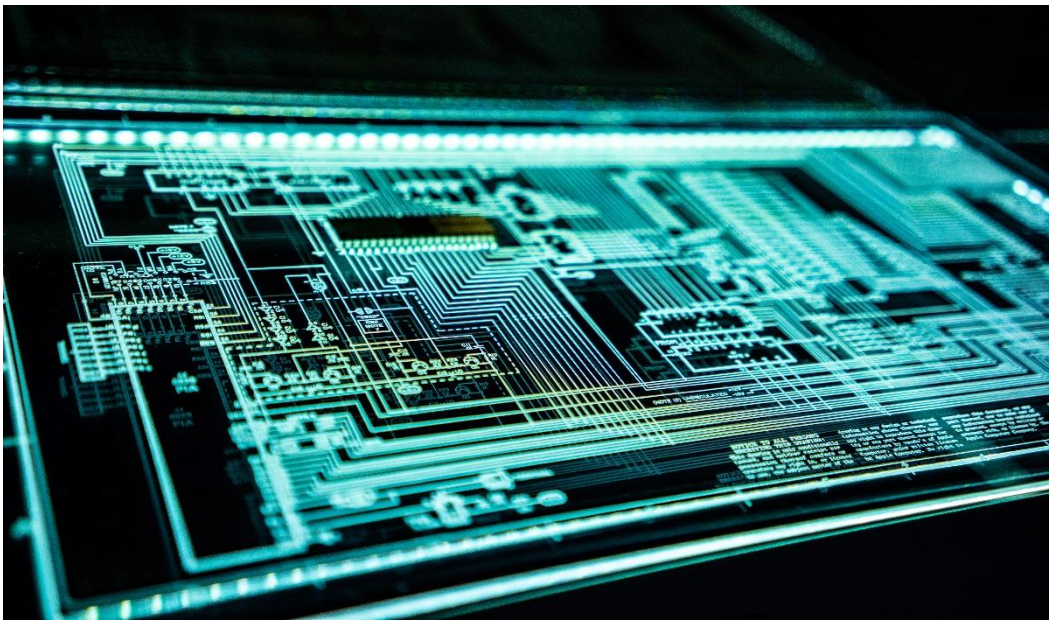
Université EUROMED de Fès
Ecole d'Ingénierie Digitale et d'Intelligence Artificielle Networking services and protocols

Final Project Enterprise Network



A PROJECT REPORT

BY

YOUSSEF IBOURK
HAMZA TAMZAITI
ASSIA JAROUI

Under the supervision
of
Prof. TADIST KHAWLA

Table of Content:

1. Introduction

1.1. Enterprise Networking
1.2. The Enterprise Description
1.3. About The Project

2. Network Architecture

3. Network Configuration

3.1. Overview
3.2. Dynamic Host Configuration Protocol (DHCP)
3.3. Domain Name System Protocol (DNS)
3.4. Hyper Text Transfer Protocol (HTTP)
3.5. Postfix Mail Server (MAIL)
3.6. File Transfer Protocol (FTP)
3.7. Lightweight Directory Access Protocol (LDAP)

4. Conclusion

1. Introduction

1.1. Enterprise Networking:

An enterprise network is the backbone for facilitating an organization's communications and connecting computers and devices throughout departments. An enterprise network environment is usually configured to facilitate access to data.

1.2. The enterprise description:

An enterprise desires to manage locally a number of services depending on the departments. It establishes a network that will allow it to provide services that:

- Deliver IP addresses to devices that connect to the network.
- Turn domain names into IP addresses (and vise-versa).
- Share files over a TCP/IP network.
- Allow the user to access web pages through a browser.
- Transfer email over the network.
- Manage directory services and authentication.

1.3. About The Project:

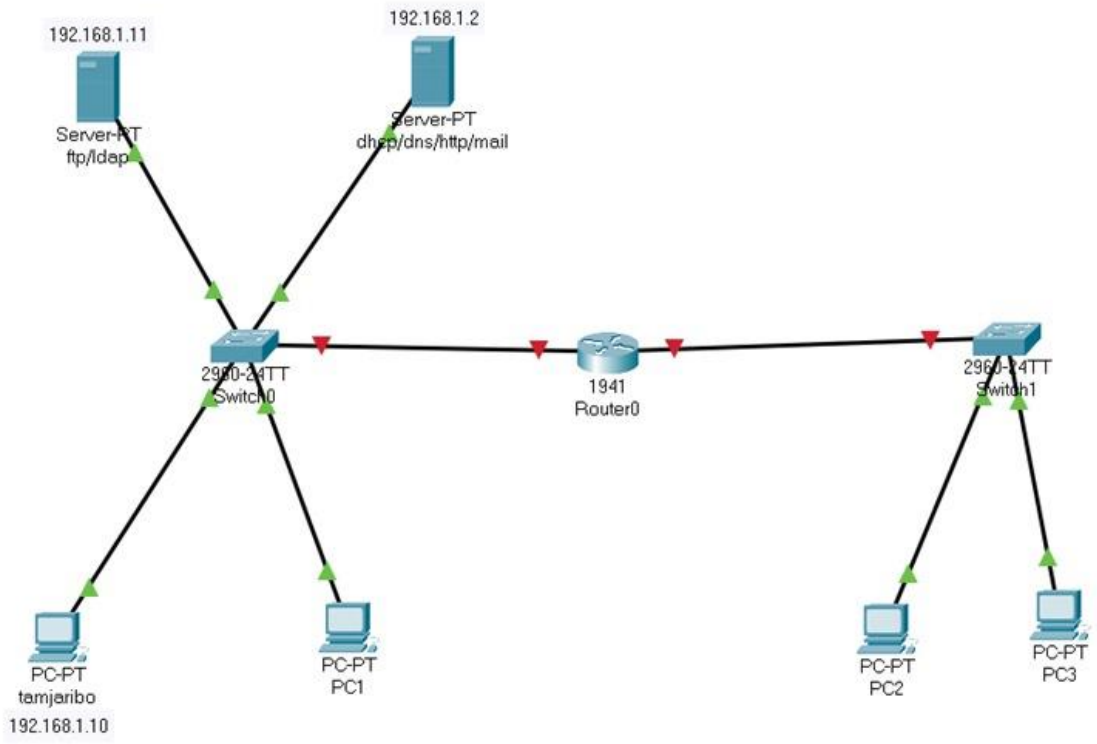
The task is to build a network divided into two subnets with:

- **One** DHCP server responsible for delivering addresses to devices connected on both subnets.
- Managing the domain names will be handled by the DNS server.

- The FTP server is used to share files in only one subnet (users on the other subnet are not allowed to access this service).
- An HTTP server managing two web pages. One is accessible to all hosts, and one with restricted access (user-password).
- A server managing the email exchange in the company.
- LDAP (Lightweight Directory Access Protocol) is used for directory services and authentication.

2. Network Architecture:

We use Cisco Packet Tracer to simulate our network architecture. Packet Tracer is a cross-platform visual simulation tool designed by Cisco Systems that allows users to create network topologies and imitate modern computer networks. The software allows users to simulate the configuration of Cisco routers and switches using a simulated command line interface.



3. Network Configuration:

A network is a collection of computers, servers, mainframes, network devices, peripherals, or other devices connected to allow data sharing. An example of a network is the Internet, which connects millions of people all over the world. To the right is an example image of a home network with multiple computers and other network devices all connected.

3.1. Overview

Before getting in the explanation of the project, we will put some basics that were needed in the creation of the project, we will start talking about the tcp/ip model, because the whole work was based on this model.

Let's start talking about upper-level protocols. Since we are parsing this level along the TCP / IP stack, we have it “one for three”.

In general, from the point of view of a networker, we do not care what happens inside the application layer. This is usually done by programmers. But it is important to know how data is formed and encapsulated in lower levels.

At work, for example, we have a rule: we ensure that the application starts up and passes it error-free over the network. If the problem lies in internal software failures, then we switch to developers, and this becomes their concern. But there are also problems that go along the thin line between us, and we solve them together.

So, application layer protocols provide interaction between a person and a network. There are a huge number of these protocols, and they perform completely different roles.

3.2. (DHCP) Dynamic Host Configuration Protocol:

Dynamic host configuration protocol. It allows nodes to dynamically obtain IP addresses and other parameters for correct network operation (default gateway, subnet mask, DNS server addresses). From myself I will say that this protocol saves the lives of many system administrators around the world. Agree that walking and manually prescribing IP parameters for each node is not the most pleasant experience. With DHCP, you can provide full control over IP addresses: create separate pools for each subnet, lease addresses, reserve addresses, and much more.

His work is very difficult for today's understanding. Too many packets, data, and frames must be transmitted before the requested address can be assigned to the computer.

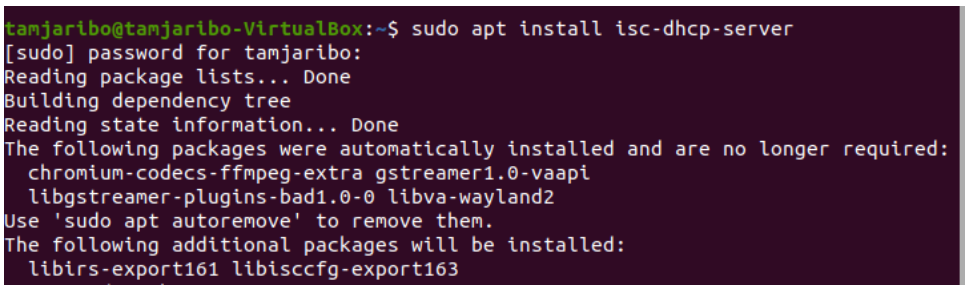
Let's configure the DHCP server so that clients can obtain IP addresses automatically.

Before downloading the DHCP server, we make an update:

```
sudo apt-get update
```

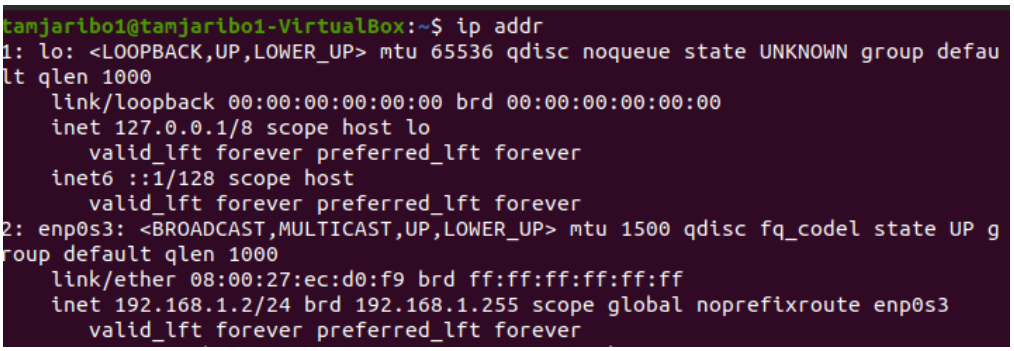
STEP 1: DHCP server installation

```
apt-get install isc-dhcp-server
```



Once the DHCP server installation is complete, we assign a static IP adresse to it.
The IP address of our DHCP server is 192.168.1.2. To find out the IP address of our DHCP server, we enter the following command in the Terminal:

```
ip addr
```



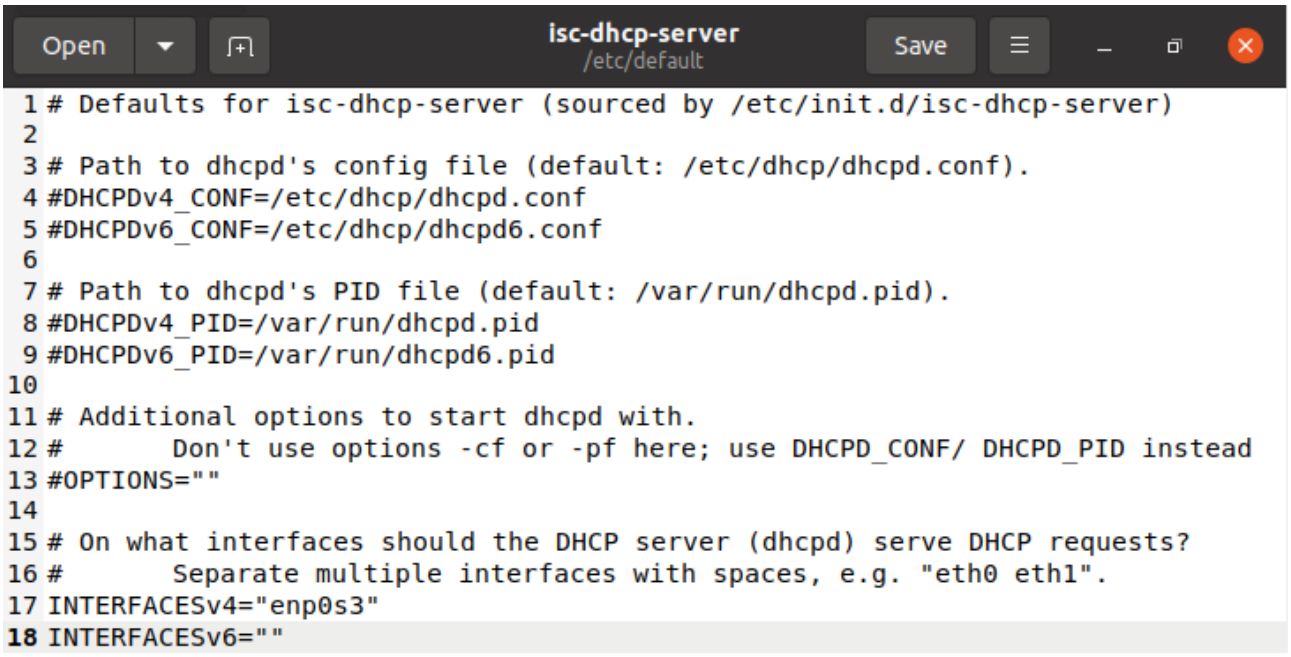
STEP 2: Configuration

We check the configue files first of our server

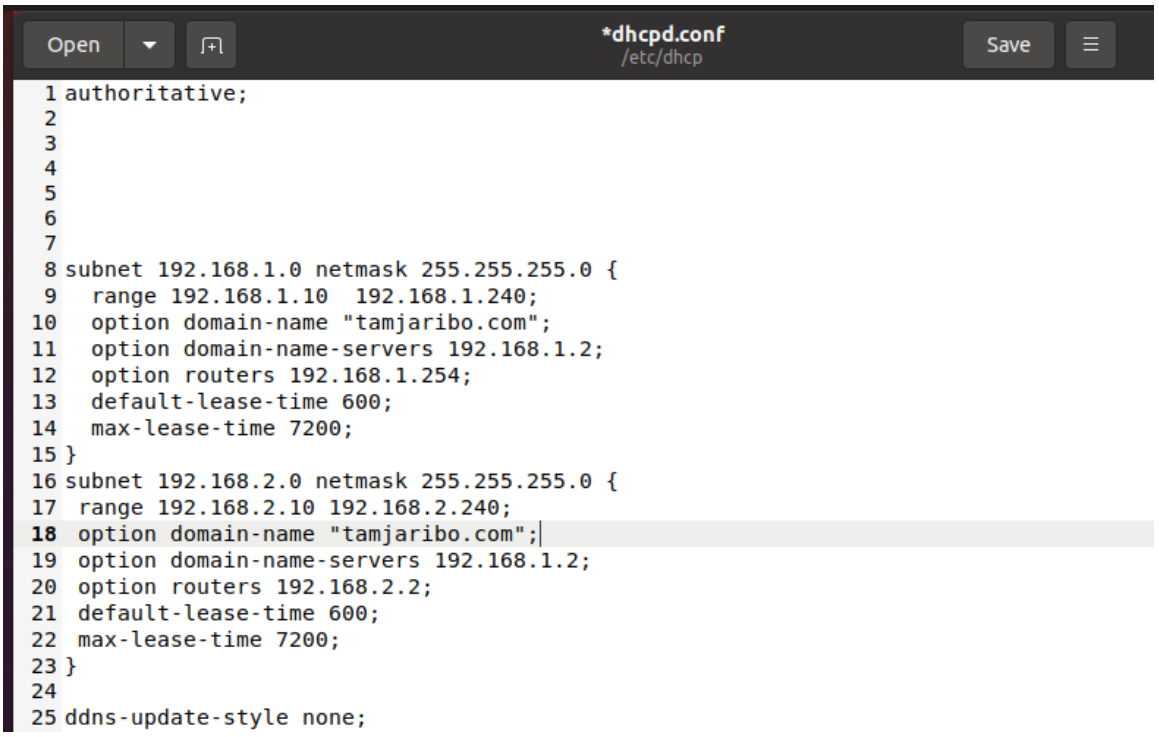
```
tamjaribo1@tamjaribo1-VirtualBox:~$ cd /etc/dhcp
tamjaribo1@tamjaribo1-VirtualBox:/etc/dhcp$ ls
ddns-keys  dhclient.conf          dhclient-exit-hooks.d  dhcpd.conf
debug      dhclient-enter-hooks.d  dhcpd6.conf
```

