

Template Week 6 – Networking

Student number: 589020

Assignment 6.1: Working from home

Screenshot installation openssh-server:

```
Lucas@lucas-VMware-Virtual-Platform:~$ sudo apt install openssh-server
[sudo] password for lucas:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openSSH-server is already the newest version (1:9.6p1-3ubuntu13.14).
openSSH-server set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 153 not upgraded.
```

Screenshot successful SSH command execution:

```
C:\Users\Lucas Bourgonje>ssh lucas@192.168.139.132
The authenticity of host '192.168.139.132 (192.168.139.132)' can't be established.
ED25519 key fingerprint is SHA256:fyLRL2FrXu1EUyj5kCTFp7HdEmXNmpytdYTorU+CFqk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.139.132' (ED25519) to the list of known hosts.
lucas@192.168.139.132's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-37-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.

154 updates can be applied immediately.
6 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\OpenSSH\scp.exe: failed to upload file windows.txt to /home/Downloads/
PS D:\Saxion\Semester 2\ItFundamentals\week6> scp windows.txt lucas@192.168.139.132:/home/lucas/Downloads/
lucas@192.168.139.132's password:                                                 100%   0     0.0KB/s  00:00
```

Screenshot remmina:

Name	Group	Labels	Server	Plugin	Last used
Quick Connect			192.168.0.143	RDP	2026-01-07 - 19:54:40

Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

```
lucas@lucas-VMware-Virtual-Platform:~/Downloads$ nslookup  
> amazon.com  
Server:      127.0.0.53  
Address:     127.0.0.53#53  
  
Non-authoritative answer:  
Name:   amazon.com  
Address: 98.87.170.71  
Name:   amazon.com  
Address: 98.82.161.185  
Name:   amazon.com  
Address: 98.87.170.74  
  
> google.com  
Server:      127.0.0.53  
Address:     127.0.0.53#53  
  
Non-authoritative answer:  
Name:   google.com  
Address: 142.251.39.142  
Name:   google.com  
Address: 2a00:1450:400e:802::200e  
> one.one.one.one  
Server:      127.0.0.53  
Address:     127.0.0.53#53  
  
Non-authoritative answer:  
Name:   one.one.one.one  
Address: 1.1.1.1  
Name:   one.one.one.one  
Address: 1.0.0.1  
Name:   one.one.one.one  
Address: 2606:4700:4700::1111  
Name:   one.one.one.one  
Address: 2606:4700:4700::1001  
  
> dns.google.com  
Server:      127.0.0.53  
Address:     127.0.0.53#53  
  
Non-authoritative answer:  
Name:   dns.google.com  
Address: 8.8.8.8  
Name:   dns.google.com  
Address: 8.8.4.4  
Name:   dns.google.com  
Address: 2001:4860:4860::8844  
Name:   dns.google.com  
Address: 2001:4860:4860::8888
```

```
> bol.com
Server:      127.0.0.53
Address:     127.0.0.53#53

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

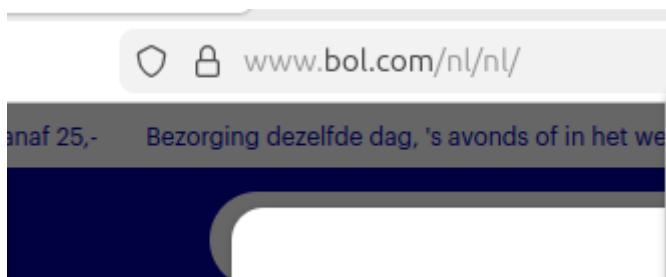
```
> w3schools.com
Server:      127.0.0.53
Address:     127.0.0.53#53

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:



When clicking enter it goes to bol.com



Assignment 6.3: subnetting

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

$$32 \text{ bits} - 25 = 7$$

$$7^2 = 128$$

128 IP addresses.

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

192.168.110.129 – 192.168.110.254

$129 + 128 = 157$. But IP addresses only go till 255 and 255 is already occupied.

