

Template Week 6 – Networking

Student number: 579864

Assignment 6.1: Working from home

Screenshot installation openssh-server:

```
sofia@sofiaVMware:~$ sudo apt install openssh-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass []
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 72 not upgraded.
Need to get 822 kB of archives.
After this operation, 6,771 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports/noble-updates/main arm64 openssh-sftp-
server arm64 1:9.6p1-3ubuntu13.14 [36.8 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports/noble-updates/main arm64 openssh-serv-
er arm64 1:9.6p1-3ubuntu13.14 [501 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports/noble/main arm64 ncurses-term all 6.4
+20240113-1ubuntu2 [275 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports/noble-updates/main arm64 ssh-import-i
d all 5.11-0ubuntu2.24.04.1 [10.1 kB]
Fetched 822 kB in 1s (691 kB/s)
Preconfiguring packages ...
Selecting previously unselected package openssh-sftp-server.
(Reading database ... 210231 files and directories currently installed.)
Preparing to unpack .../openssh-sftp-server_1%3a9.6p1-3ubuntu13.14_arm64.deb ...
Unpacking openssh-sftp-server (1:9.6p1-3ubuntu13.14) ...
Selecting previously unselected package openssh-server.
Preparing to unpack .../openssh-server_1%3a9.6p1-3ubuntu13.14_arm64.deb ...
Unpacking openssh-server (1:9.6p1-3ubuntu13.14) ...
Selecting previously unselected package ncurses-term.
```

Screenshot successful SSH command execution:

```
sofia@sofiaVMware:~$ ssh sofia@192.168.176.128
The authenticity of host '192.168.176.128' (192.168.176.128) can't be established.
ED25519 key fingerprint is SHA256:QLSMhULhC2Yd010J+9HOPGencJkyr7GFZsDpQyj/o0g.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.176.128' (ED25519) to the list of known hosts.
sofia@192.168.176.128's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-37-generic aarch64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

31 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

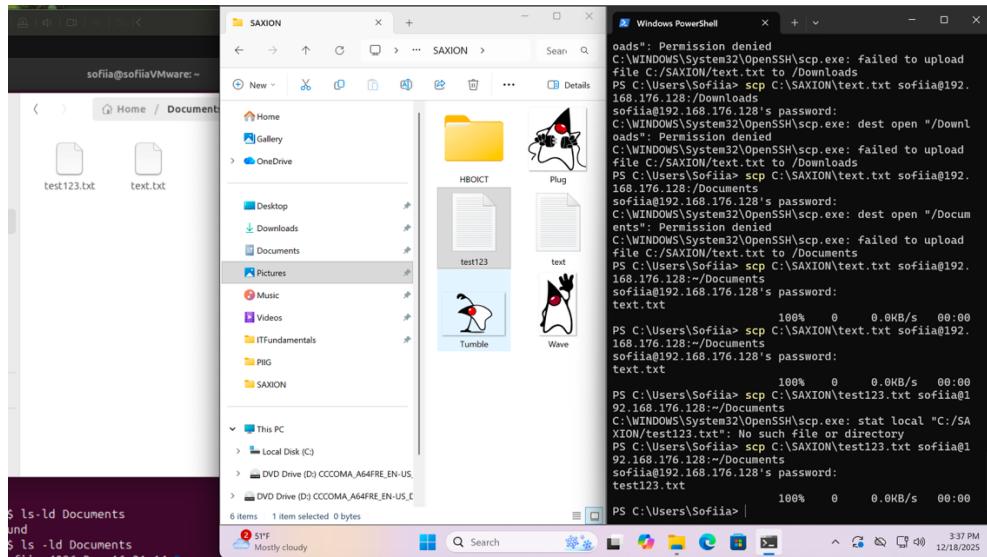
6 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

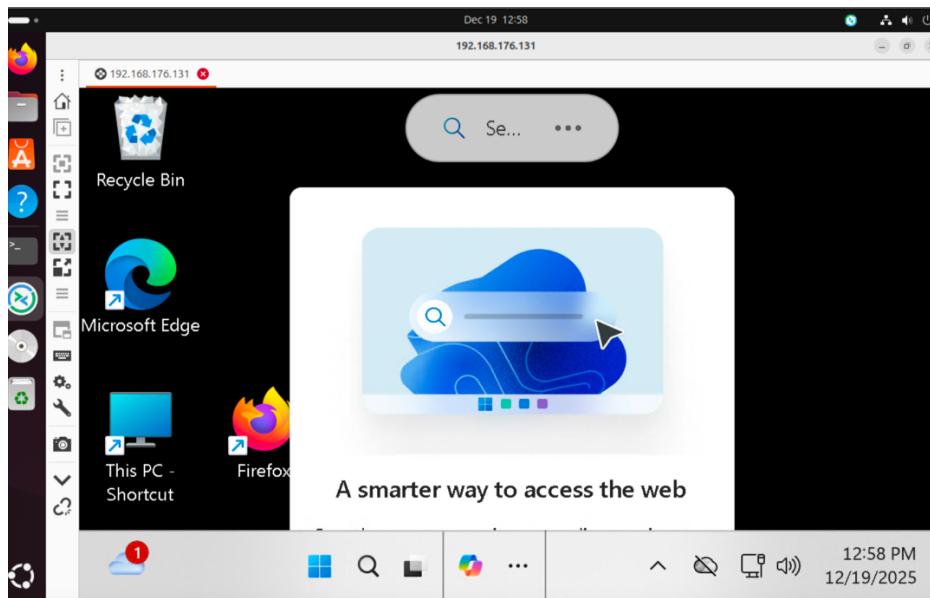
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

sofia@sofiaVMware:~$
```

