

Template Week 6 – Networking

Student number: 589871

Assignment 6.1: Working from home

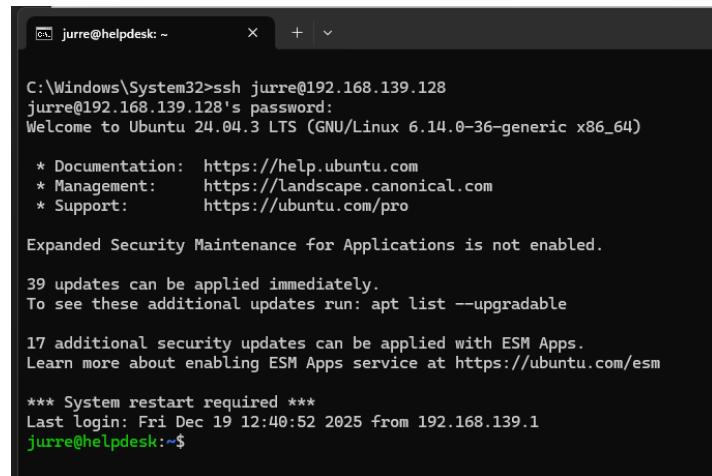
Screenshot installation openssh-server:

```
jurre@helpdesk:~/website$ sudo apt install openssh-server
[sudo] password for jurre:

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 69 not upgraded.
Need to get 832 kB of archives.
```

```
jurre@helpdesk:~/website$ sudo systemctl enable --now ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /usr/lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /usr/lib/systemd/system/ssh.service.
```

Screenshot successful SSH command execution:



```
C:\Windows\System32>ssh jurre@192.168.139.128
jurre@192.168.139.128's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-36-generic x86_64)

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

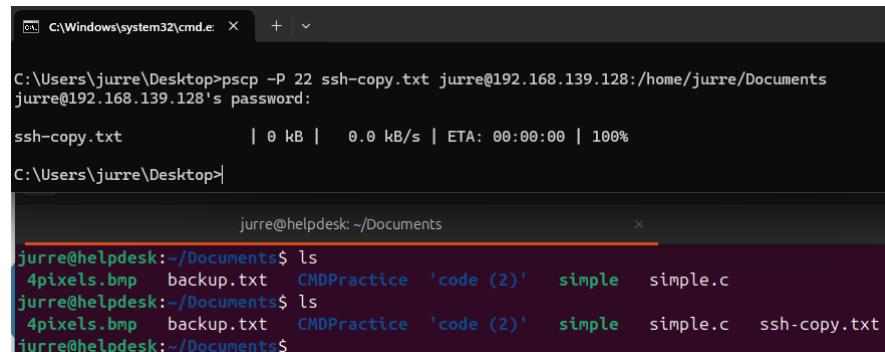
Expanded Security Maintenance for Applications is not enabled.

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

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

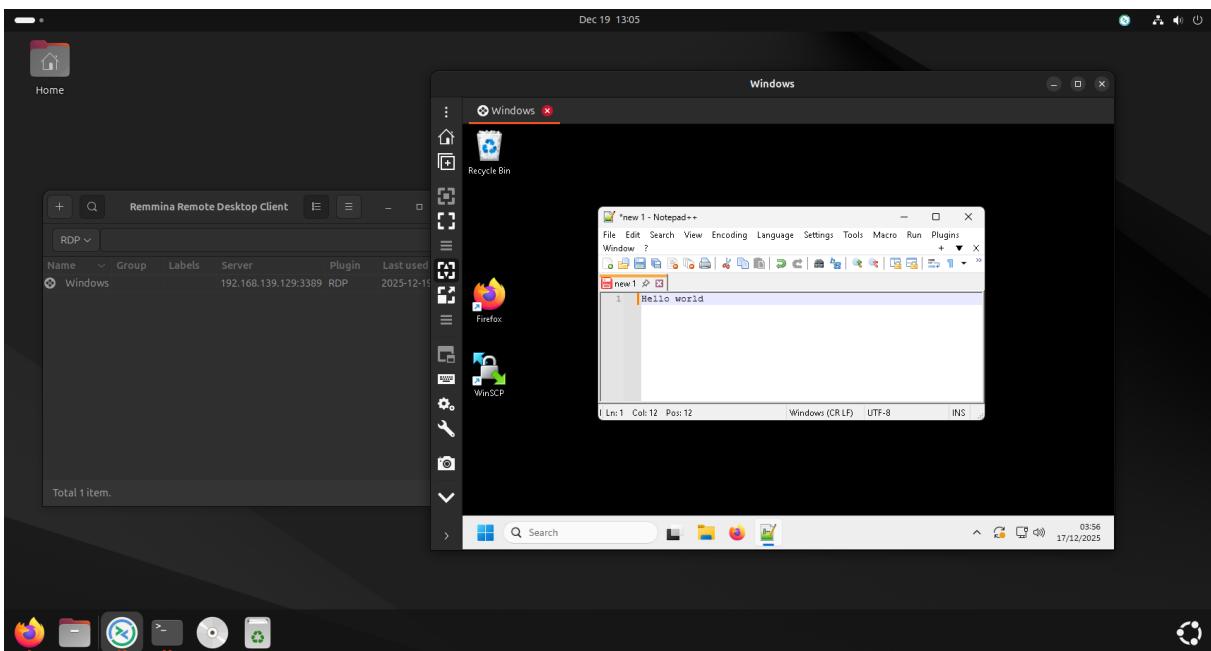
*** System restart required ***
Last login: Fri Dec 19 12:40:52 2025 from 192.168.139.1
jurre@helpdesk:~$
```