In the file ***/etc/default/isc-dhcp-server*** we edit the line:

```
INTERFACESv4="enp0s3"
```



We go to ***/etc/dhcp***, and open ***dhcpd.conf*** and set the settings:
We add the following lines to the configuration file to define the lease-time, domain and domain name servers, subnet, IP address range and default gateway as shown below:
And to make the DHCP server the official DHCP server for clients, we add **"authoritative"** line in the configuration file



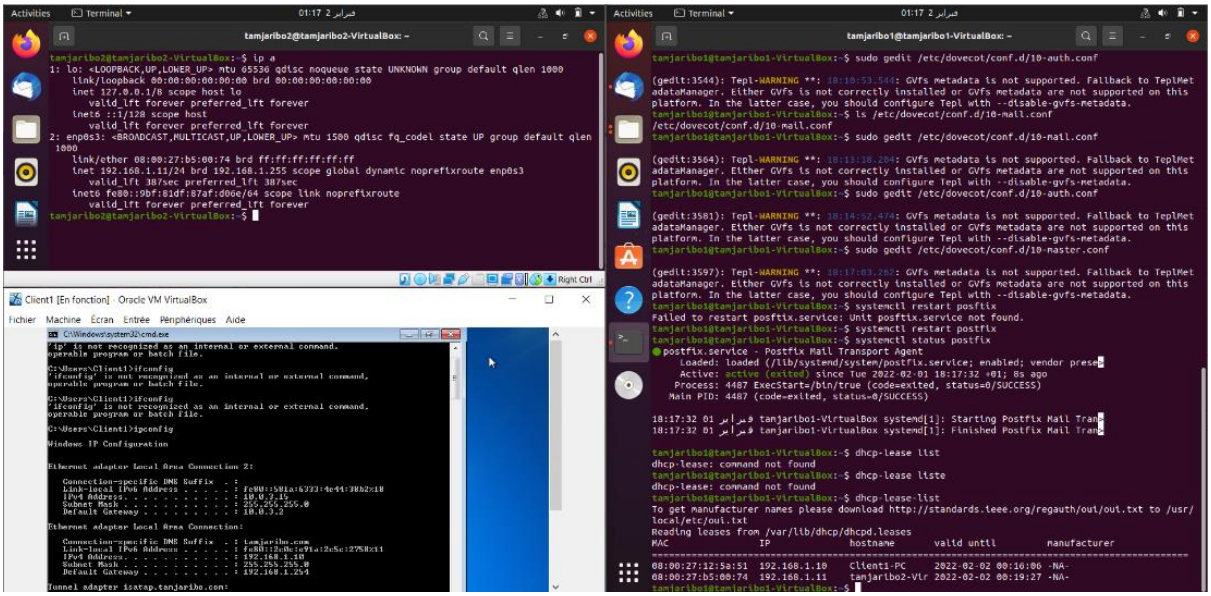
We restart now the DHCP server and check its status:

```
service isc-dhcp-server restart
Service isc-dhcp-server status
```

```
tamjaribo1@tamjaribo1-VirtualBox:/etc/dhcp$ sudo service isc-dhcp-server restar
t
tamjaribo1@tamjaribo1-VirtualBox:/etc/dhcp$ sudo service isc-dhcp-server status
● isc-dhcp-server.service - ISC DHCP IPv4 server
     Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; ven>
     Active: active (running) since Mon 2022-01-31 08:25:55 +01; 1s ago
       Docs: man:dhcpd(8)
   Main PID: 4477 (dhcpd)
      Tasks: 4 (limit: 2295)
     Memory: 4.4M
     CGroup: /system.slice/isc-dhcp-server.service
             └─4477 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/>

08:25:55 31 يناير  tamjaribo1-VirtualBox sh[4477]: PID file: /run/dhcp-server/d>
08:25:55 31 يناير  tamjaribo1-VirtualBox dhcpd[4477]: Wrote 0 leases to leases >
08:25:55 31 يناير  tamjaribo1-VirtualBox sh[4477]: Wrote 0 leases to leases fil>
08:25:55 31 يناير  tamjaribo1-VirtualBox dhcpd[4477]: Listening on LPF/enp0s3/0>
08:25:55 31 يناير  tamjaribo1-VirtualBox sh[4477]: Listening on LPF/enp0s3/08:0>
08:25:55 31 يناير  tamjaribo1-VirtualBox dhcpd[4477]: Sending on   LPF/enp0s3/0>
08:25:55 31 يناير  tamjaribo1-VirtualBox sh[4477]: Sending on   LPF/enp0s3/08:0>
08:25:55 31 يناير  tamjaribo1-VirtualBox dhcpd[4477]: Sending on   Socket/fallb>
08:25:55 31 يناير  tamjaribo1-VirtualBox sh[4477]: Sending on   Socket/fallback>
08:25:55 31 يناير  tamjaribo1-VirtualBox dhcpd[4477]: Server starting service.
lines 1-20/20 (END)
```

STEP 3: Test a client



Here we Tow machines hosts with IP addresses (192.168.1.10 and 192.168.1.11)

By Typing in our terminal, the command below:

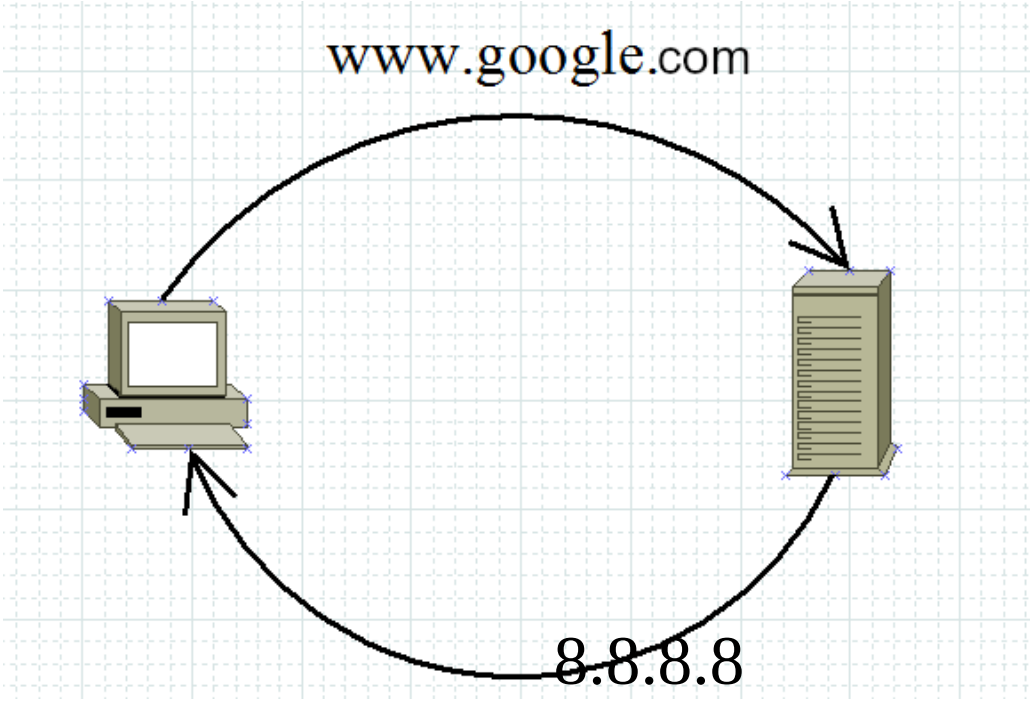
```
Dhcp-lease-list
```

We have seen our two clients in the lease-list

So far, everything is going well, and clients are now receiving addresses automatically.

3.3. Domain Name System Protocol (DNS)

Domain name system. Generally speaking, it stores information about domains. For example, what IP address corresponds to a certain name. Let me give you an example: when you open your favourite site, you refer to it by name. But in the Source Address and Destination Address fields, which work at the network layer, you cannot insert a name. The IP address must be there. This is exactly what DNS does. It tells you what IP address the requested name has. You, for example, apply to google.com.
Your computer has no idea who or what it is. He asks the DNS server **who is google.com?** And the server replies that google.com is 8.8.8.8 this is one of its addresses.
And after that, the computer sends a request to 8.8.8.8. For the user, everything will remain the same, as usual, I'll show it in the picture.



STEP 1: Installation

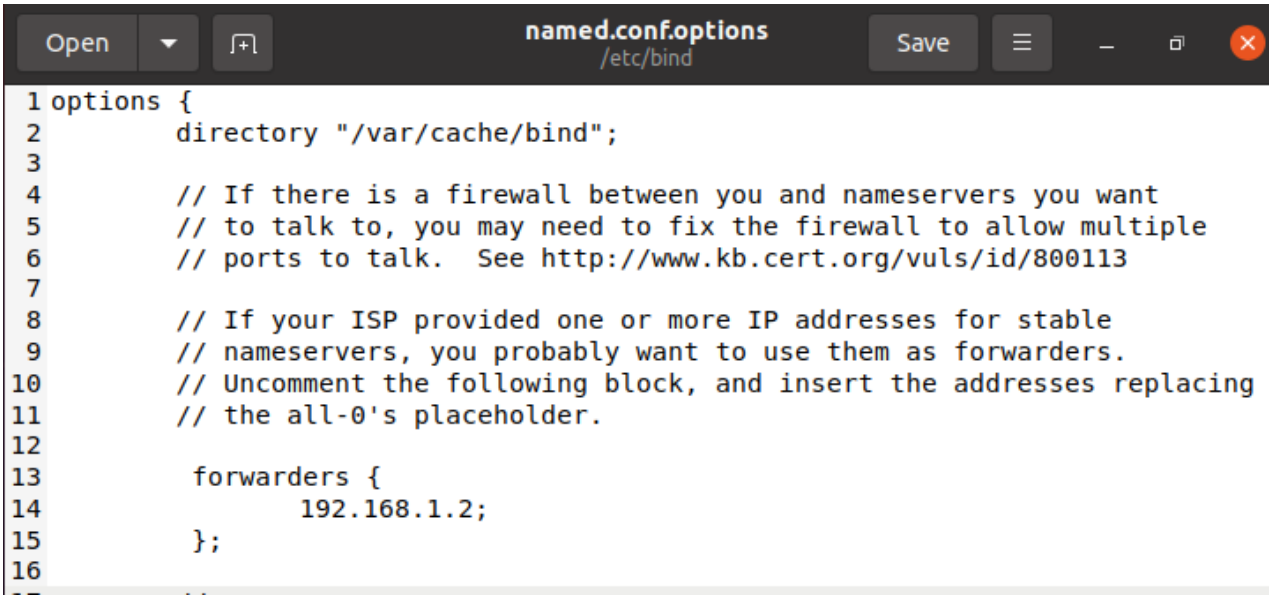
```
sudo apt-get install bind9 dnsutils
```

```
tamjaribo@tamjaribo-VirtualBox:~$ sudo apt install bind9 dnsutils -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  bind9-dnsutils bind9-libs bind9-utils python3-ply
Suggested packages:
  bind-doc resolvconf python-ply-doc
The following NEW packages will be installed:
  bind9 bind9-utils dnsutils python3-ply
The following packages will be upgraded:
  bind9-dnsutils bind9-libs
2 upgraded, 4 newly installed, 0 to remove and 191 not upgraded.
Need to get 454 kB/1,695 kB of archives.
After this operation, 1,956 kB of additional disk space will be used.
Get:1 http://ma.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-ply all 3.11-3ubuntu0.1 [46.3 kB]
Get:2 http://ma.archive.ubuntu.com/ubuntu focal-updates/main amd64 bind9-utils amd64 1:9.16.1-0ubuntu2.9 [172 kB]
Get:3 http://ma.archive.ubuntu.com/ubuntu focal-updates/main amd64 bind9 amd64 1:9.16.1-0ubuntu2.9 [233 kB]
Get:4 http://ma.archive.ubuntu.com/ubuntu focal-updates/universe amd64 dnsutils all 1:9.16.1-0ubuntu2.9 [2,756 B]
Fetched 454 kB in 2s (288 kB/s)
```

STEP 2: Configuration

We open and edit the /etc/bind/named.conf.options and configure the name server:

```
forwarders
{
192.168.1.2;
};
```



And edit the /etc/bind/named.conf.local file to add a DNS zone. We configure the forward zone first:

db.forwzone
/etc/bind

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     tamjaribo.com. root.tamjaribo.com. (
                              2                 ; Serial
                         604800                 ; Refresh
                          86400                 ; Retry
                        2419200                 ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.tamjaribo.com.
@       IN      A       192.168.1.2
@       IN      AAAA    ::1
ns      IN      A       192.168.1.2
```

Then the reverse zone:

db.revzone
/etc/bind

