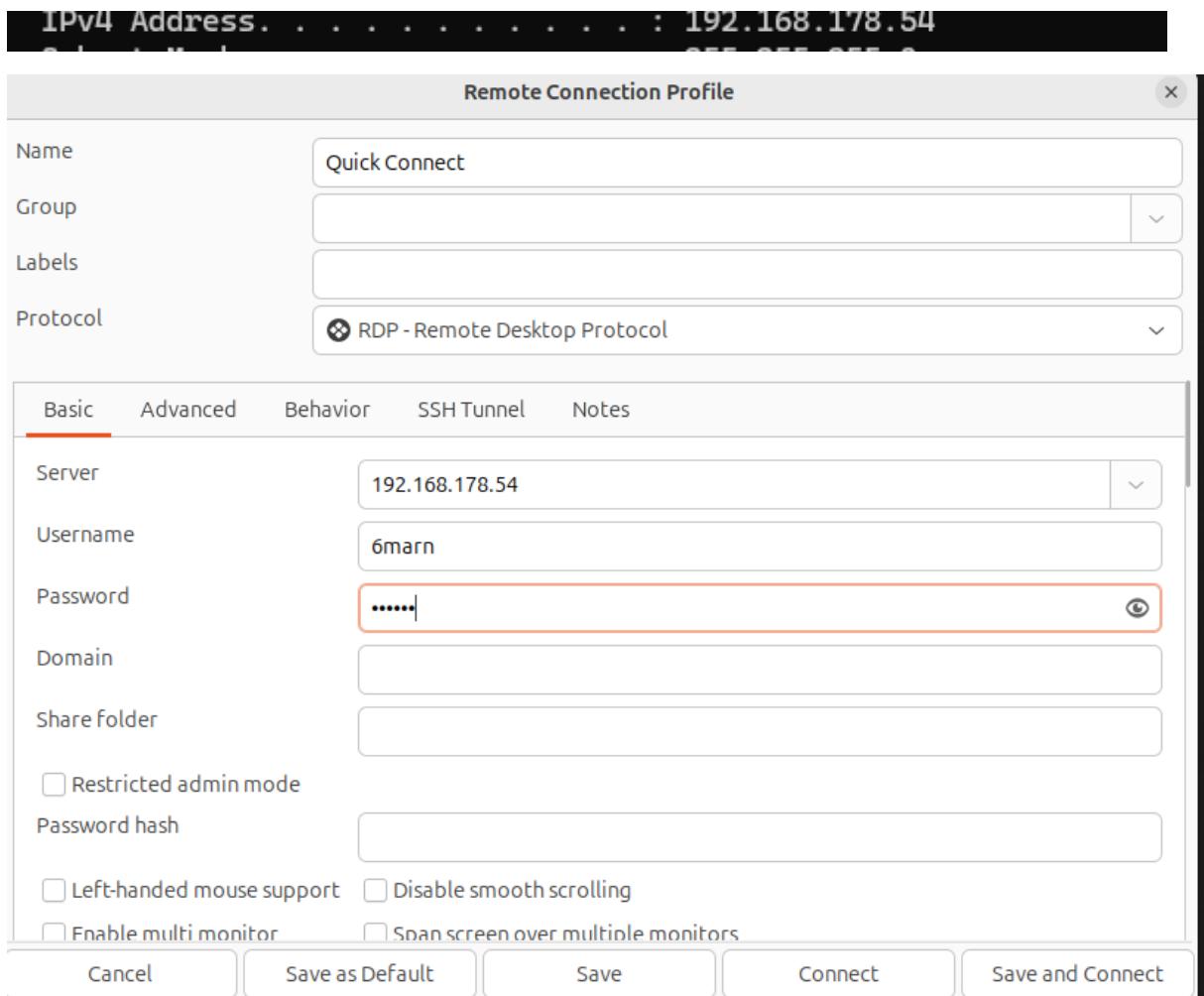
```
C:\Windows\System32\cmd.e X + - □ ×
Microsoft Windows [Version 10.0.26200.7462]
(c) Microsoft Corporation. All rights reserved.

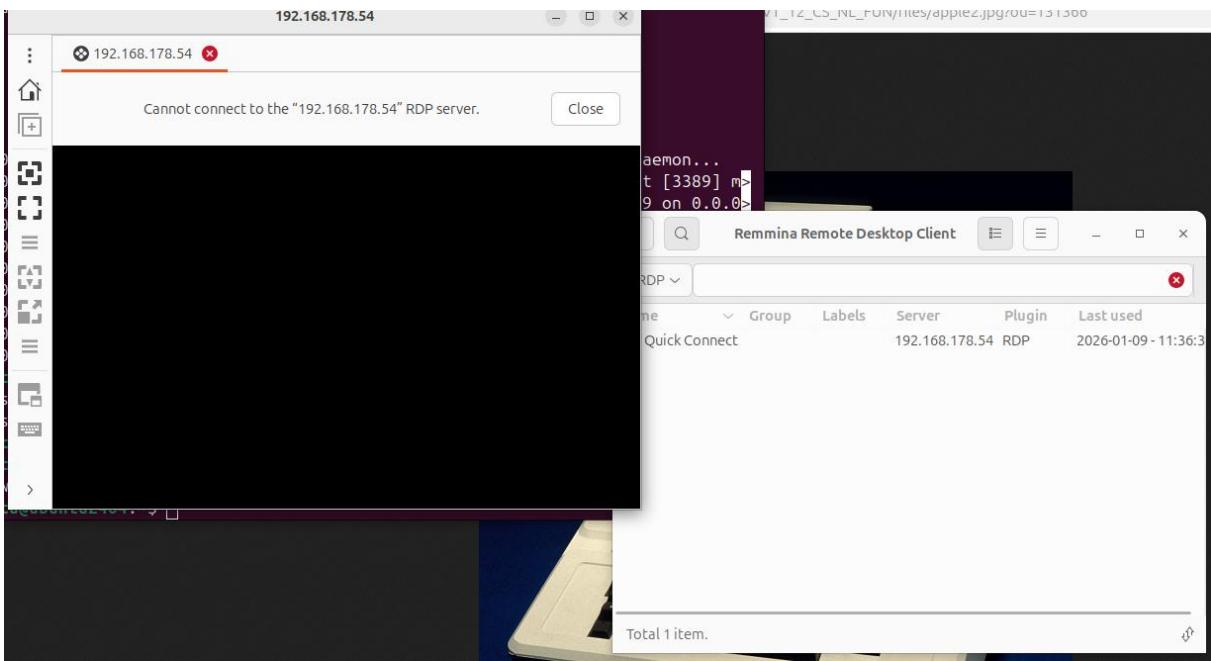
C:\Users\6marn\Documents>scp C:\Users\6marn\Documents\marnick.txt ubuntu@192.168.139.131:/home/ubuntu/
ssh: connect to host 0.0.0.131 port 22: Unknown error
scp: Connection closed

C:\Users\6marn\Documents>scp C:\Users\6marn\Documents\marnick.txt ubuntu@192.168.139.131:/home/ubuntu/
The authenticity of host '192.168.139.131 (192.168.139.131)' can't be established.
ED25519 key fingerprint is SHA256:6u5vPdqrR9X2XtbR83HSI092UBrLPSN13Yx2o4SuaHc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '192.168.139.131' (ED25519) to the list of known hosts.
ubuntu@192.168.139.131's password:
marnick.txt                                         100%    7      3.4KB/s   00:00

C:\Users\6marn\Documents>
```