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

```
lucas@lucas-VMware-Virtual-Platform:~$ 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 1111111
=>
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 1111110
Broadcast: 192.168.110.255    11000000.10101000.01101110.1 1111111
Hosts/Net: 126                Class C, Private Internet
```

Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

```
lucas@lucas-VMware-Virtual-Platform:~/Downloads/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 fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:34:28:64 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
        inet 192.168.139.132/24 brd 192.168.139.255 scope global dynamic noprefixroute ens33
            valid_lft 1612sec preferred_lft 1612sec
        inet6 fe80::20c:29ff:fe34:2864/64 scope link
            valid_lft forever preferred_lft forever
```

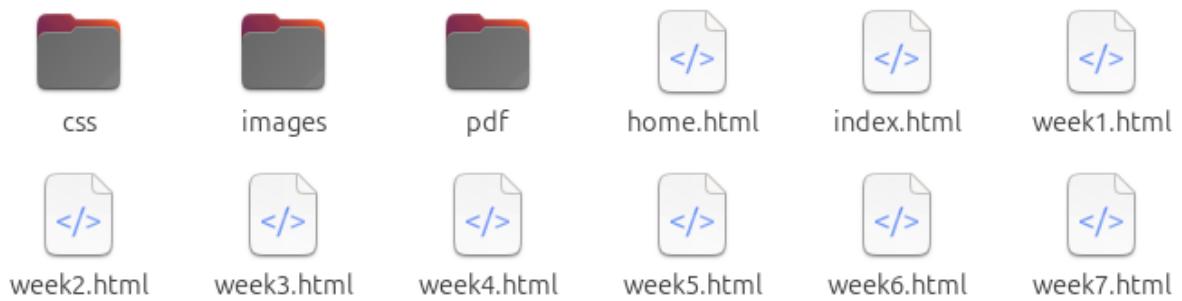
Ip: 192.168.139.132/24

Screenshot of Site directory contents:

```
lucas@lucas-VMware-Virtual-Platform:~/Downloads/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:

```
lucas@lucas-VMware-Virtual-Platform:~/Downloads/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

On windows:

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 nl.saxion.app.SaxionApp;

import java.awt.*;

public class Application implements Runnable {

    public static void main(String[] args) {
        SaxionApp.start(new Application(), 800, 600);
    }

    public void run() {
        SaxionApp.printLine("Calculate network segment.");
        SaxionApp.printLine("Enter IP address");
        String ipAddress = SaxionApp.readString();
        SaxionApp.printLine("Enter subnet mask");
        String subnetMask = SaxionApp.readString();

        calculateNetworkSegment(ipAddress, subnetMask);
    }

    public static void calculateNetworkSegment(String ipAddress, String subnetMask) {
        String[] ipParts = ipAddress.split("\\.");
        String[] subnetParts = subnetMask.split("\\.");

        int[] ip = new int[4];
        int[] subnet = new int[4];
        int[] network = new int[4];

        // Convert strings to integers
        for (int i = 0; i < 4; i++) {
            ip[i] = Integer.parseInt(ipParts[i]);
            subnet[i] = Integer.parseInt(subnetParts[i]);
        }

        SaxionApp.print("IP Address:  ");
        for (int i = 0; i < 4; i++) {
            SaxionApp.print(binary(ip[i]) + (i < 3 ? "." : "\n"));
        }

        SaxionApp.print("Subnet Mask:  ");
        for (int i = 0; i < 4; i++) {
            SaxionApp.print(binary(subnet[i]) + (i < 3 ? "." : "\n"));
        }

        SaxionApp.printLine("-----");
    }
}
```

```

SaxionApp.print("Network Addr: ");
for (int i = 0; i < 4; i++) {
    network[i] = ip[i] & subnet[i];
    SaxionApp.print(binary(network[i]) + (i < 3 ? "." : "\n"));
}

SaxionApp.printLine();
SaxionApp.printLine(
    "This gives " +
    network[0] + "." + network[1] + "." +
    network[2] + "." + network[3] +
    " in decimal as the network address."
);
SaxionApp.printLine();

int blockSize = 256 - subnet[3];
int start = network[3];
int end = start + blockSize - 1;

SaxionApp.printLine(
    "For a /27 subnet, each segment (or subnet) has 32 IP addresses ( $2^5$ ). " +
    "The range of this network segment is from " +
    network[0] + "." + network[1] + "." + network[2] + "." + start +
    " to " +
    network[0] + "." + network[1] + "." + network[2] + "." + end
);
}

private static String binary(int value) {
    return String.format("%8s", Integer.toBinaryString(value))
        .replace(' ', '0');
}
}

```

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