```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns.tamjaribo.com. root.tamjaribo.com. (
                              1                 ; Serial
                         604800                 ; Refresh
                          86400                 ; Retry
                        2419200                 ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.
2       IN      PTR     ns.tamjaribo.com.
```

Now we add the forward zone and reverse zone to **named.conf.local**

*named.conf.local
/etc/bind

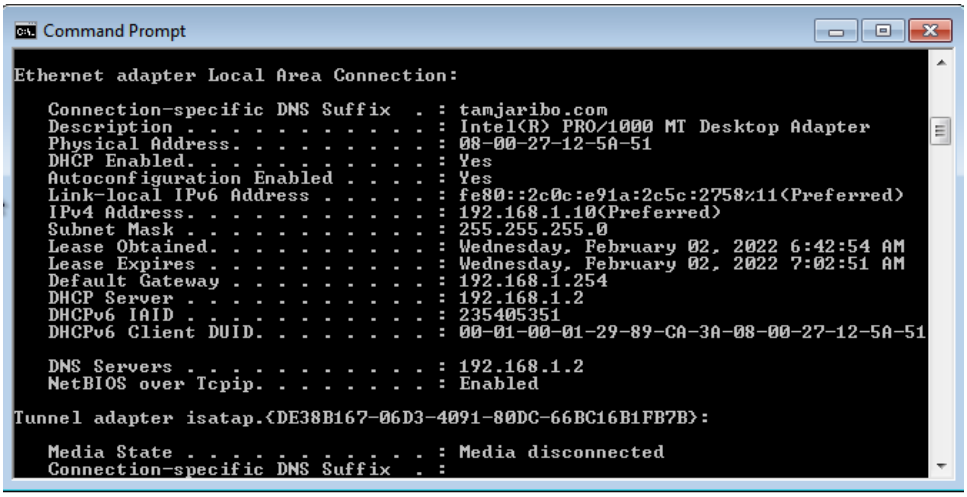
```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "tamjaribo.com"{
        type master;
        file "/etc/bind/db.forwzone";
};

zone "1.168.192.in-addr.arpa"{
        type matser;
        file "/etc/bind/db.revzone";
};
```

STEP 3: Test



Here our client is connected to our DNS server successfully.

3.4. Hyper Text Transfer Protocol (HTTP)

A data transfer protocol commonly used to retrieve information from websites. Every year this protocol becomes more and more popular, and there are more and more opportunities for its application. It uses a "client-server" model. That is, there are clients that form and send a request. And servers that listen to requests and, accordingly, respond to them.

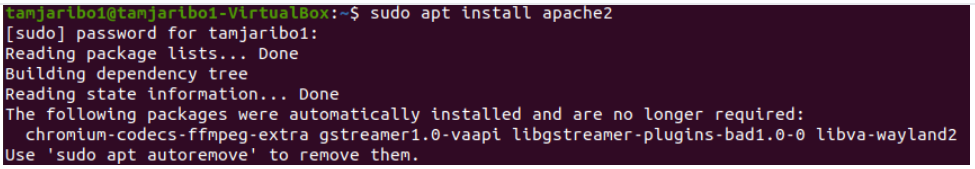
The clients are well-known web browsers: Internet Explorer, Mozilla Firefox, Google Chrome, etc. And as server software they use: Apache, IIS, nginx, etc.

We will work with Apache Server, because it is one of the most popular web servers in the world, it is an open-source and cross-platform HTTP server that powers a large percentage of the Internet’s websites. And provides many powerful features that can be extended through additional modules.

Now let's dig into the configuration of our HTTP server.

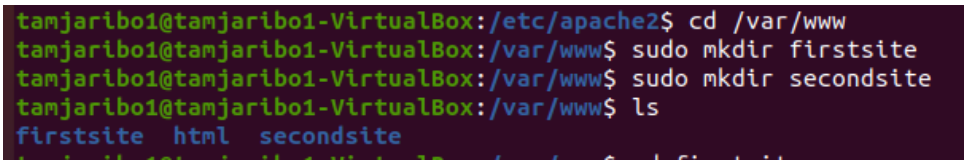
STEP 1 : Installation

```
sudo apt-get install apache2
```

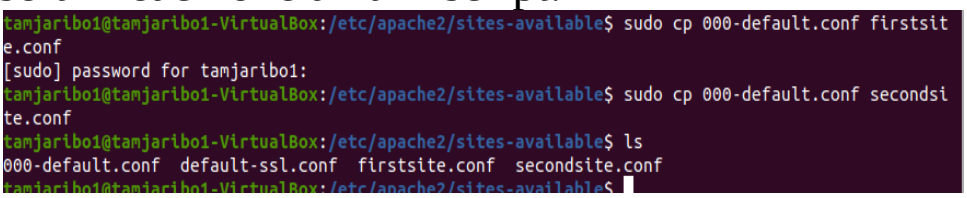


STEP 2 : Configuraton

We created two files in **/var/www**, First file called **firstsitr,** and the second one called secondsite.



Then we created two home pages (index.html) in each file and insert in each one a html script.



Here we created in **/etc/apache2/sites-available** two files **firstsite.conf** and **secondsite.conf**.

And because we have the default configuration file **"000-default.conf"** inside **/etc/apache2/sites-available**, then we copied the content of the default file, inside our two files.

```
sudo cp 000-default.conf firstsite.conf
```

```
sudo cp 000-default.conf secondsite.conf
```

After copying the **"000-default.conf"** we modify our

configuration files as we want:

→ Firstsite.conf

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    ServerName firstsite.com
    ServerAlias www.firstsite.com
    DocumentRoot /var/www/firstsite

    <Directory "/var/www/firstsite">
            AuthType Basic
            AuthName "Restricted Content"
            AuthUserFile /etc/apache2/.htpasswd
            Require valid-user
    </Directory>

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

→ Secondsite.conf

Here there is a difference is that we didn't include the part of **<directory>**, because we don't need and authentication for the second website.

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    ServerName secondsite.com
    ServerAlias www.secondsite.com
    DocumentRoot /var/www/secondsite

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

- The default **"VirtualHost"** is configured to handle any

request on port 80, the standard http port. It is defined in the declaration header where it says.

- **"80"**meaning port 80 on any interface.
- The **"ServerAdmin"** option specifies a contact email that should be used when there are server problems.
- We add a **"ServerName"** definition that specifies the domain name or IP address that the request should handle.
- We can also make the Virtual Host apply to more than one name by using the **"ServerAlias"** definition. This provides alternate paths to get to the same content, in this case we have added just **"www."** to our domain name.
- The **"DocumentRoot"** option specifies where the content that is requested for this Virtual Host will be located.
- Within the Virtual Host definition, there are definitions for how the server handles the web page, and this is the use of **"Directory".**

After creating and modifying the configuration files, we add our hostname to ***/etc/hosts***

```
1 127.0.0.1       localhost
2 127.0.1.1       tamjaribo1-VirtualBox.tamjaribo.com
3 192.168.1.2     tamjarib.com
4 192.168.1.2     ns.tamjaribo.com       ns
5 192.168.1.2     firstsite.com
6 192.168.1.2     secondsite.com
7
8 # The following lines are desirable for IPv6 capable hosts
9 ::1     ip6-localhost ip6-loopback
10 fe00::0 ip6-localnet
11 ff00::0 ip6-mcastprefix
12 ff02::1 ip6-allnodes
13 ff02::2 ip6-allrouters
```

Then we modify the reverse and forward zone that was already created (look the DNS configuration above) and we assign a domain name to the IP address:

→ Reverse zone

```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns.tamjaribo.com. root.tamjaribo.com. (
                              2                 ; Serial
                         604800                 ; Refresh
                          86400                 ; Retry
                        2419200                 ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.
2       IN      PTR     ns.tamjaribo.com.
firstsite.com   IN      PTR     192.168.1.2.
secondsite.com  IN      PTR     192.168.1.2.
```

→ Forward zone

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     tamjaribo1-VirtualBox.tamjaribo.com. root.tamjaribo.com. (
                              2                 ; Serial
                         604800                 ; Refresh
                          86400                 ; Retry
                        2419200                 ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.tamjaribo.com.
@       IN      A       192.168.1.2
@       IN      AAAA    ::1
ns      IN      A       192.168.1.2
firstsite.com   IN      A       192.168.1.2
secondsite.com  IN      A       192.168.1.2
```

To create a symbolic link in the "**sites-enabled**" directory to an existing file in the "**sites-available**" directory, we issue the following command to enable the created virtual hosts:

```
sudo a2ensite firstsite.conf
```

```
sudo a2ensite secondsite.conf
```

```
tamjaribo1@tamjaribo1-VirtualBox:~$ sudo a2ensite firstsite.conf
Enabling site firstsite.
To activate the new configuration, you need to run:
  systemctl reload apache2
tamjaribo1@tamjaribo1-VirtualBox:~$ sudo a2ensite secondsite.conf
Enabling site secondsite.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

Now after enabling our useful conf files, we will disable the useless one (the default one)

```
sudo a2dissite 000-default.conf
```

```
tamjaribo1@tamjaribo1-VirtualBox:~$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

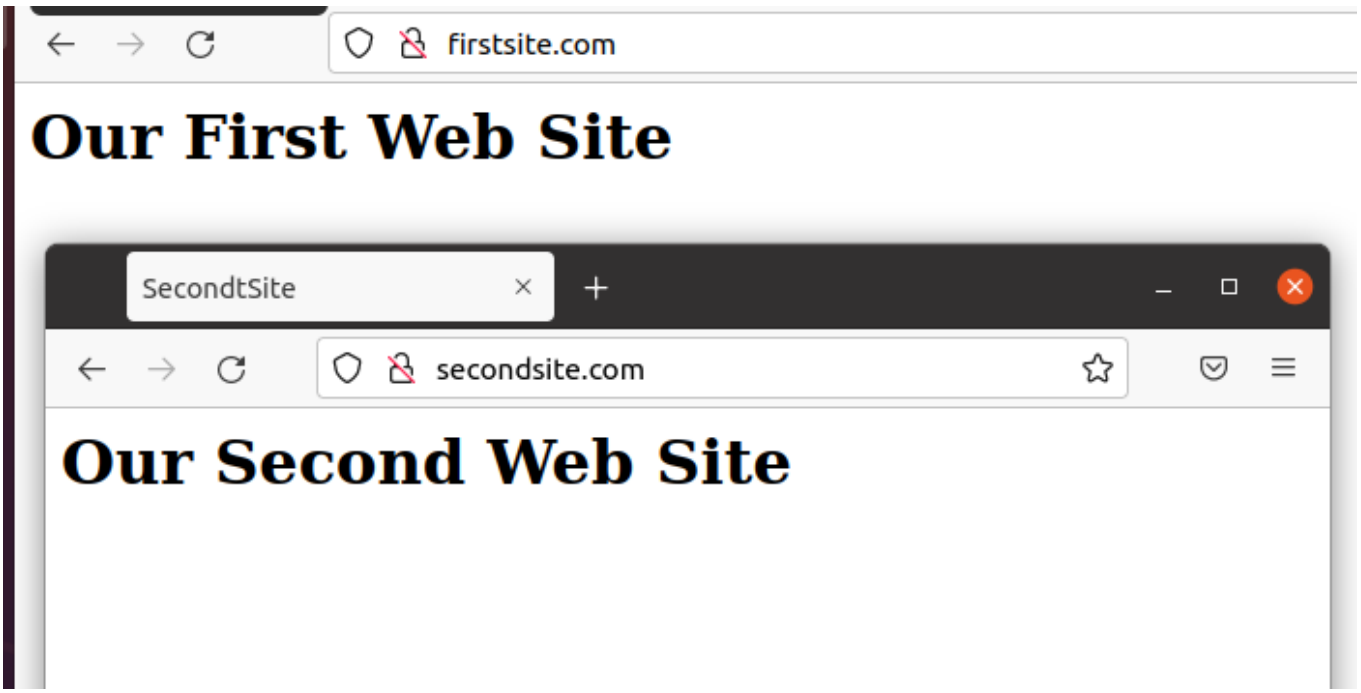
STEP 3: Test

Now we tested first the apache server by restartng it and checked the status.

```
tamjaribo1@tamjaribo1-VirtualBox:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2022-02-01 02:04:39 +01; 1min 9s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 677 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
    Process: 2589 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
   Main PID: 723 (apache2)
      Tasks: 55 (limit: 2295)
     Memory: 7.2M
     CGroup: /system.slice/apache2.service
             ├─ 723 /usr/sbin/apache2 -k start
             ├─2593 /usr/sbin/apache2 -k start
             └─2594 /usr/sbin/apache2 -k start

02:04:38 01 فبر|ير tamjaribo1-VirtualBox systemd[1]: Starting The Apache HTTP Server...
02:04:39 01 فبر|ير tamjaribo1-VirtualBox apachectl[697]: AH00558: apache2: Could not reliably d>
02:04:39 01 فبر|ير tamjaribo1-VirtualBox systemd[1]: Started The Apache HTTP Server.
02:05:44 01 فبر|ير tamjaribo1-VirtualBox systemd[1]: Reloading The Apache HTTP Server.
02:05:45 01 فبر|ير tamjaribo1-VirtualBox systemd[1]: Reloaded The Apache HTTP Server.
lines 1-19/19 (END)
```

Then we tested if our two-web sites worked on the server machine.



After that we do the same but in our client machine.

Also, we did another test in **Wireshark** to be sure 100% our server HTTP is working.

No.	Time	Source	Destination	Protocol	Length	Info
98	116.291634816	192.168.1.10	192.168.1.2	HTTP	312	GET / HTTP/1.1
100	116.295072382	192.168.1.2	192.168.1.10	HTTP	796	HTTP/1.1 401 Unauthorized (text/html)
108	120.985319150	192.168.1.10	192.168.1.2	HTTP	285	GET /favicon.ico HTTP/1.1
110	120.986323508	192.168.1.2	192.168.1.10	HTTP	796	HTTP/1.1 401 Unauthorized (text/html)
227	358.386551989	192.168.1.10	192.168.1.2	HTTP	312	GET / HTTP/1.1
229	358.386789910	192.168.1.2	192.168.1.10	HTTP	796	HTTP/1.1 401 Unauthorized (text/html)
249	367.281856759	192.168.1.10	192.168.1.2	HTTP	355	GET / HTTP/1.1
251	367.284448328	192.168.1.2	192.168.1.10	HTTP	796	HTTP/1.1 401 Unauthorized (text/html)
263	373.773932198	192.168.1.10	192.168.1.2	HTTP	359	GET / HTTP/1.1
265	373.778875381	192.168.1.2	192.168.1.10	HTTP	485	HTTP/1.1 200 OK (text/html)

3.5. Postfix Mail Server (MAIL)

Postfix consists of several small, cooperative programs that send network messages, receive messages, deliver mail locally, and more. Communication between programs is provided using Unix sockets or FIFO algorithms. The architecture is different from Sendmail, where all the work has to be done by one big program.

The master program starts and controls all processes. Its config - master.cf - lists auxiliary programs and information on how and when they need to be run.