Screenshot successful execution SCP command:



Screenshot remmina:



Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

```

[sofiaivanova@sofias-air ~ % nslookup
[> amazon.com
Server:      145.2.14.10
Address:     145.2.14.10#53

Non-authoritative answer:
Name:  amazon.com
Address: 98.87.170.74
Name:  amazon.com
Address: 98.87.170.71
Name:  amazon.com
Address: 98.82.161.185
[> google.com
Server:      145.2.14.10
Address:     145.2.14.10#53

Non-authoritative answer:
Name:  google.com
Address: 142.250.179.142
[> one.one.one.one
Server:      145.2.14.10
Address:     145.2.14.10#53

Non-authoritative answer:
Name:  one.one.one.one
Address: 1.0.0.1
Name:  one.one.one.one
Address: 1.1.1.1
[> dns.google.com
Server:      145.2.14.10
Address:     145.2.14.10#53

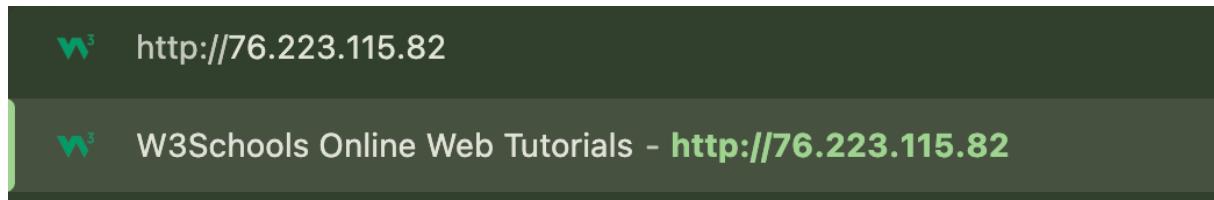
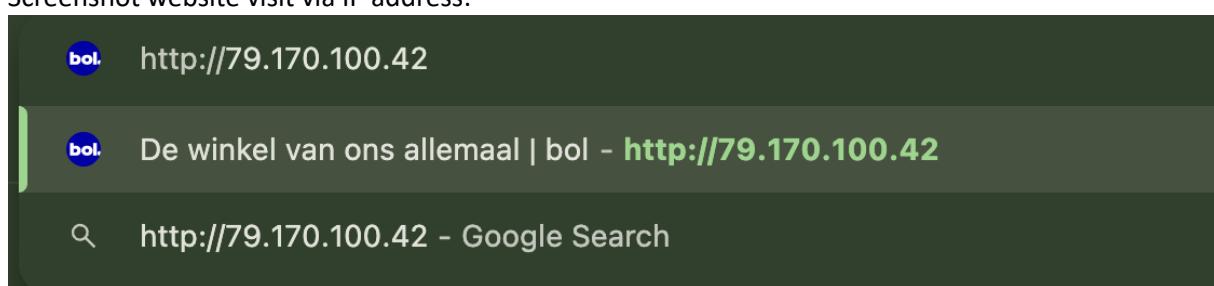
Non-authoritative answer:
Name:  dns.google.com
Address: 8.8.4.4
Name:  dns.google.com
Address: 8.8.8.8
[> bol.com
Server:      145.2.14.10
Address:     145.2.14.10#53

Non-authoritative answer:
Name:  bol.com
Address: 79.170.100.42
[> w3schools.com
Server:      145.2.14.10
Address:     145.2.14.10#53

Non-authoritative answer:
Name:  w3schools.com
Address: 13.248.240.135
Name:  w3schools.com
Address: 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?

126

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

```
sofiia@sofiiaVMware:~$ 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 11111110
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.

