

# Template Week 6 – Networking

Student number: 564595

## Assignment 6.1: Working from home

Screenshot installation openssh-server:

```
reza-hekmatirad@hekmatirad:~$ sudo systemctl start ssh
reza-hekmatirad@hekmatirad:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
    Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
    Active: active (running) since Fri 2025-12-19 12:53:50 CET; 32s ago
      TriggeredBy: ● ssh.socket
        Docs: man:sshd(8)
               man:sshd_config(5)
      Process: 5009 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
        Main PID: 5011 (sshd)
           Tasks: 1 (limit: 4545)
          CPU: 21ms
         CGroup: /system.slice/ssh.service
                 └─5011 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Dec 19 12:53:50 hekmatirad systemd[1]: Starting ssh.service - OpenBSD Secure Shell...
Dec 19 12:53:50 hekmatirad sshd[5011]: Server listening on 0.0.0.0 port 22.
Dec 19 12:53:50 hekmatirad sshd[5011]: Server listening on :: port 22.
Dec 19 12:53:50 hekmatirad systemd[1]: Started ssh.service - OpenBSD Secure Shell...
lines 1-18/18 (END)
```

Screenshot successful SSH command execution:

```
C:\Users\Asus>ssh reza-hekmatirad@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:FK8CQJ6iE8K2qU3+gMWYnVMtSJR13ud2A2EYIGT6QfE.
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.
reza-hekmatirad@192.168.139.131'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.

151 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

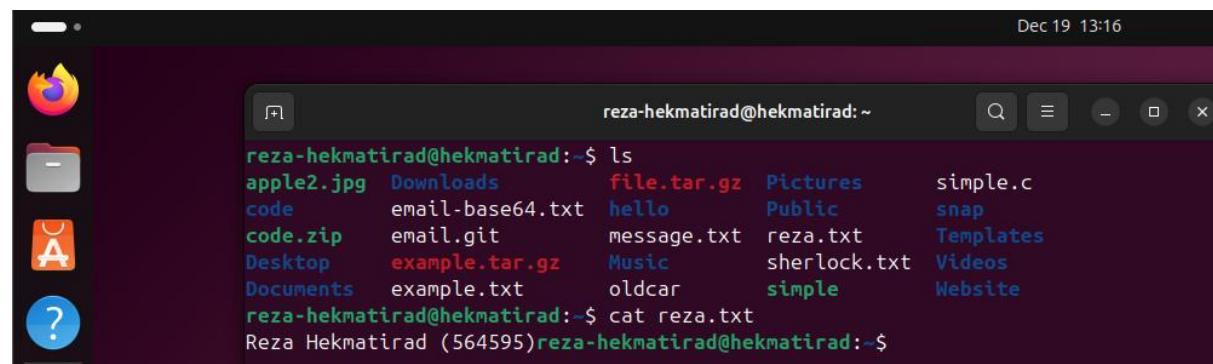
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

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

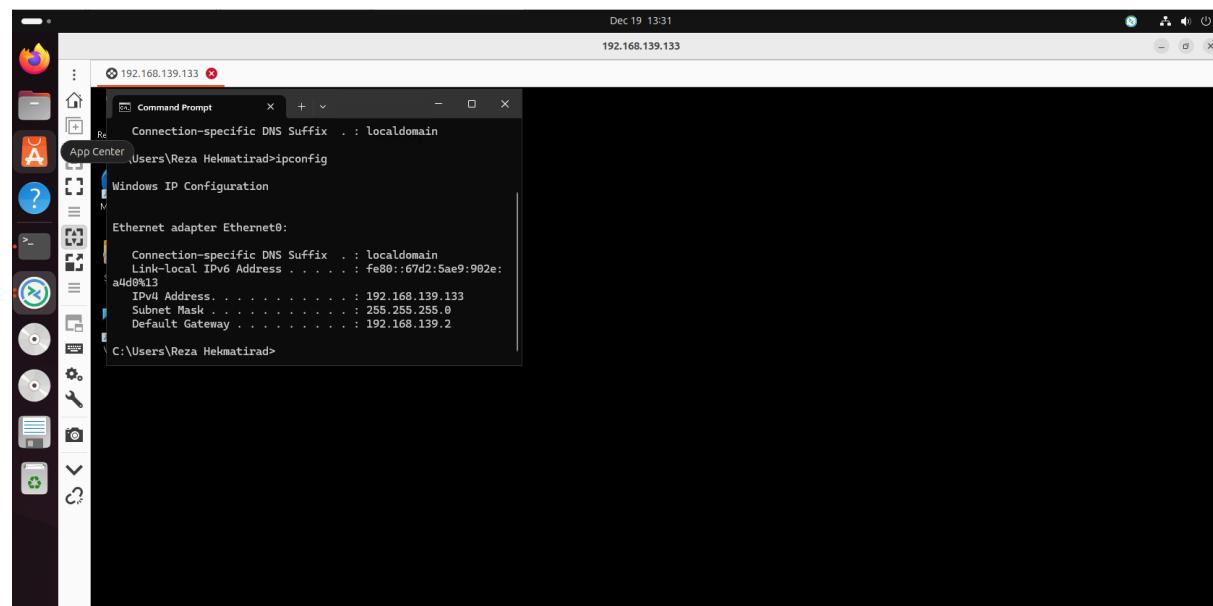
reza-hekmatirad@hekmatirad:~$ |
```