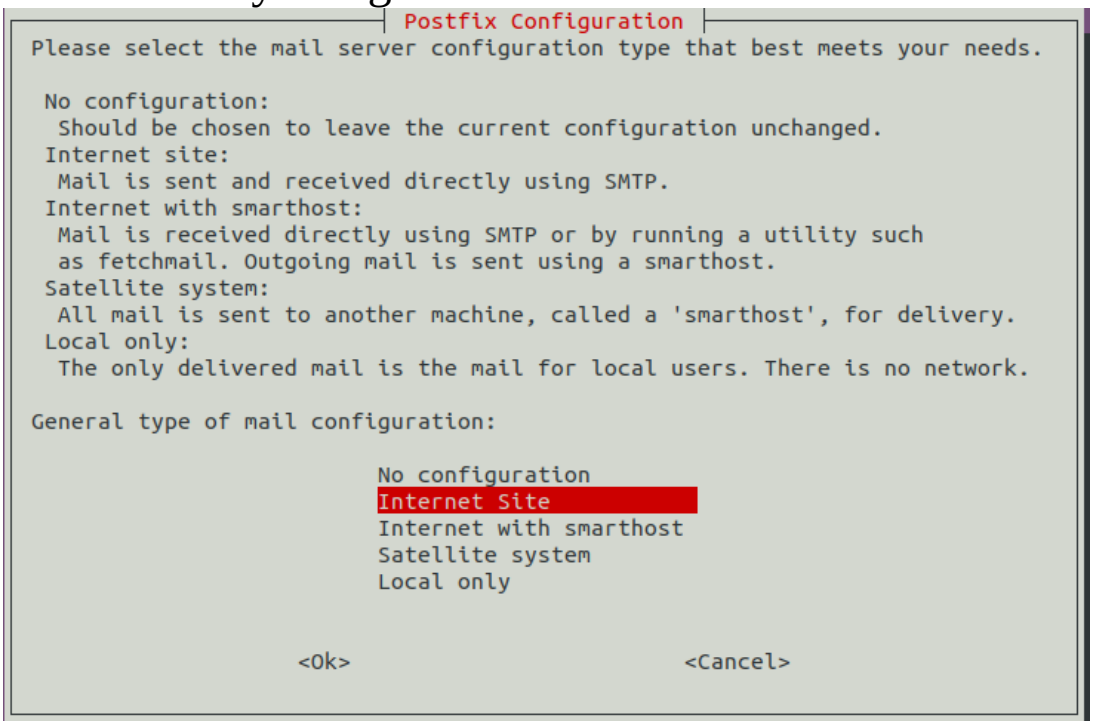
Smtpd is responsible for receiving mail on the SMTP port. It also checks if the client is authorized to send mail. If the mail is sent locally, via /usr/lib/sendmail compatibility, the file will be written to the /var/spool/postfix/maildrop directory. This directory is scanned by the **pickup** program, which processes the found files. Incoming mail is processed by the **cleanup** program. It adds the missing headers and rewrites the addresses according to the **canonical** and **virtual** maps. Before the email is placed on the queue, the incoming cleanup program passes the email to the **trivial-rewrite** program, which also performs minor corrections to the addresses, adding the domain and the partially filled address.

STEP1: Insallation
We installed the package needed

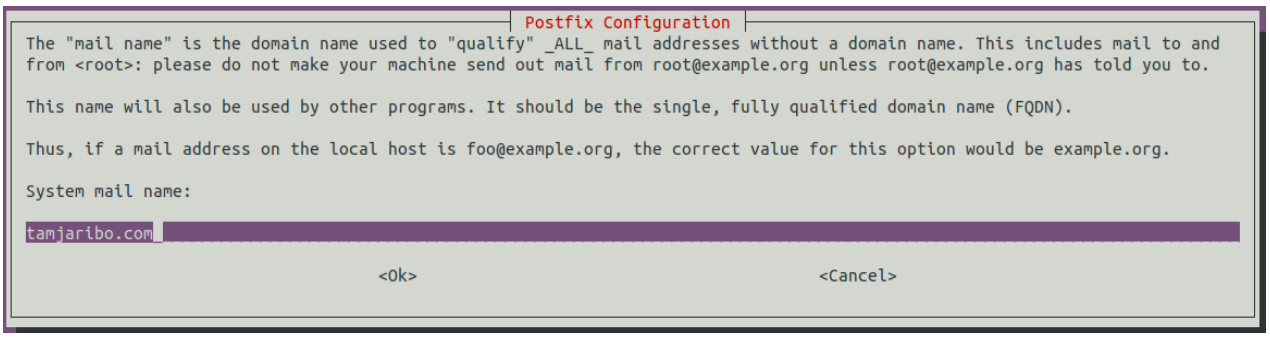
```
Sudo apt-get install postfix
Sudo apt install dovecot-imapd dovecot=pop3d
```

```
tamjaribo@tamjaribo-VirtualBox:~$ sudo apt install dovecot-imapd dovecot-pop3d
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  dovecot-core
Suggested packages:
  dovecot-gssapi dovecot-ldap dovecot-lmtpd dovecot-lucene dovecot-managesieved dovecot-mysql dov
  dovecot-solr dovecot-sqlite dovecot-submissiond ntp
The following NEW packages will be installed:
  dovecot-core dovecot-imapd dovecot-pop3d
0 upgraded, 3 newly installed, 0 to remove and 191 not upgraded.
Need to get 3,111 kB of archives.
After this operation, 11.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

STEP 2: Configuration
We strated by configuring the Postfix:
We chose the mail configuration, our mails are sent and recovered directly using SMTP.



We type our domain name.



Postfix is an SMTP server, find out its listening port with nmap.

```
tamjaribo1@tamjaribo1-VirtualBox:~$ nmap localhost
Starting Nmap 7.80 ( https://nmap.org ) at 2022-02-01 08:35 +01
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0011s latency).
Not shown: 990 closed ports
PORT    STATE SERVICE
21/tcp  open  ftp
22/tcp  open  ssh
25/tcp  open  smtp
53/tcp  open  domain
80/tcp  open  http
110/tcp open  pop3
143/tcp open  imap
631/tcp open  ipp
993/tcp open  imaps
995/tcp open  pop3s

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
```

We created two users: assia and hamza

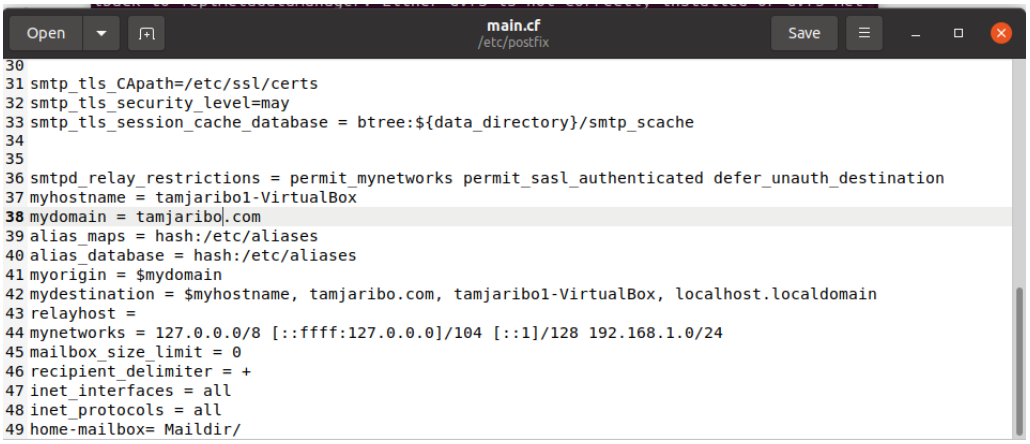
```
tamjaribo1@tamjaribo1-VirtualBox:~$ sudo useradd -m assia
[sudo] password for tamjaribo1:
tamjaribo1@tamjaribo1-VirtualBox:~$ sudo useradd -m hamza
```

Set the passwords of our two users

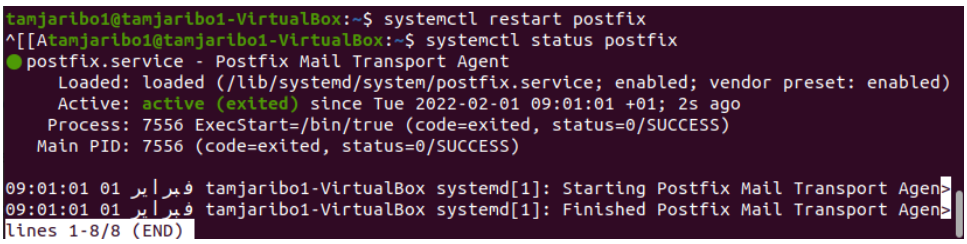
```
tamjaribo1@tamjaribo1-VirtualBox:~$ sudo passwd assia
New password:
Retype new password:
passwd: password updated successfully
tamjaribo1@tamjaribo1-VirtualBox:~$ sudo passwd hamza
New password:
Retype new password:
passwd: password updated successfully
```

Edit the main postfix configuration file **/etc/postfix/main.cf.**

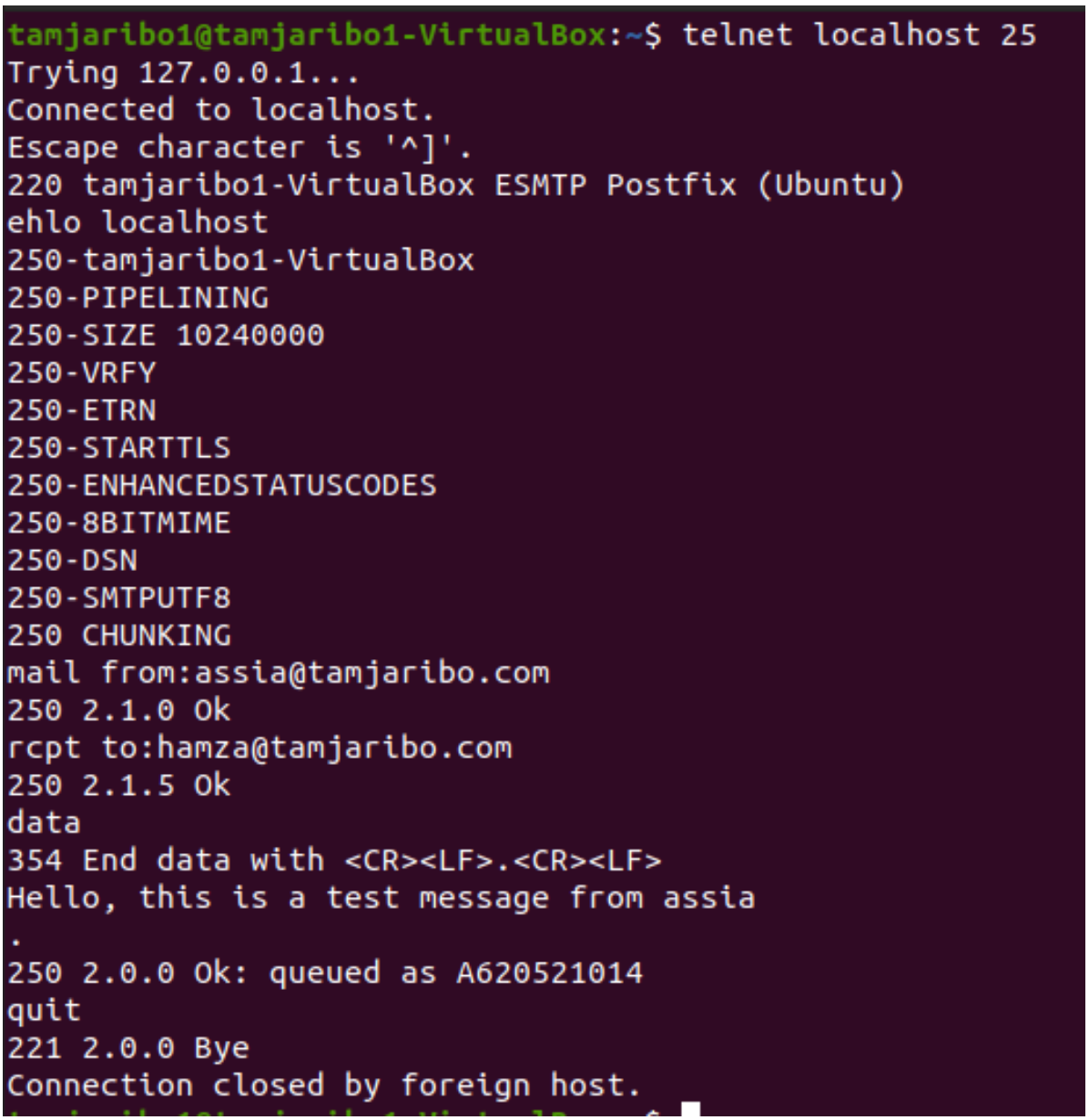
- Change the directives that follow as indicated



- The mydomain directive must be added.
- Add the 192.168.1.0/24 network to the end of the mynetwork entry (x depends on the private network that Ubuntu and our Windows belong to.
- At the end of the filewe add **Maildir/**
- After that we restart the Postfix



Testing local postfix access from the command line between the two users:



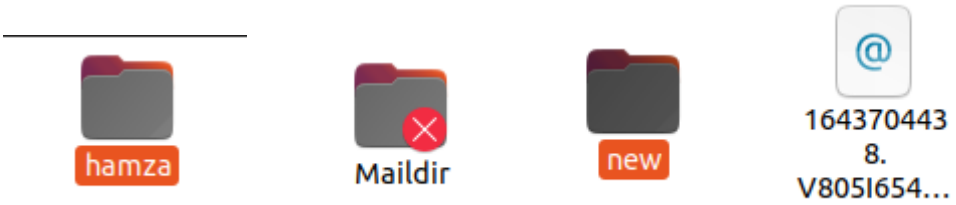
- **ehlo localhost**: identifies the sending computer (either using the IP address or the FQDN).

The server responds with lines with a return **code = 250**;

250 means: Messaging action completed, success

- **data** = email body
- **quit** = SMTP server exit

We checked in our data base the files contaning the message content




```
 1 Return-Path: <assia@tamjaribo.com>
 2 X-Original-To: hamza@tamjaribo.com
 3 Delivered-To: hamza@tamjaribo.com
 4 Received: from localhost (localhost [127.0.0.1])
 5         by tamjaribo1-VirtualBox (Postfix) with ESMTP id EE35E21014
 6         for <hamza@tamjaribo.com>; Tue,  1 Feb 2022 09:32:49 +0100 (+01)
 7 Message-Id: <20220201083329.EE35E21014@tamjaribo1-VirtualBox>
 8 Date: Tue,  1 Feb 2022 09:32:49 +0100 (+01)
 9 From: assia@tamjaribo.com
10
11 Hello, this is my test message from assia
```

After the installation of dovecot, we checked its status

```
tamjaribo1@tamjaribo1-VirtualBox:~$ systemctl status dovecot
● dovecot.service - Dovecot IMAP/POP3 email server
     Loaded: loaded (/lib/systemd/system/dovecot.service; enabled; vendor prese>
     Active: active (running) since Tue 2022-02-01 09:42:27 +01; 30s ago
       Docs: man:dovecot(1)
             http://wiki2.dovecot.org/
   Main PID: 4551 (dovecot)
      Tasks: 4 (limit: 2295)
     Memory: 3.0M
     CGroup: /system.slice/dovecot.service
             ├─4551 /usr/sbin/dovecot -F
             ├─4552 dovecot/anvil
             ├─4553 dovecot/log
             └─4554 dovecot/config

09:42:27 01 فبراير tamjaribo1-VirtualBox systemd[1]: Started Dovecot IMAP/POP3 >
09:42:27 01 فبراير tamjaribo1-VirtualBox dovecot[4551]: master: Dovecot v2.3.7.>
```

Now we edited the file **/etc/dovecot/dovecot.conf**.