Screenshot remmina:





Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

```
C:\Users\6marn>nslookup
Default Server: fritz.box
Address: fd00::de15:c8ff:fe76:bc80

> > amazon.com
Unrecognized command: > amazon.com
> amazon.com
Server: fritz.box
Address: fd00::de15:c8ff:fe76:bc80

Non-authoritative answer:
Name: amazon.com
Addresses: 98.87.170.74
           98.82.161.185
           98.87.170.71

> google.com
Server: fritz.box
Address: fd00::de15:c8ff:fe76:bc80

Non-authoritative answer:
Name: google.com
Addresses: 2a00:1450:400e:802::200e
           142.250.179.142

> one.one.one.one
Server: fritz.box
```

```

> one.one.one.one
Server: fritz.box
Address: fd00::de15:c8ff:fe76:bc80

Non-authoritative answer:
Name: one.one.one.one
Addresses: 2606:4700:4700::1111
           2606:4700:4700::1001
           1.0.0.1
           1.1.1.1

> dns.google.com
Server: fritz.box
Address: fd00::de15:c8ff:fe76:bc80

Non-authoritative answer:
Name: dns.google.com
Addresses: 2001:4860:4860::8844
           2001:4860:4860::8888
           8.8.8.8
           8.8.4.4

> bol.com
Server: fritz.box
Address: fd00::de15:c8ff:fe76:bc80

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