Screenshot successful execution SCP command:



```
C:\Windows\system32\cmd.e: ~ + ▾
C:\Users\jurre\Desktop>pscp -P 22 ssh-copy.txt jurre@192.168.139.128:/home/jurre/Documents
jurre@192.168.139.128's password:
ssh-copy.txt | 0 kB | 0.0 kB/s | ETA: 00:00:00 | 100%
C:\Users\jurre\Desktop>
jurre@helpdesk:~/Documents
jurre@helpdesk:~/Documents$ ls
4pixels.bmp  backup.txt  CMDPractice 'code (2)'  simple  simple.c
jurre@helpdesk:~/Documents$ ls
4pixels.bmp  backup.txt  CMDPractice 'code (2)'  simple  simple.c  ssh-copy.txt
jurre@helpdesk:~/Documents$
```

Screenshot remmina:



Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

```
C:\Windows\System32>nslookup
Default Server: d-hk-mer-ib02.infra.saxion.net
Address: 145.2.14.10

> amazon.com
Server: d-hk-mer-ib02.infra.saxion.net
Address: 145.2.14.10

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

> google.com
Server: d-hk-mer-ib02.infra.saxion.net
Address: 145.2.14.10

Non-authoritative answer:
Name: google.com
Addresses: 2a00:1450:400e:801::200e
          142.250.179.142

> one.one.one.one
Server: d-hk-mer-ib02.infra.saxion.net
Address: 145.2.14.10

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

> dns.google.com
Server: d-hk-mer-ib02.infra.saxion.net
Address: 145.2.14.10

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

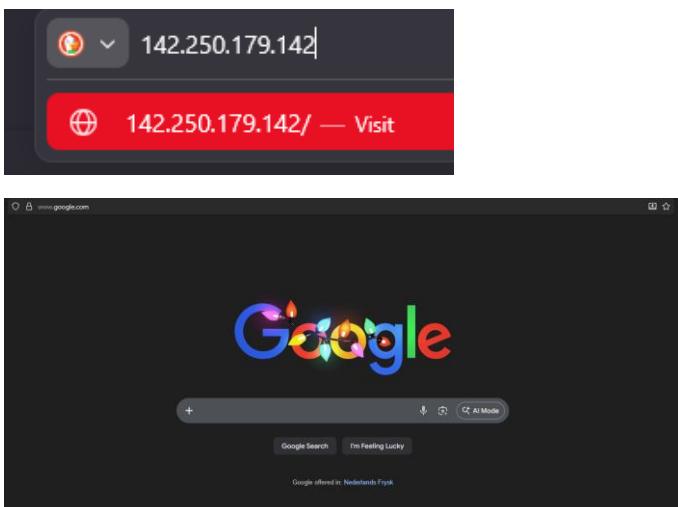
> bol.com
Server: d-hk-mer-ib02.infra.saxion.net
Address: 145.2.14.10

Non-authoritative answer:
Name: bol.com
Address: 79.170.100.42

> w3schools.com
Server: d-hk-mer-ib02.infra.saxion.net
Address: 145.2.14.10

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?

$$32-25 = 7$$

$2^7 = 128$ addresses

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

IP Address: 11000000.10101000.01101110.10000000

Subnet Mask: 11111111.11111111.11111111.10000000

Network Addr: 11000000.10101000.01101110.10000000

Start: 192.168.110.128

End: 192.168.110.255

Usable Start: 192.168.110.129

Usable End: 192.168.110.254

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