```
Open   *dovecot.conf  /etc/dovecot   Save

 95 # Most of the actual configuration gets included below. The filenames are
 96 # first sorted by their ASCII value and parsed in that order. The 00-prefixes
 97 # in filenames are intended to make it easier to understand the ordering.
 98 !include conf.d/*.conf
 99 # A config file can also tried to be included without giving an error if
100 # it's not found:
101 !include_try local.conf
102 protocols = pop3 imap
103 mail_location = maildir:~/Maildir
104
```

We add at the end of the file the following
protocols = pop3 imap
mail_location = maildir:~/Maildir

- The **protocols** directive indicates the protocol versions supported by dovecot (there are also pop3s and imaps which are the secure versions of pop3 and imap).
- The **mail_location** directive indicates the location from which dovecot will read mails for submit to MUA.

Now we restart the dovecot server

```
tamjaribo1@tamjaribo1-VirtualBox:~$ systemctl restart dovecot
tamjaribo1@tamjaribo1-VirtualBox:~$ systemctl status dovecot
● dovecot.service - Dovecot IMAP/POP3 email server
     Loaded: loaded (/lib/systemd/system/dovecot.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2022-02-01 09:48:40 +01; 1s ago
       Docs: man:dovecot(1)
             http://wiki2.dovecot.org/
   Main PID: 4626 (dovecot)
      Tasks: 4 (limit: 2295)
     Memory: 3.0M
     CGroup: /system.slice/dovecot.service
             ├─4626 /usr/sbin/dovecot -F
             ├─4627 dovecot/anvil
             ├─4628 dovecot/log
             └─4629 dovecot/config

09:48:40 01 فبراير tamjaribo1-VirtualBox systemd[1]: Started Dovecot IMAP/POP3 email serve>
09:48:40 01 فبراير tamjaribo1-VirtualBox dovecot[4626]: master: Dovecot v2.3.7.2 (3c910f64>
lines 1-16/16 (END)
```

We will test access to dovecot locally from a terminal:

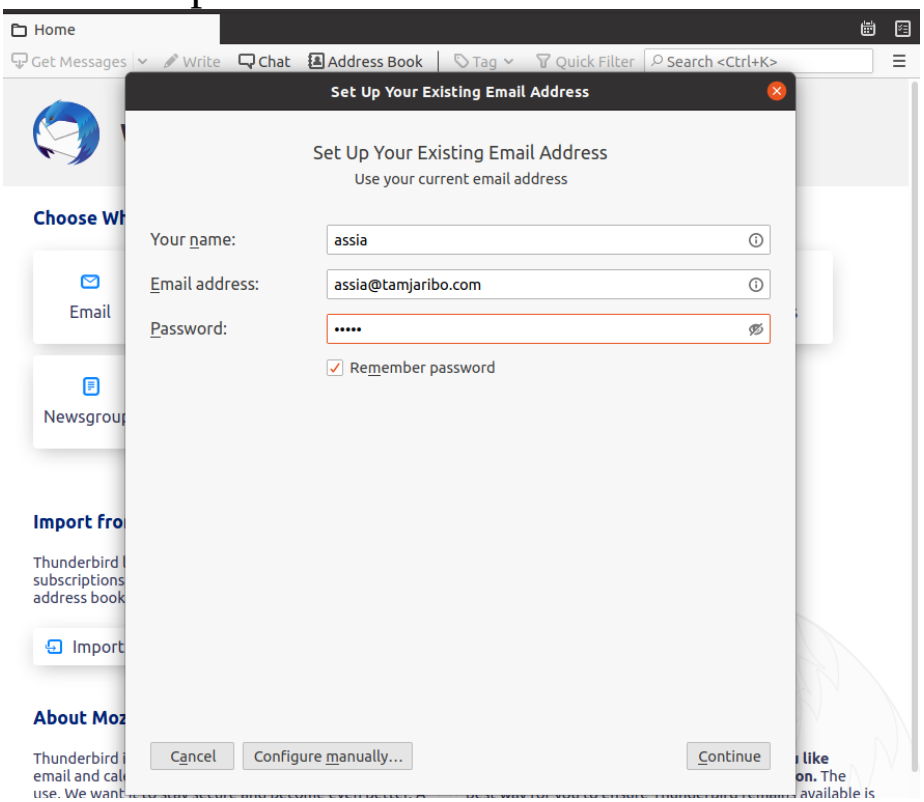
```
tamjaribo1@tamjaribo1-VirtualBox:~$ telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK Dovecot (Ubuntu) ready.
user hamza
+OK
pass hamza
+OK Logged in.
list
+OK 1 messages:
1 466
.
quit
+OK Logging out.
Connection closed by foreign host.
```

telnet localhost 110

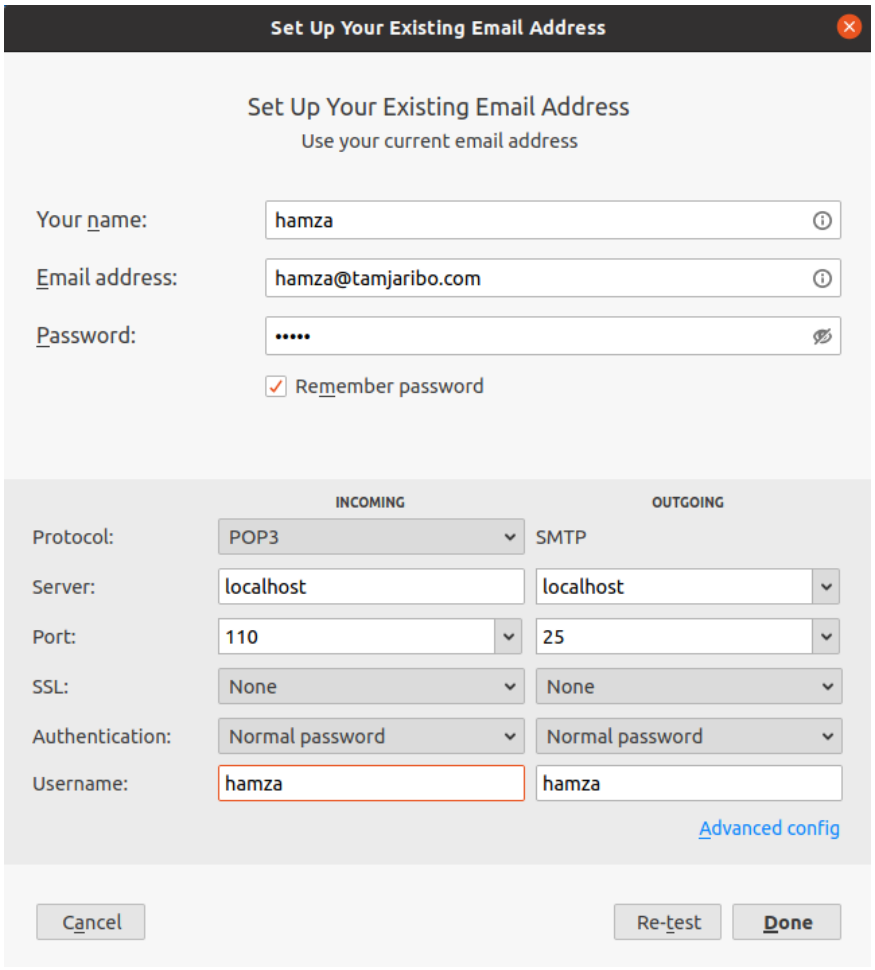
It shows us:

+OK Dovecot Ready
We type:
user hamza
then
pass hamza
It showsus:
+OK logged in
user and **pass** are pop3 commands!
We type:
List
It shows us:
+OK 1 message:

Now we are going to configure **Thunderbird Mail** (an email client)
Here we enter the parameters of our user:



We are now configuring the incoming server (for receiving mail) and the outgoing server (for sending mail):



✓ incoming server: **pop3** (instead of IMAP), server name: **localhost**, port: **110**

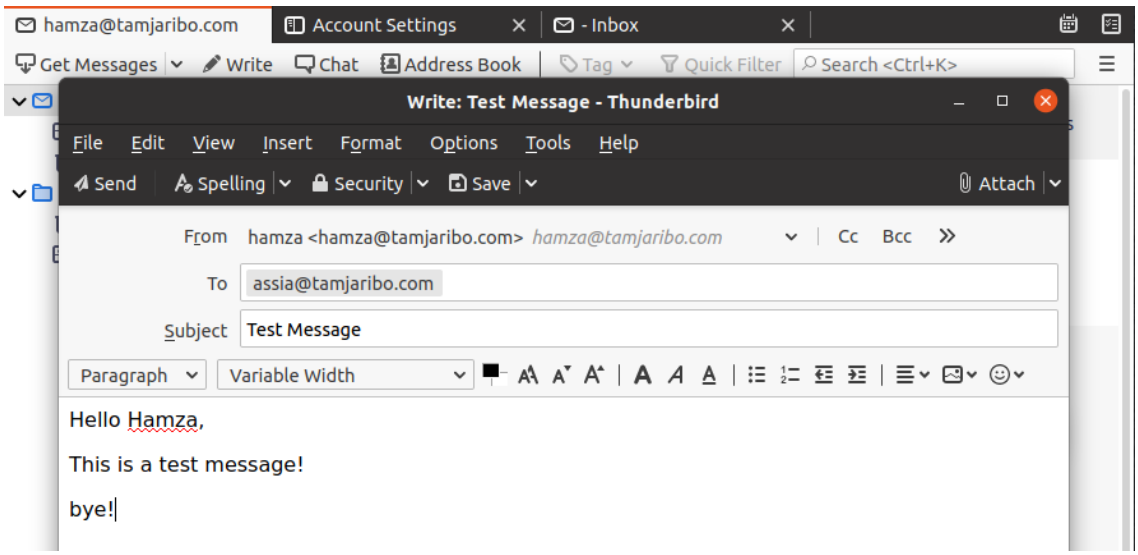
✓ outgoing server: **smtp**, server name: **localhost**, port: **25**

✓ SSL: none

To secure the connection with the servers (message encryption, authentication, etc.),
we had to choose STARTTLS or SSL/TLS for SSL (Secure Socket Layer).

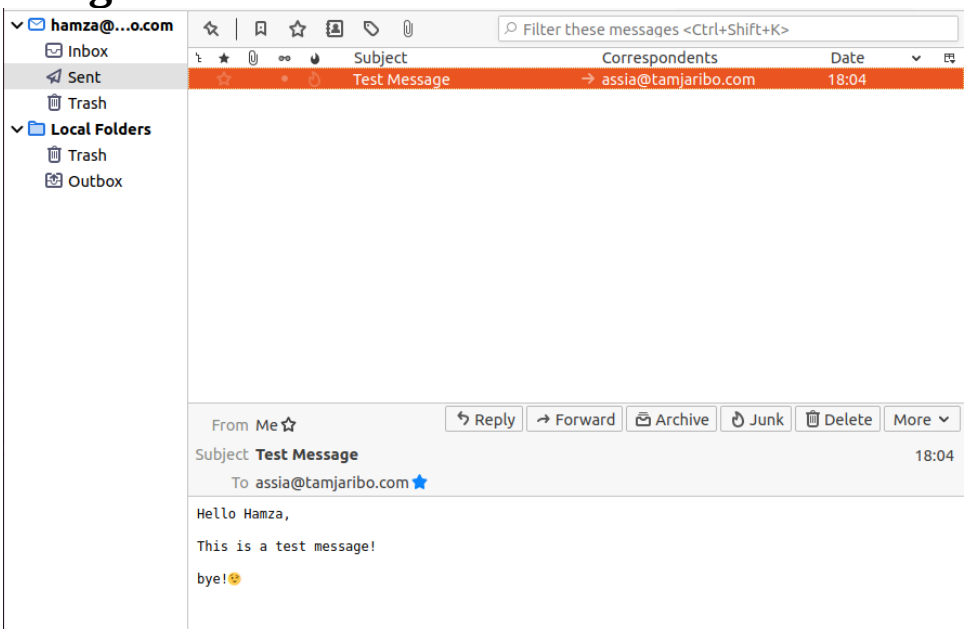
✓ As the authentication method, choose: **normal password**

→ **Message sent**



- If we chose SSL: None, Thunderbird tells us that our connection will not be secure because there is no encryption. You must accept by ticking "I understand the risks"
- If we choose STARTTLS or SSL/TLS, Thunderbird asks us to add a security exception because the certificate it will use is self-generated: it is not issued by a trusted authority, it is not sure. In this case, confirm the security exception.

→ **Message received**



Step 3: Test two clients

We now want to simulate sending/receiving an email between hamza and assia using two mail clients, Thunderbird (hamza) on ubuntu, Thunderbird (hamza) on another ubuntu machine.

We will configure dovecot:

We will modify three files: **10-mail.conf**, **10-auth.conf**, and **10-master.conf**.

→ **10-mail.conf**

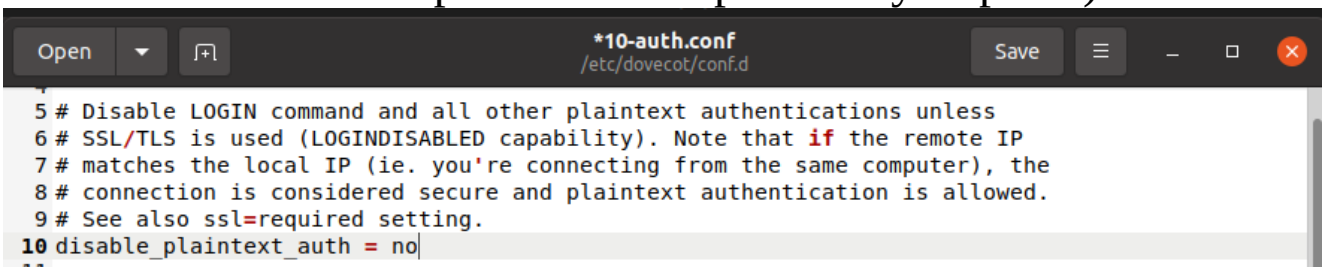
We uncomment the directive: **mail_location = maildir:~/Maildir**

→ **10-auth.conf**

We enable the disable_plaintext_auth directive for plaintext authentication (the identifiers will circulate in the clear):

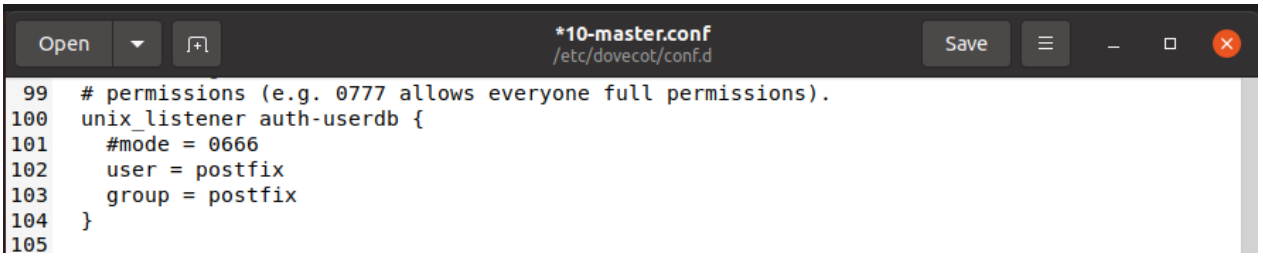
disable_plaintext_auth = no

We add to the **auth_mechanisms** directive the value **login** (the different mechanisms possible are separated by a space!).



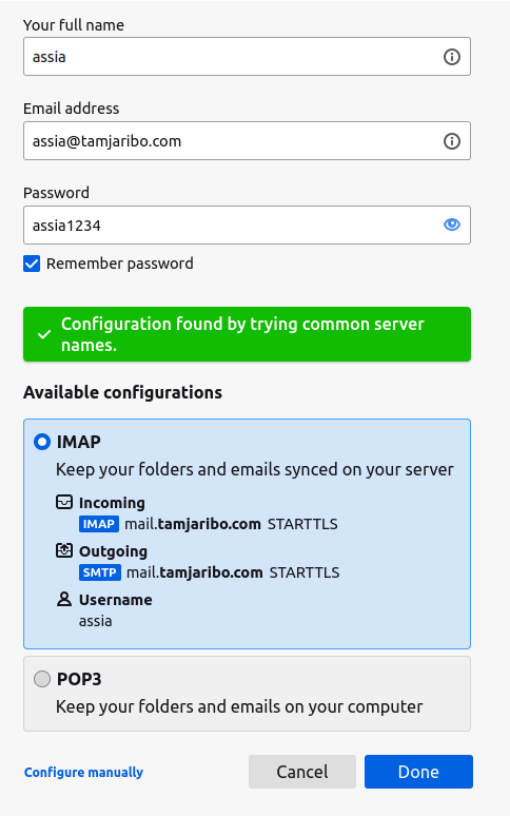
→ **10-master.conf**

We look for the entry: **service auth and** modify it as follows:

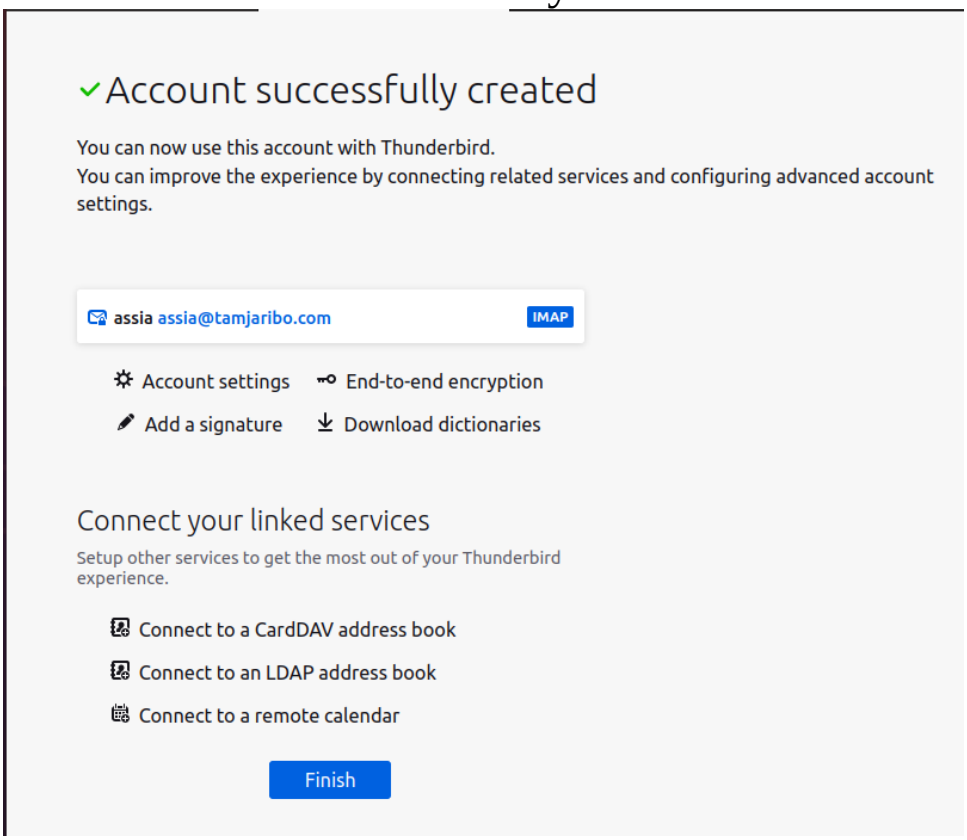


After that we restart dovecot

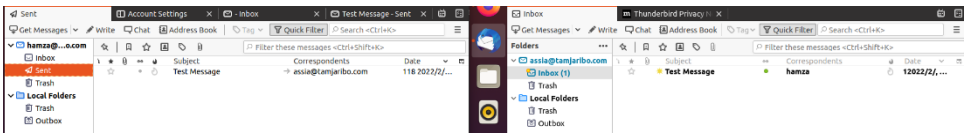
The first Client



We add our mail successfully



Send and received successfully



3.6. File Transfer Protocol (FTP)

It's time to organize file storage, both inside the network and outside. There are many ftp servers for linux, many good ones, many bad ones, but there is a better one and that is **vsftpd**