In case of /25 network configuration we have to do:

- 1) $32-25=7$
- 2) $2^7=128$ so a total of 128 IP addresses are available in this configuration

But only 126 of them can be handed out to users because two IP addresses are reserved by the network and the broadcast.

Assignment 6.4: HTML

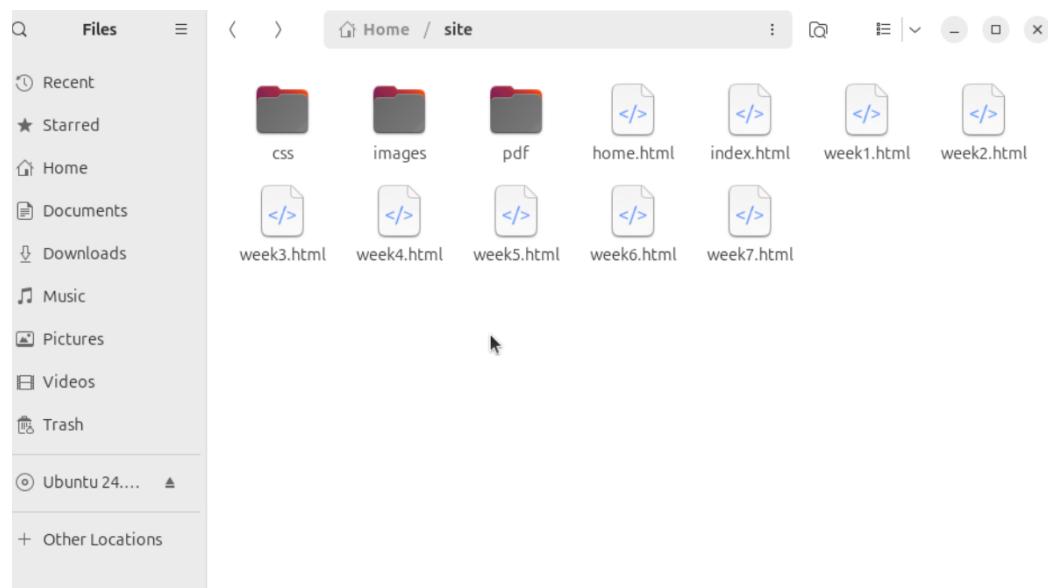
Screenshot IP address Ubuntu VM:

```

sofiia@sofiiaVMware:~/site$ cd ~/site
sofiia@sofiiaVMware:~/site$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    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: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    link/ether 00:0c:29:65:54:46 brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    inet 192.168.176.128/24 brd 192.168.176.255 scope global dynamic noprefixroute
        valid_lft 1089sec preferred_lft 1089sec
    inet6 fe80::20c:29ff:fe65:5446/64 scope link
        valid_lft forever preferred_lft forever
sofiia@sofiiaVMware:~/site$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...

```

Screenshot of Site directory contents:



Screenshot python3 webserver command:

```

sofiia@sofiiaVMware:~/site$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.176.1 - - [19/Dec/2025 14:00:00] "GET / HTTP/1.1" 200 -
192.168.176.1 - - [19/Dec/2025 14:00:00] "GET /home.html HTTP/1.1" 200 -
192.168.176.1 - - [19/Dec/2025 14:00:00] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
192.168.176.1 - - [19/Dec/2025 14:00:00] "GET /images/art.jpg HTTP/1.1" 200 -
192.168.176.1 - - [19/Dec/2025 14:00:00] code 404, message File not found
192.168.176.1 - - [19/Dec/2025 14:00:00] "GET /favicon.ico HTTP/1.1" 404 -

```

Screenshot web browser visits your site



Hobby: Art



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.util.ArrayList;

public class Application implements Runnable {
    public static void main(String[] args) {
        SaxionApp.start(new Application(), 800, 768);
    }

    public void run() {
        boolean continueProgram = true;
        while (continueProgram) {
            SaxionApp.printLine("1. Is number odd?");
            SaxionApp.printLine("2. Is number a power of 2?");
            SaxionApp.printLine("3. Two's complement of number?");
            SaxionApp.printLine("4. Calculate network segment");
            SaxionApp.print("Please select an option: ");
            int input = SaxionApp.readInt();
            if (input == 1) {
                isNumberOdd();
            } else if (input == 2) {
                isNumberAPowerOfTwo();
            } else if (input == 3) {
                isNumberNegative();
            } else if (input == 4) {
                calculateNetworkSegment();
            } else if (input == 0) {
                continueProgram = false;
            }
            SaxionApp.pause();
            SaxionApp.clear();
        }
    }