Screenshot successful execution SCP command:

```
C:\Users\Asus> scp C:\Users\Asus\OneDrive\Desktop\reza.txt reza-hekmatirad@192.168.139.131:~  
reza-hekmatirad@192.168.139.131's password:  
reza.txt  
      100%   24    11.7KB/s  00:00  
C:\Users\Asus>
```



Screenshot remmina:



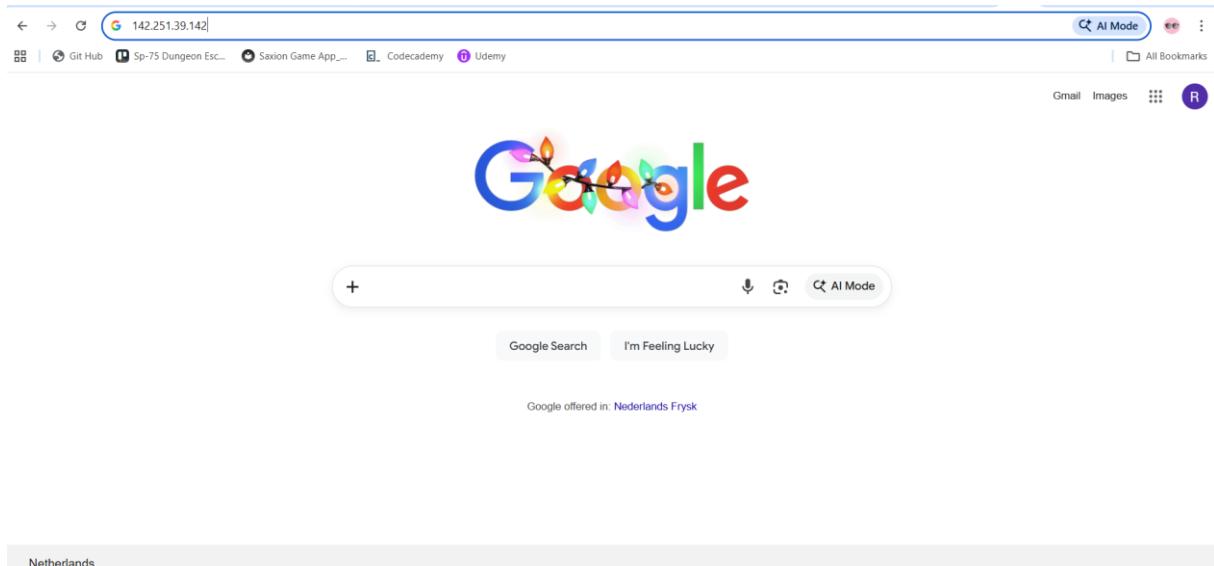
## Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

```
C:\WINDOWS\system32\cmd > 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: 98.87.170.74  
         98.82.161.185  
         98.87.170.71  
  
> 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:804::200e  
          142.251.39.142  
  
> 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
```

```
C:\WINDOWS\system32\cmd >  
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  
          8.8.4.4  
          8.8.8.8  
  
> 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: 76.223.115.82  
          13.248.240.135
```