```
Sudo apt-get install vsftpd
```

Once the installation completes, the service will be disabled initially, therefore, we need to start it manually for the mean time and also enable it to start automatically from the next system boot:

```
Sudo systemctl restart vsftpd.service
Sudo systemctl status vsftpd.service
```

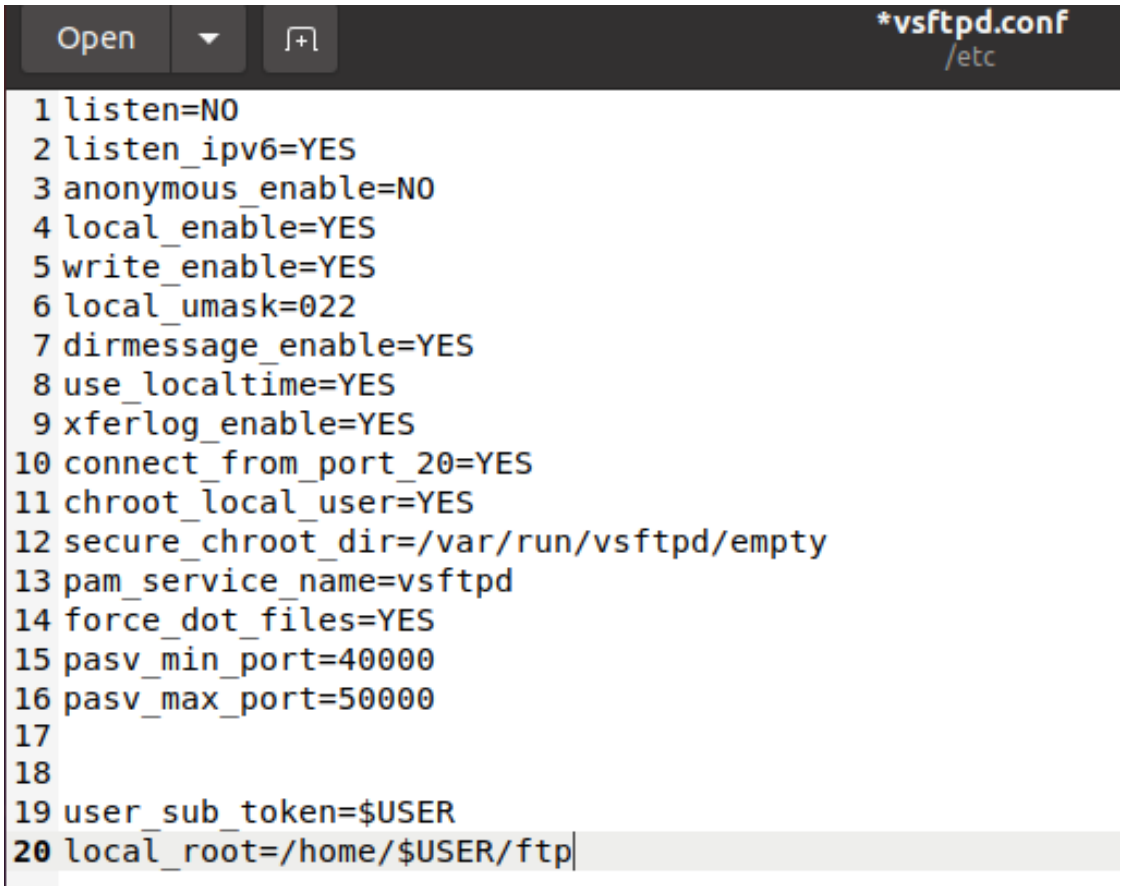
Next, we enable the ufw firewall on the server, and we have to open ports **21** and **20** where the FTP daemons are listening, in order to allow access to FTP services from remote machines, and then we add the new firewall rules as follows:

```
Sudo ufw allow 20/tcp
Sudo ufw allow 20/tcp
sudo ufw status
```

After that we take a look on the status to see the ports opened:

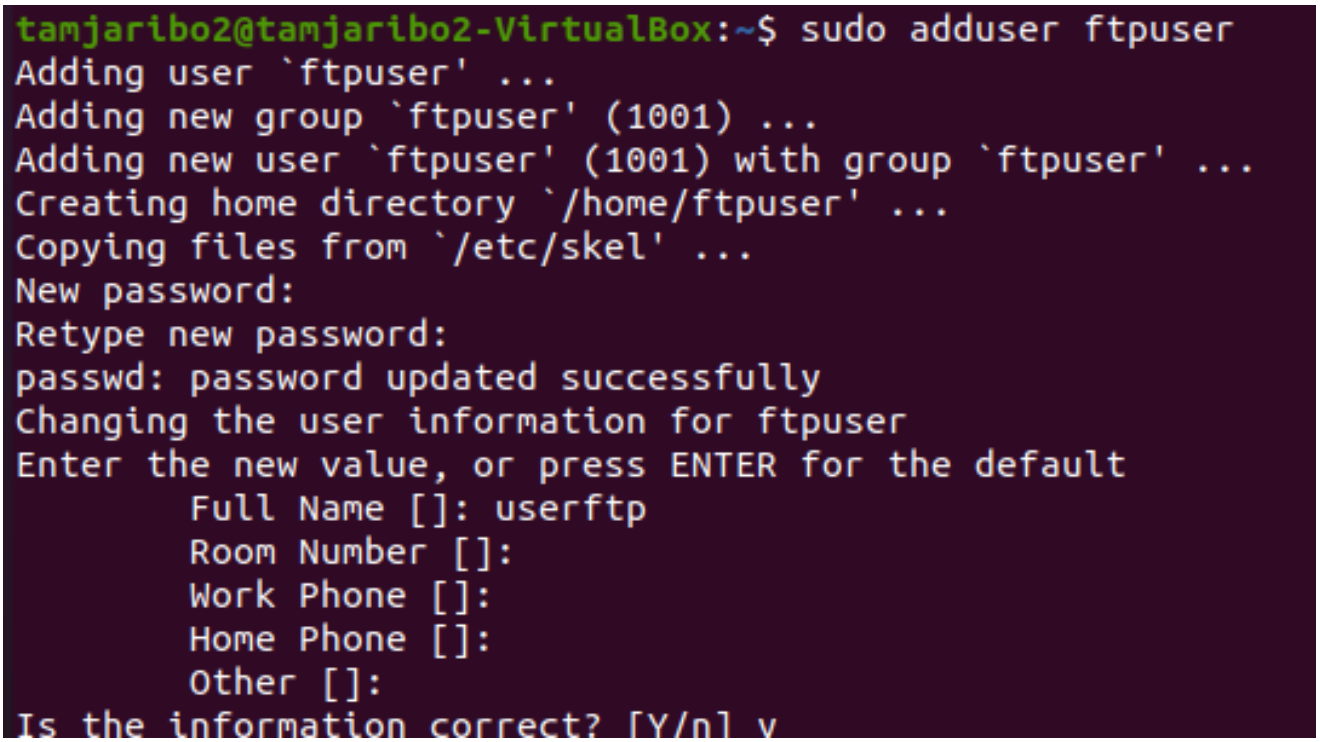
Let’s now perform a few configurations to setup and secure our FTP server, we open the configuration file vsftpd.conf

```
Sudo gedit /etc/vsftpd.conf
```

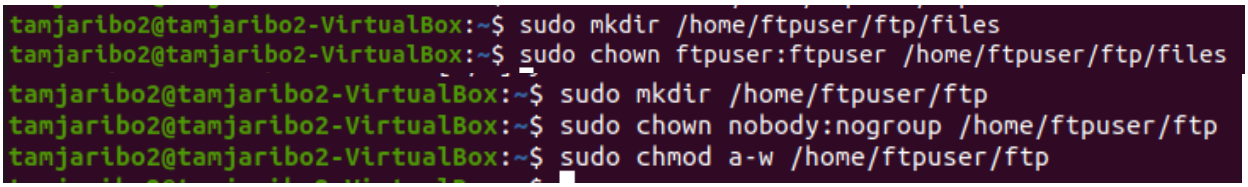


After that, we restart out server