```

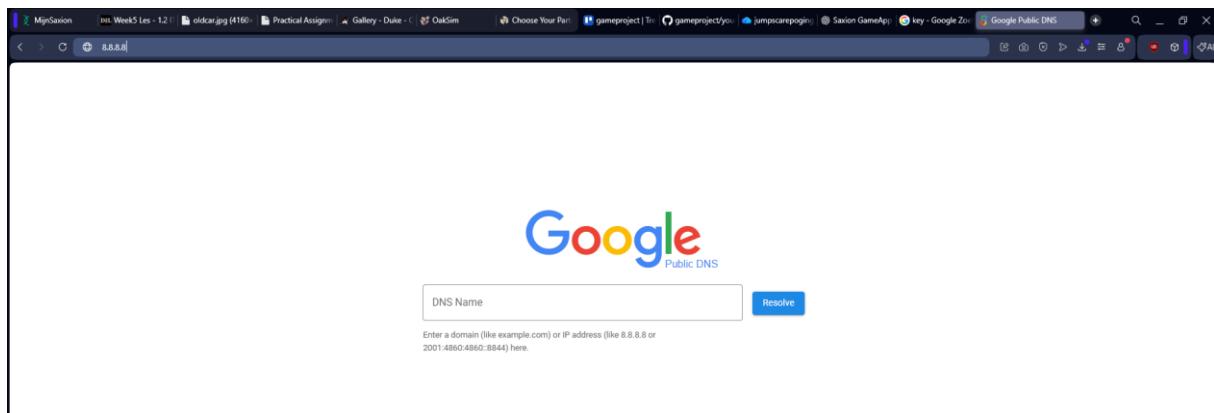
```

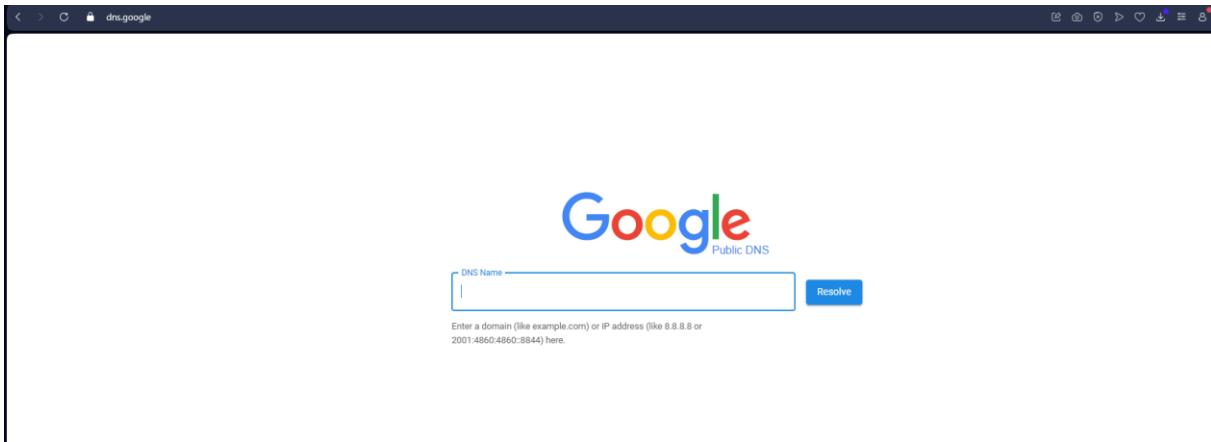
> w3schools.com
Server: fritz.box
Address: fd00::de15:c8ff:fe76:bc80

Non-authoritative answer:
Name: w3schools.com
Addresses: 13.248.240.135
           76.223.115.82
>

```

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

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

192.168.119.129 tot 192.168.119.254

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

```
ubuntu@ubuntu2404:~$ ipcalc 192.168.110.128/25
Address: 192.168.110.128      11000000.10101000.01101110.1 00000000
Netmask: 255.255.255.128 = 25 11111111.11111111.11111111.1 00000000
Wildcard: 0.0.0.127          00000000.00000000.00000000.0 11111111
=>
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 11111100
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.

/25 houdt in dat je 7 bits over hebt voor unieke IP adressen alleen de eerste en de laatste zijn gereserveerd

Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