Screenshot website visit via IP address:

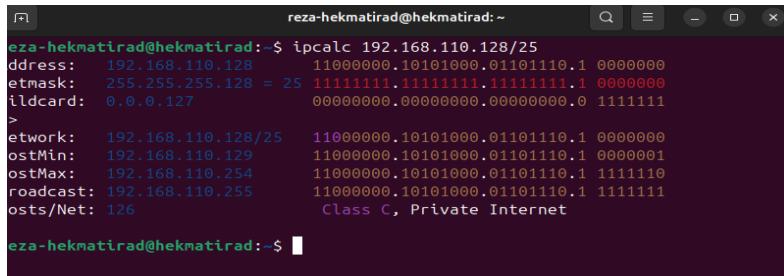


### Assignment 6.3: subnetting

How many IP addresses are in this network configuration 192.168.110.128/25?  $2^7 = 128$

What is the usable IP range to hand out to the connected computers?  $128 - 2 = 126$

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



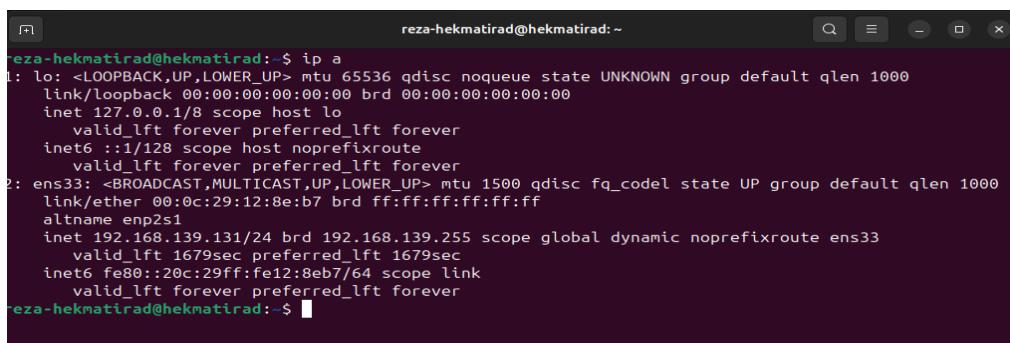
```
reza-hekmatirad@hekmatirad:~$ ipcalc 192.168.110.128/25
address: 192.168.110.128      11000000.10101000.01101110.1 00000000
netmask: 255.255.255.128 = 25 11111111.11111111.11111111.1 00000000
wildcard: 0.0.0.127          00000000.00000000.00000000.0 11111111
>
network: 192.168.110.128/25 11000000.10101000.01101110.1 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.

- It tells me all the important info about the network. The /25 means 25 bits are for the network and 7 bits for hosts, so the first IP, 192.168.110.128, is the network address and the last, 192.168.110.255, is the broadcast address. The usable IPs are the ones in between, from 192.168.110.129 to 192.168.110.254, which gives 126 addresses we can assign to devices. It also shows the subnet mask and wildcard mask, basically doing all the binary math for me so I can quickly see the network setup.

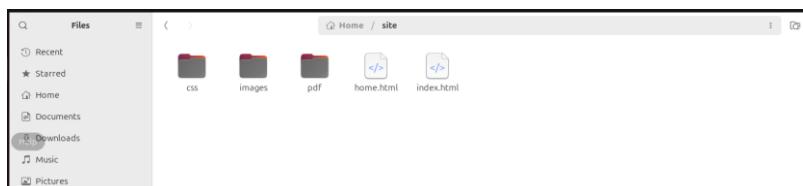
### Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:



```
reza-hekmatirad@hekmatirad:~$ 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:29:12:8e:b7 brd ff:ff:ff:ff:ff:ff
        altname enps1
        inet 192.168.139.131/24 brd 192.168.139.255 scope global dynamic noprefixroute ens33
            valid_lft 1679sec preferred_lft 1679sec
        inet6 fe80::20c:29ff:fe12:8eb7/64 scope link
            valid_lft forever preferred_lft forever
reza-hekmatirad@hekmatirad:~$
```

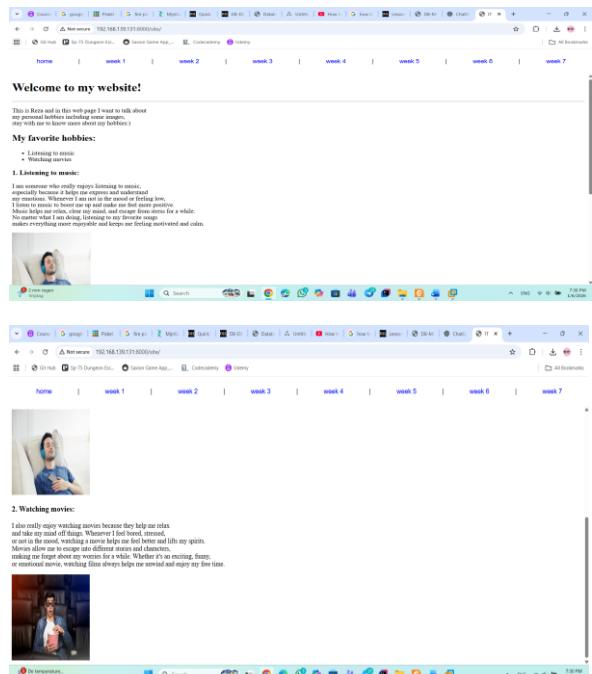
Screenshot of Site directory contents:



Screenshot python3 webserver command:

```
reza-hekmatirad@hekmatirad:~$ 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:



(192.168.139.131)

### 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.

source code:

```
import java.util.Scanner;

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

Scanner sc = new Scanner(System.in);
        System.out.println("Enter an IP address: (format: --> 192.168.1.100)");
        String ip = sc.nextLine();
        System.out.println("Enter a Subnet: (format: --> 255.255.255.224)");
        String subnet = sc.nextLine();

        String[] ipParts = ip.split("\\.");
        String[] subnetParts = subnet.split("\\.");

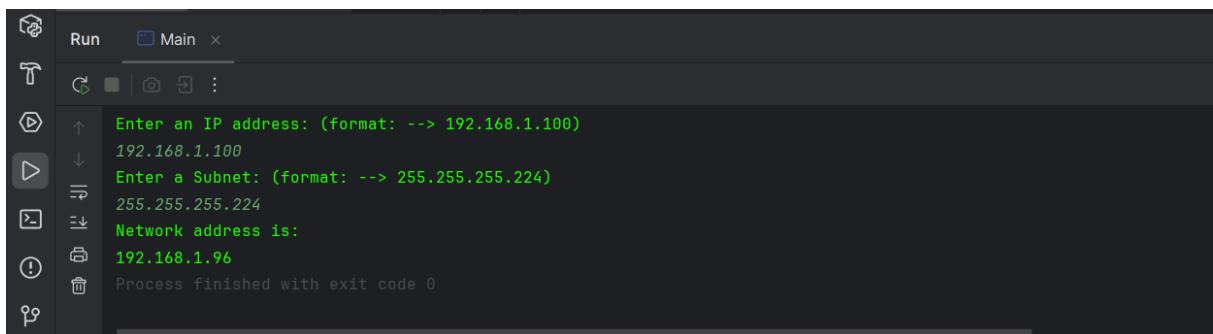
        int[] network = new int[4];

        for (int i = 0; i < 4; i++){
            int ipInt = Integer.parseInt(ipParts[i]);
            int subnetInt = Integer.parseInt(subnetParts[i]);
            network[i] = ipInt & subnetInt;
        }

        System.out.println("Network address is: ");

        for (int i = 0; i < 4; i++){
            System.out.print(network[i]);
            if (i < 3){
                System.out.print(".");
            }
        }
        sc.close();
    }
}
```

A working example:



```
Run Main x
G | : 
↑ Enter an IP address: (format: --> 192.168.1.100)
↓ 192.168.1.100
Enter a Subnet: (format: --> 255.255.255.224)
255.255.255.224
Network address is:
192.168.1.96
Process finished with exit code 0
```

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