    public int askForInput() {
        SaxionApp.printLine("Enter number: ");
        return SaxionApp.readInt();
    }

    public void isNumberNegative() {
        int userInput = askForInput();
        userInput = ~userInput + 1;
        String binary = Integer.toBinaryString(userInput);
        SaxionApp.printLine("Number: " + binary);
    }

    public void isNumberOdd() {
        int userInput = askForInput();
        if ((userInput & 1) == 1) SaxionApp.printLine("number is odd");
        else SaxionApp.printLine("number is even");
    }

    public void isNumberAPowerOfTwo() {
        int userInput = askForInput();
        if ((userInput & userInput - 1) == 0) SaxionApp.printLine("number
is a power of 2");
        else SaxionApp.printLine("number isn't a power of 2");
    }
}
```

```

public void calculateNetworkSegment() {
    SaxionApp.printLine("Enter IP address: ");
    String ipAddr = SaxionApp.readString();
    SaxionApp.printLine("Enter subnet: ");
    String subnet = SaxionApp.readString();

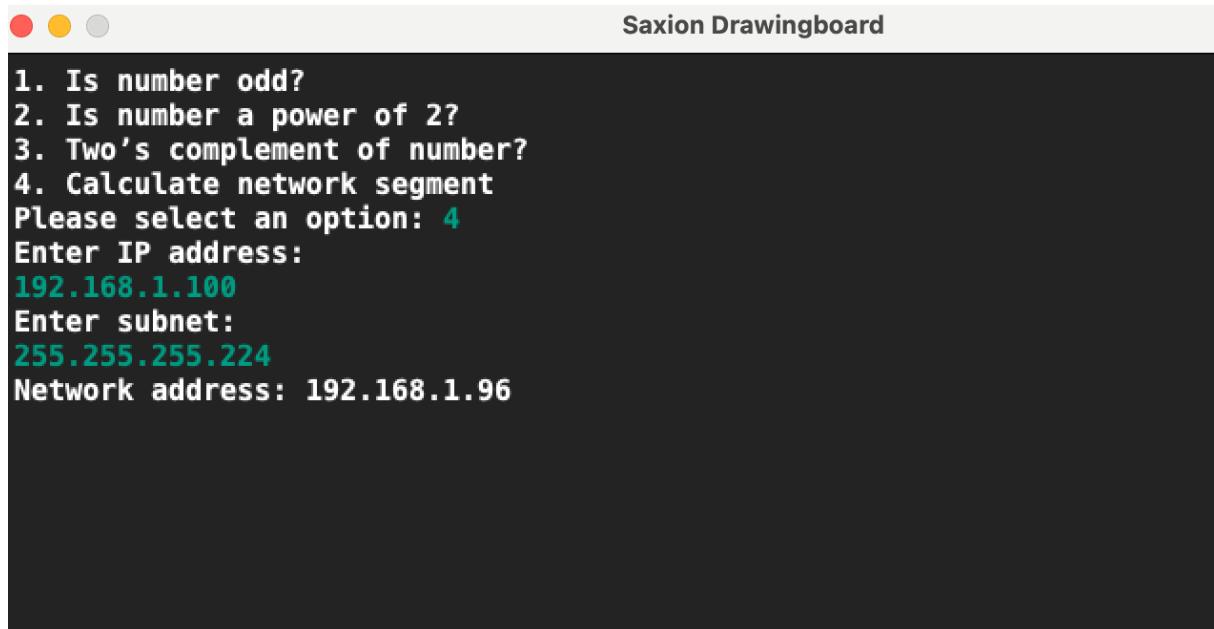
    String[] ipArray = ipAddr.split("\\.");
    int[] ipIntegers = new int[ipArray.length];
    String[] subnetArray = subnet.split("\\.");
    int[] subnetIntegers = new int[subnetArray.length];

    for (int i = 0; i < ipArray.length; i++) {
        ipIntegers[i] = Integer.parseInt(ipArray[i].trim());
    }
    for (int i = 0; i < subnetArray.length; i++) {
        subnetIntegers[i] = Integer.parseInt(subnetArray[i].trim());
    }

    int[] network = new int[4];

    for (int i = 0; i < 4; i++) {
        network[i] = ipIntegers[i] & subnetIntegers[i];
    }
    SaxionApp.print("Network address: ");
    for (int i = 0; i < 4; i++) {
        SaxionApp.print(network[i]);
        if (i < 3) {
            SaxionApp.print(".");
        }
    }
    SaxionApp.printLine("");
}
}

```



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