```
ubuntu@ubuntu2404:~$ cd site
ubuntu@ubuntu2404:~/site$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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 pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:dc:89:90 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.139.131/24 brd 192.168.139.255 scope global dynamic noprefixroute ens33
        valid_lft 1021sec preferred_lft 1021sec
ubuntu@ubuntu2404:~/site$
```

Screenshot of Site directory contents:

```
ubuntu@ubuntu2404:~/site$ ls
css      images      pdf      week2.html  week4.html  week6.html
home.html  index.html  week1.html  week3.html  week5.html  week7.html
```

Screenshot python3 webserver command:

```
ubuntu@ubuntu2404:~/site$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Screenshot web browser visits your site

mijn hobby: klimmen

Ik vind boulderen leuk omdat je beweegt en het is leuk als je een route haalt.

wat foto's



Gemaakt door Marnick

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.

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
```

```

System.out.print("Please input a number: ");
int number = sc.nextInt();

System.out.println("Please select an option:");
System.out.println("1: Is this number even or odd?");
System.out.println("2: Is this number a power of 2?");
System.out.println("3: This number in two's complement");
System.out.println("4: Calculate the network address using bitwise AND (IP and subnet mask)");

int userChoice = sc.nextInt();

switch (userChoice) {
    case 1: {
        if ((number & 1) == 1) {
            System.out.println("Number is odd");
        } else {
            System.out.println("Number is even");
        }
        break;
    }
    case 2: {
        if ((number & (number - 1)) == 0) {
            System.out.println("Number is a power of 2");
        } else {
            System.out.println("Number isn't a power of 2");
        }
        break;
    }
    case 3: {
        int twosComplement = (~number) + 1;
        String binary = Integer.toBinaryString(twosComplement);
        System.out.println("Two's complement (binary): " + binary);
        break;
    }
    case 4: {
        sc.nextLine(); // Consume the leftover newline character
        System.out.print("Enter IP Address (e.g., 192.168.1.100): ");
        String ipAddress = sc.nextLine();

        System.out.print("Enter Subnet Mask (e.g., 255.255.255.224): ");
        String subnetMask = sc.nextLine();

        String ipBinary = toBinaryString(ipAddress);
        String subnetBinary = toBinaryString(subnetMask);

        String networkAddressBinary = calculateNetworkAddress(ipBinary, subnetBinary);
        String networkAddress = toDecimalIp(networkAddressBinary);
    }
}

```

```

        System.out.println("\nNetwork Address: " + networkAddress);

        calculateRange(networkAddress);
        break;
    }
    default:
        System.out.println("Invalid option!");
        break;
    }
}

public static String toBinaryString(String ip) {
    StringBuilder binary = new StringBuilder();
    String[] parts = ip.split("\\.");

    for (String part : parts) {
        int num = Integer.parseInt(part);
        String bin = String.format("%8s", Integer.toBinaryString(num)).replace(' ', '0');
        binary.append(bin);
    }
    return binary.toString();
}

public static String calculateNetworkAddress(String ipBinary, String subnetBinary) {
    StringBuilder networkAddress = new StringBuilder();

    for (int i = 0; i < ipBinary.length(); i++) {
        char ipBit = ipBinary.charAt(i);
        char subnetBit = subnetBinary.charAt(i);
        if (ipBit == '1' && subnetBit == '1') {
            networkAddress.append('1');
        } else {
            networkAddress.append('0');
        }
    }
    return networkAddress.toString();
}

public static String toDecimalIp(String binary) {
    StringBuilder ip = new StringBuilder();

    for (int i = 0; i < 4; i++) {
        String octetBinary = binary.substring(i * 8, (i + 1) * 8);
        int octetDecimal = Integer.parseInt(octetBinary, 2);
        ip.append(octetDecimal);
        if (i < 3) ip.append('.');
    }
    return ip.toString();
}

```

```
}

public static void calculateRange(String networkAddress) {
    String[] octets = networkAddress.split("\\.");
    int lastOctet = Integer.parseInt(octets[3]);

    int firstIp = lastOctet;
    int lastIp = firstIp + 31;

    System.out.println("Network Address: " + networkAddress);
    System.out.println("First IP in Range: " + octets[0] + "." + octets[1] + "." + octets[2] + "." + firstIp);
    System.out.println("Last IP in Range: " + octets[0] + "." + octets[1] + "." + octets[2] + "." + lastIp);
}
}
```

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