```
jurre@helpdesk:~$ 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
```

Explain the above calculation in your own words.

The range of static bits is defined by 25 with 32 total bits that gives us 7 dynamic bits. Using the equation 2^{BITS} (7 in this case) we can get the amount of values that can be stored within this bit range which is 128. Then using a 25 bit subnet mask and the given configuration we do a Bitwise AND to get the starting value, add one for the start of the usable space gives us 129. Starting value (128) + total amount of bits gives us the end value 255, subtract one for the end of the usable space.

Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

```
jurre@helpdesk:~/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:f1:22:76 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.139.128/24 brd 192.168.139.255 scope global dynamic noprefixroute ens33
        valid_lft 1374sec preferred_lft 1374sec
```

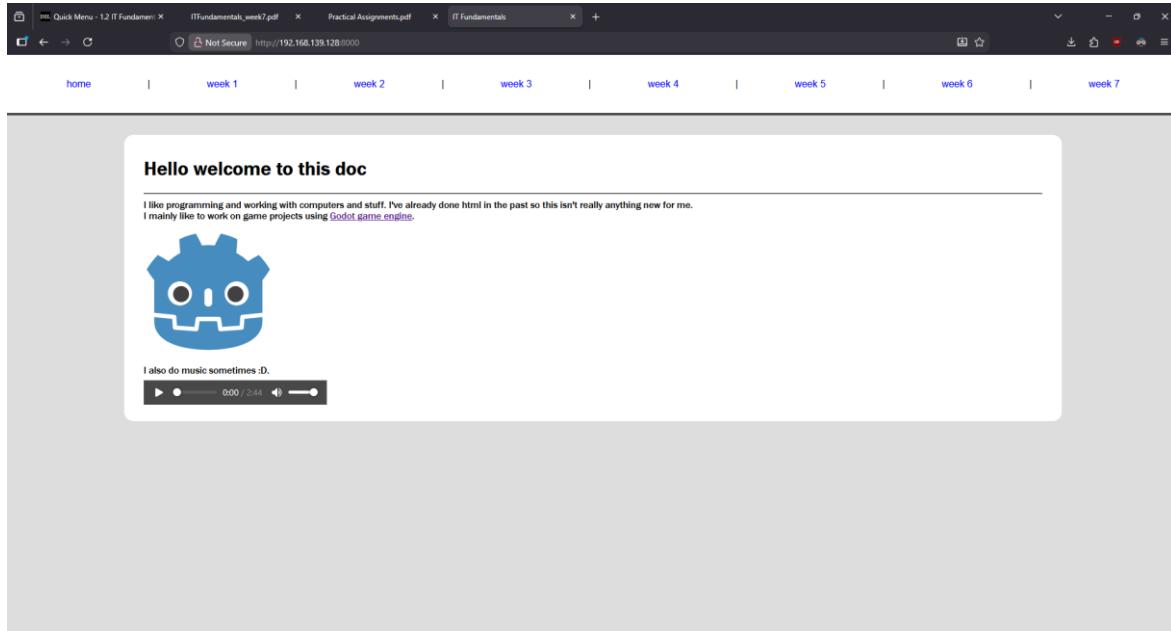
Screenshot of Site directory contents:

```
jurre@helpdesk:~/Downloads/site$ ls
content  home.html  pdf          week2.html  week4.html  week6.html
css       index.html  week1.html  week3.html  week5.html  week7.html
```

Screenshot python3 webserver command:

```
jurre@helpdesk:~/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



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.

```
1: Is number odd
2: Is number a power of 2
3: Two's complement
4: Network Segment
5: Exit
Please select command:
4
Ip and subnet:
196.168.1.100
255.255.255.224
196.168.1.96
```

PRESS ANY KEY TO CONTINUE

```
public void networkSegment(String ip, String mask) { 1 usage
    String[] ipNumbers = ip.split("\\.");
    String[] maskNumbers = mask.split("\\.");
    StringBuilder segment = new StringBuilder();

    for (int i = 0; i < ipNumbers.length; i++) {
        segment.append(Integer.parseInt(ipNumbers[i]) & Integer.parseInt(maskNumbers[i]));
        if (i != ipNumbers.length-1)
            segment.append(".");
    }

    SaxionApp.printLine(segment);
}
```