```
Sudo adduser ftpuser
```

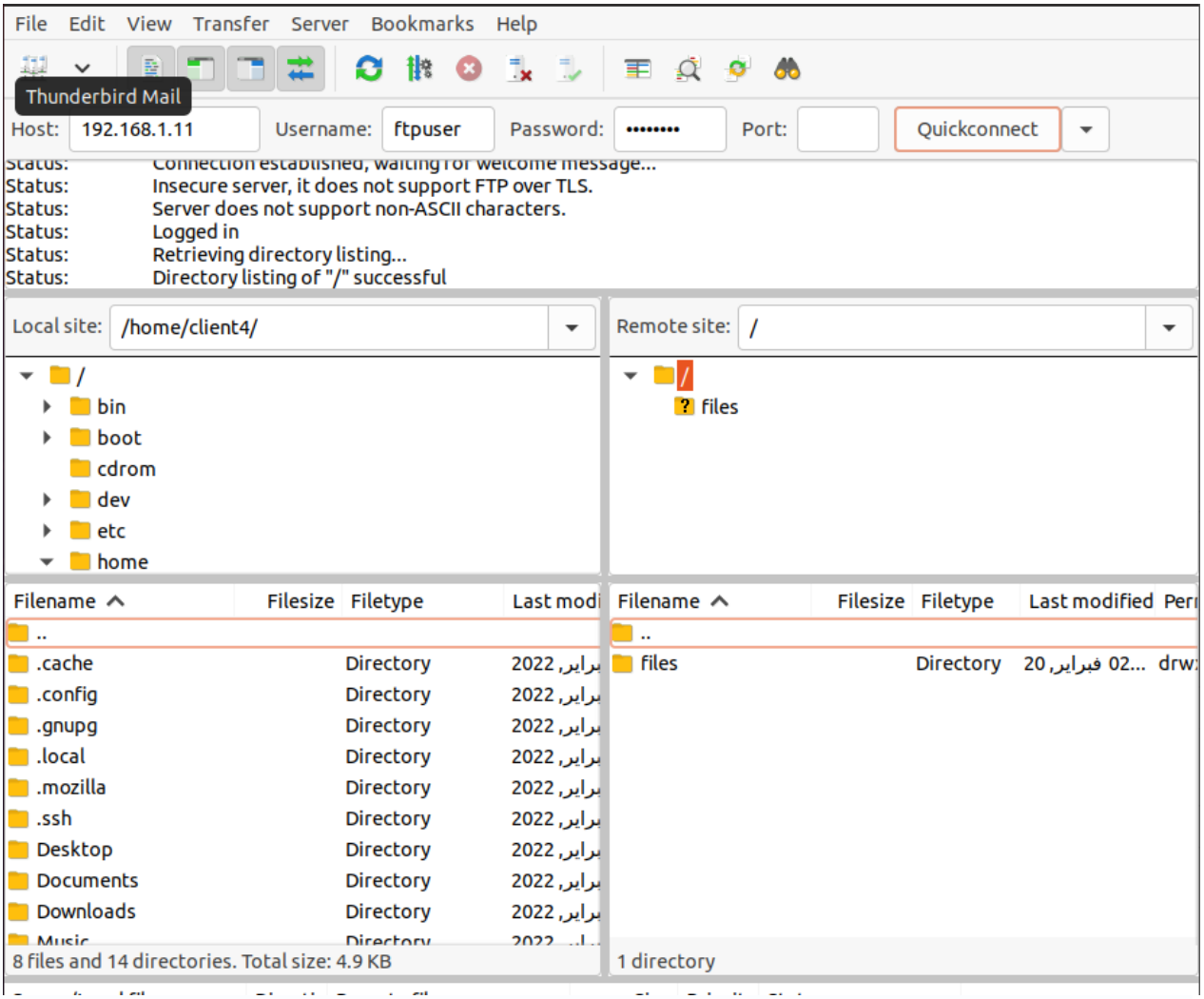


And then, we create an FTP folder, and set the ownership



After that, we install filezilla

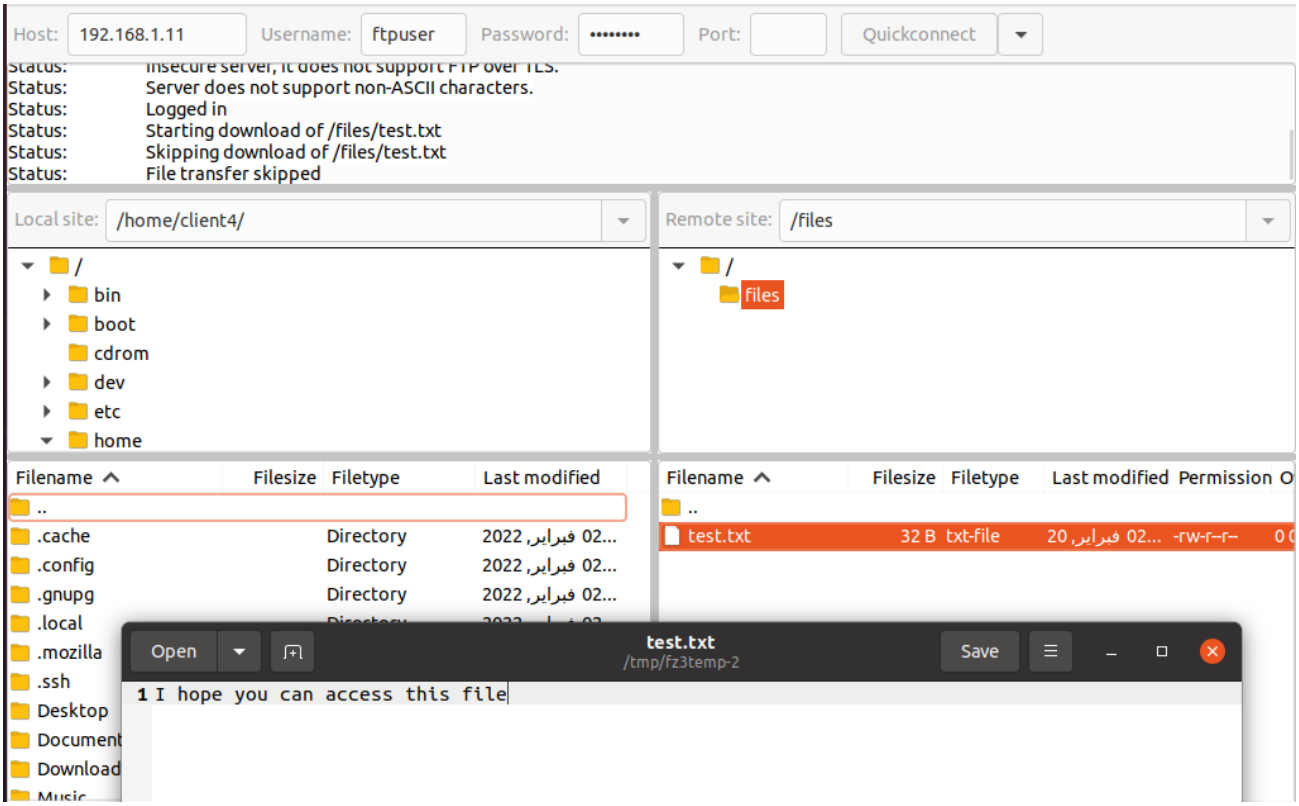
```
Sudo apt-get install filezilla
```



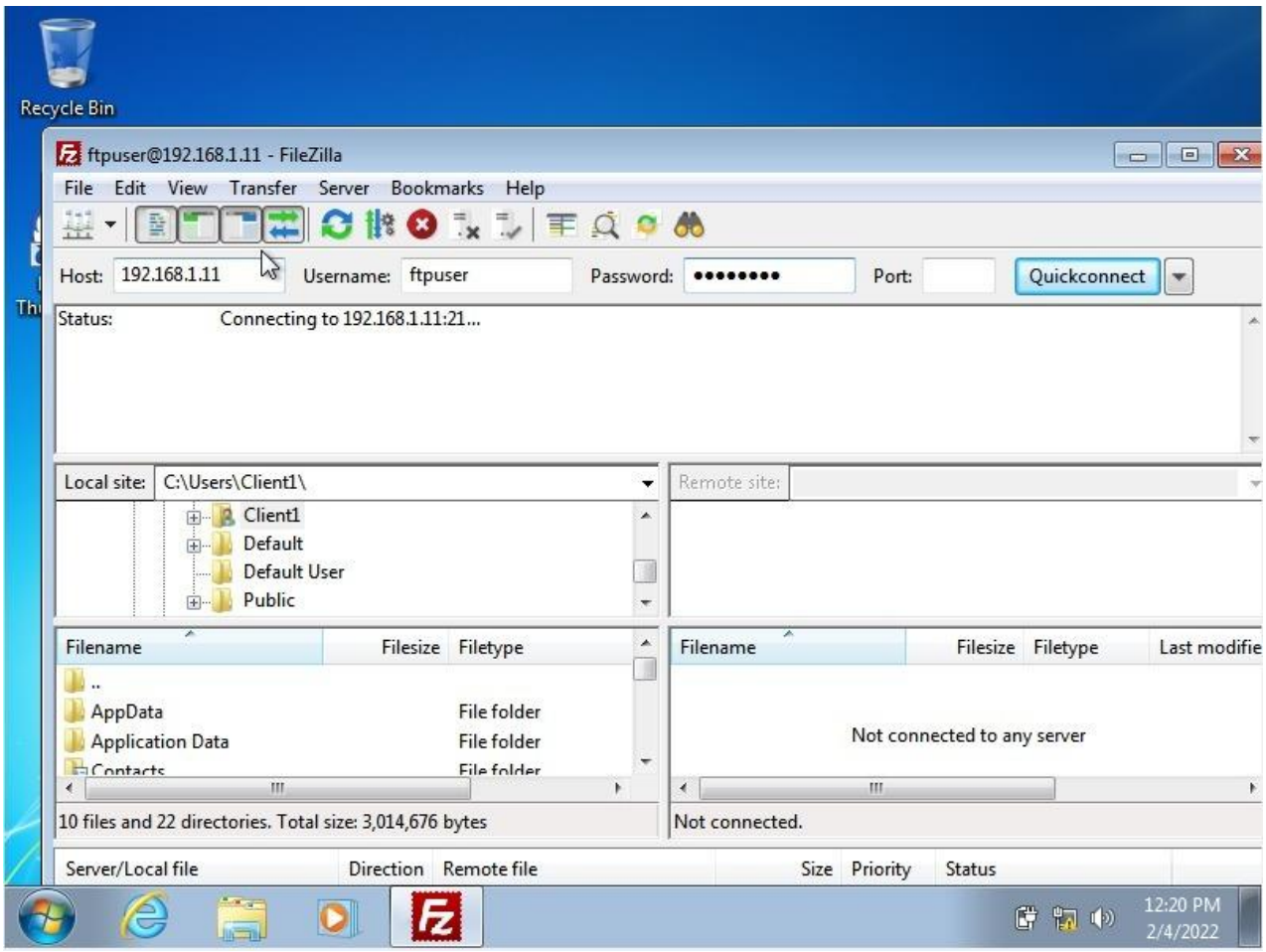
We are now in the process of testing, we will create a file “test.txt” and we will try to access it.

```
tamjaribo2@tamjaribo2-VirtualBox:~$ sudo gedit /home/ftpuser/ftp/files/test.txt
```

Here, with “filezilla” we have that “.txt” file, we open it and we found the same file edited previously.



After that, we tested in a machine that belongs to the second subnet, (whose IP is different from the the first one) we should have the next output.


Recycle Bin
ftpuser@192.168.1.11 - FileZilla
File Edit View Transfer Server Bookmarks Help
Host: 192.168.1.11 Username: ftpuser Password: •••••••• Port: Quickconnect
Status: Connecting to 192.168.1.11:21...
Local site: C:\Users\Client1\
Client1
Default
Default User
Public
Remote site:
Filename Filesize Filetype
AppData File folder
Application Data File folder
Filename Filesize Filetype Last modifie
Not connected to any server
10 files and 22 directories. Total size: 3,014,676 bytes
Not connected.
Server/Local file Direction Remote file Size Priority Status
12:20 PM
2/4/2022

3.6. Lightweight Directory Access Protocol (LDAP)

LDAP, or Lightweight Directory Access Protocol, is a communication *protocol* that defines the methods in which a directory service can be accessed. More broadly, LDAP defines the way in which data within a directory service should be presented to users, defines the requirements for the components used to create data entries within a directory service, and describes the way in which various primitive elements are used to compose entries.
Because LDAP is an open protocol, there are many different implementations. The OpenLDAP project is one of the most well-maintained open-source options.

STEP1: Installation

To install SLAP and other LDAP utilities, run the command below;

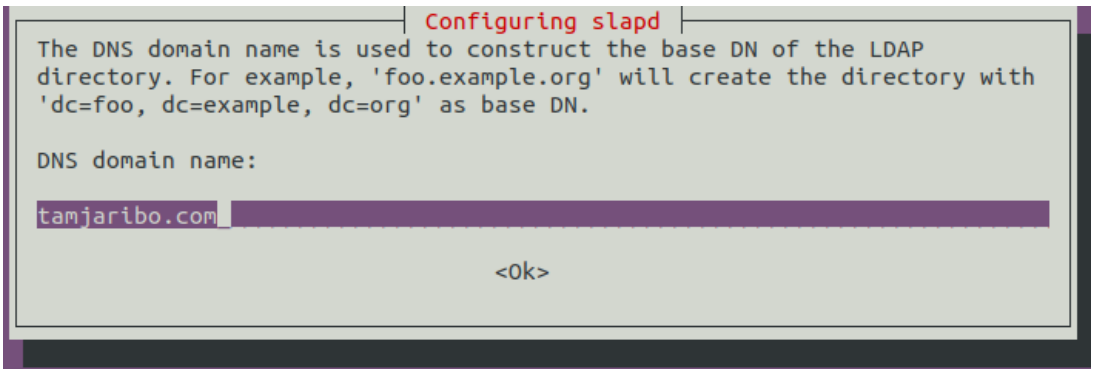
```
apt install slapd ldap-utils
```

During the installation, we will be promoted to set the OpenLDAP administrative password. Then we will need to reconfigure our SLAPD packages

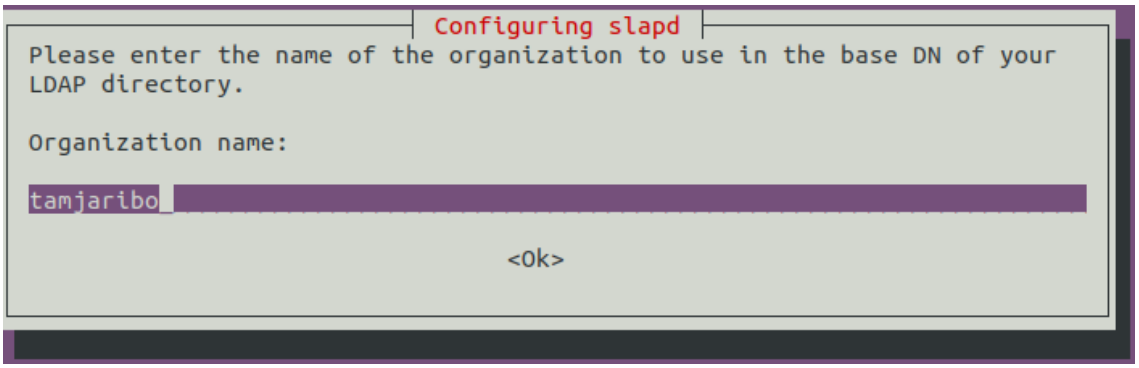
```
dpkg-reconfigure slapd
```

We run this command and set the following settings:

- Set our DNS domain name for constructing the base DN of our LDAP directory.

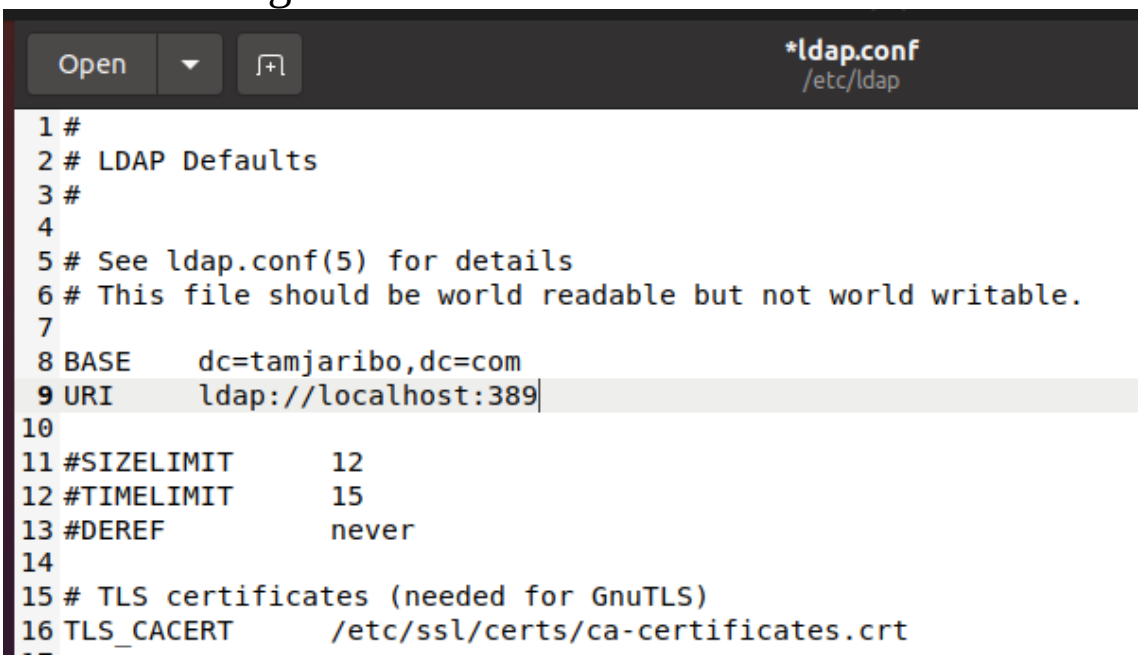


- Enter the name of our organization to be used in the base DN.



STEP2: Configuration

We will configure our LDAP.conf file and set our domain name:



Now, let’s search and show our default configuration by using the following command:

```
tamjaribo2@tamjaribo2-VirtualBox:~$ ldapsearch -x -LLL -H ldap:/// -b dc=tamjari
bo,dc=com
dn: dc=tamjaribo,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: tamjaribo
dc: tamjaribo

dn: cn=admin,dc=tamjaribo,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
```

Once we have OpenLDAP running, it is time to install phpLDAPadmin which will help you manage your LDAP server. phpLDAPadmin requires PHP and Apache web server installed on your Ubuntu system.

First we will install the phpLDAPadmin :

```
sudo apt install phpldapadmin
```

Then configure it :

My LDAP Server: The name of our LDAP server. This is the name that will appear on phpLDAPadmin web interface.

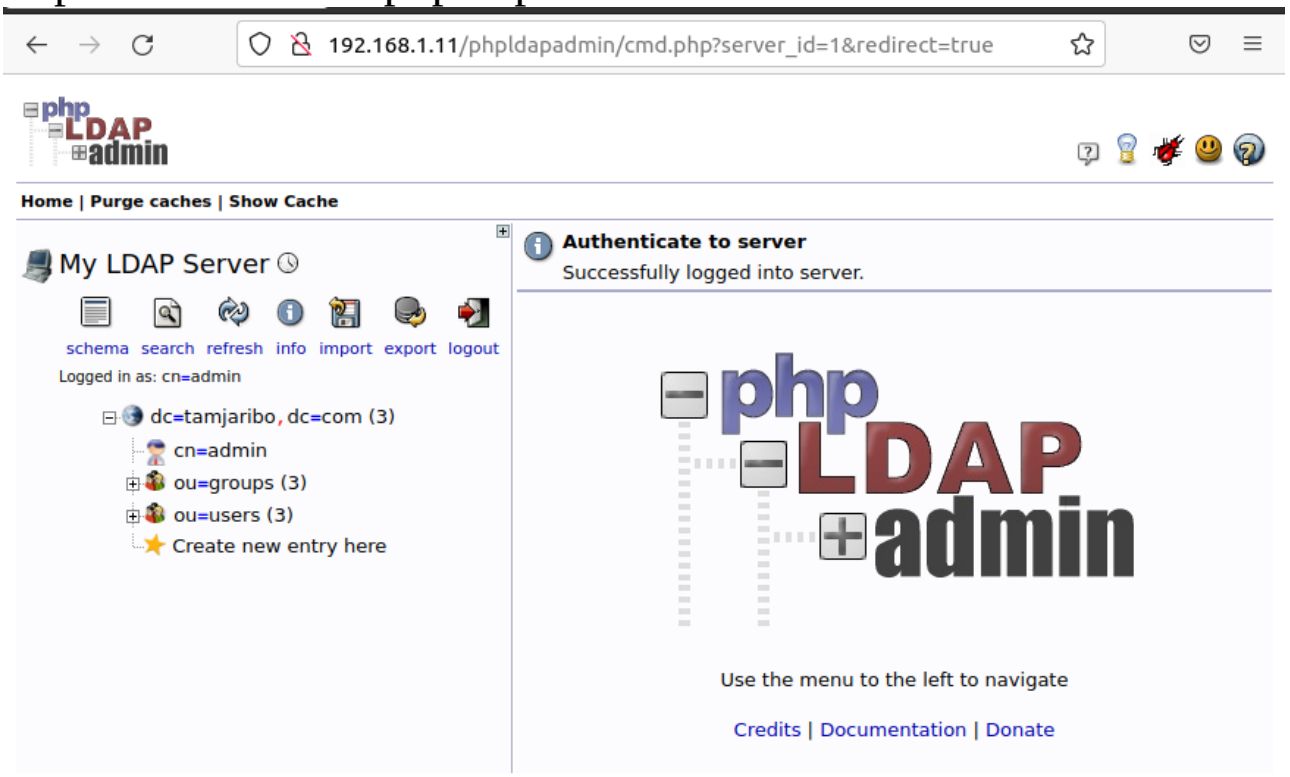
```
293 $servers->setValue('server','host','192.168.1.11');
```

192.168.1.11: We define the IP address of our OpenLDAP server;

```
307
308 /* The DN of the user for phpLDAPadmin to bind with. For anonymous binds or
309    'cookie','session' or 'sasl' auth_types, LEAVE THE LOGIN_DN AND LOGIN_PASS
310    BLANK. If you specify a login_attr in conjunction with a cookie or session
311    auth_type, then you can also specify the bind_id/bind_pass here for searching
312    the directory for users (ie, if your LDAP server does not allow anonymous
313    binds. */
314 $servers->setValue('login','bind id','cn=admin,dc=tamjaribo,dc=com');
```

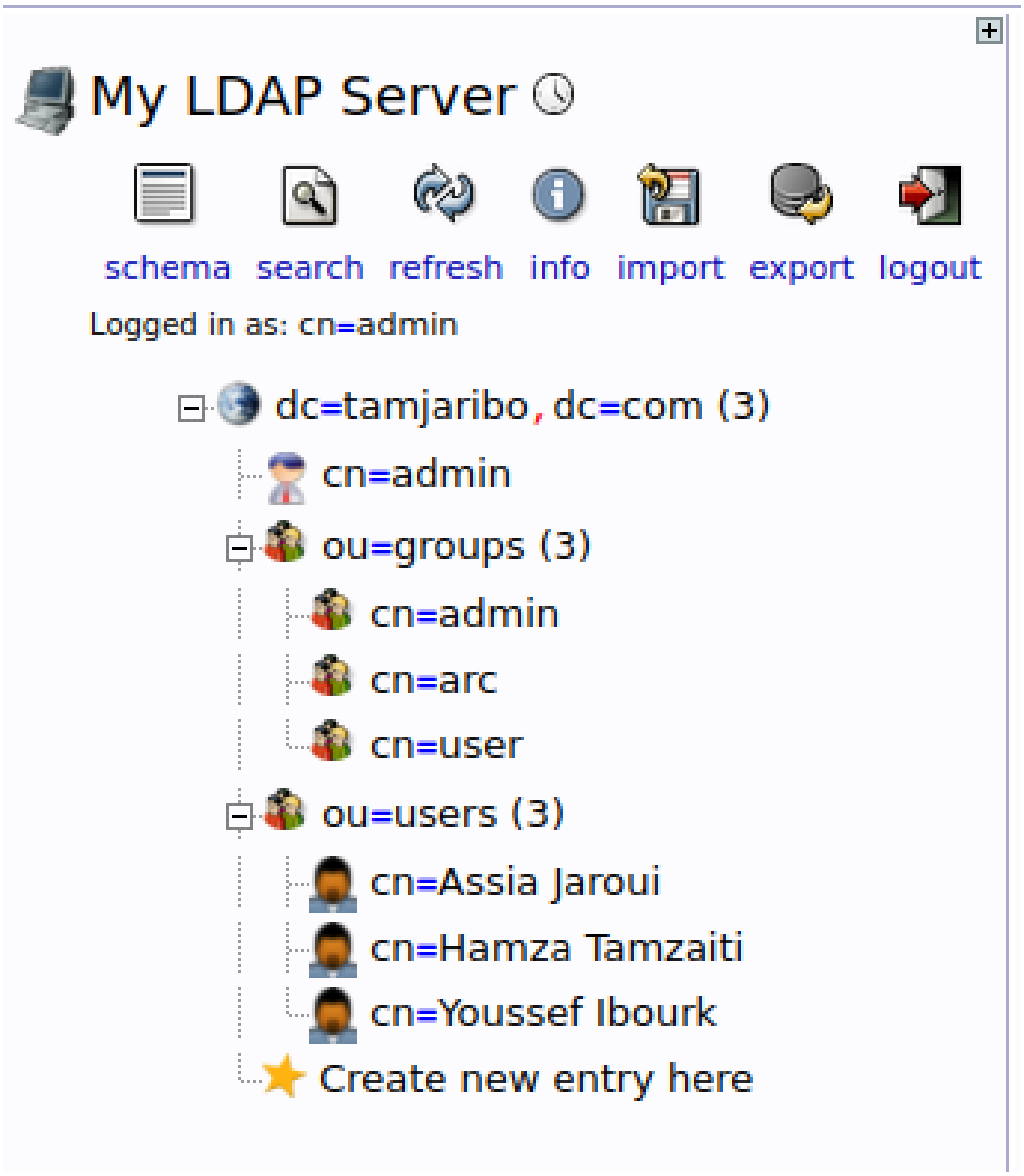
cn=admin: We define the Bind DN of the administrative user to login to phpLDAPadmin;

We will try to log in with our LDAP administrator and check In order to access phpLDAPadmin we use the address ip like follow:
http://**192.168.1.11**/phpldapadmin.



It appears that it is successfully logged into server

Now we will add some groups and users for authentication

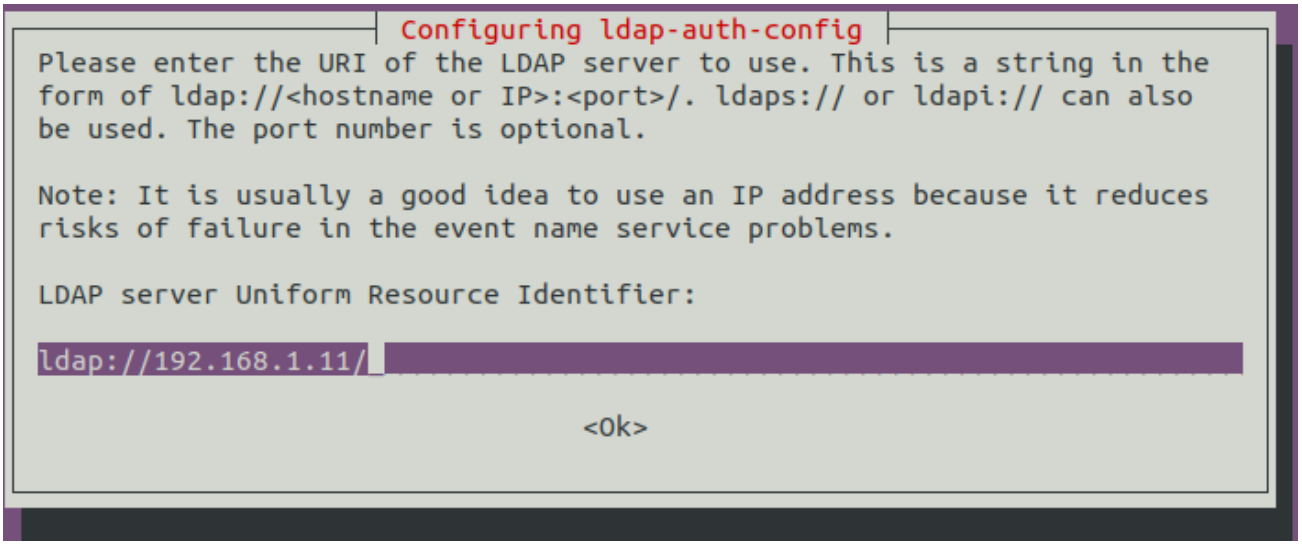


STEP3: Test to log in from a client machine:

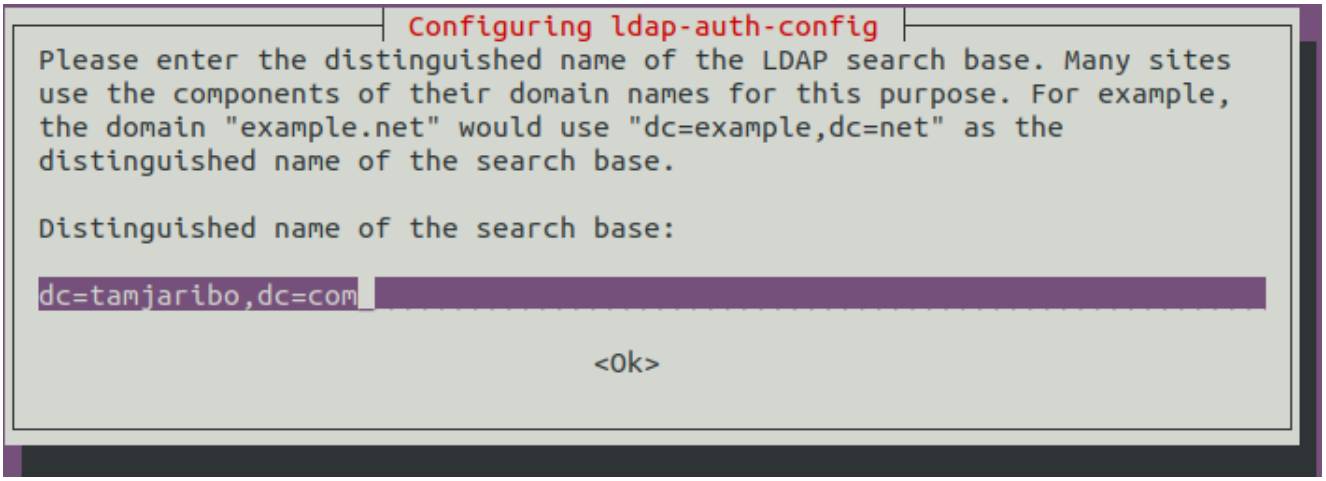
We first install the LDAP client:

```
sudo apt-get install ldap-auth-client
```

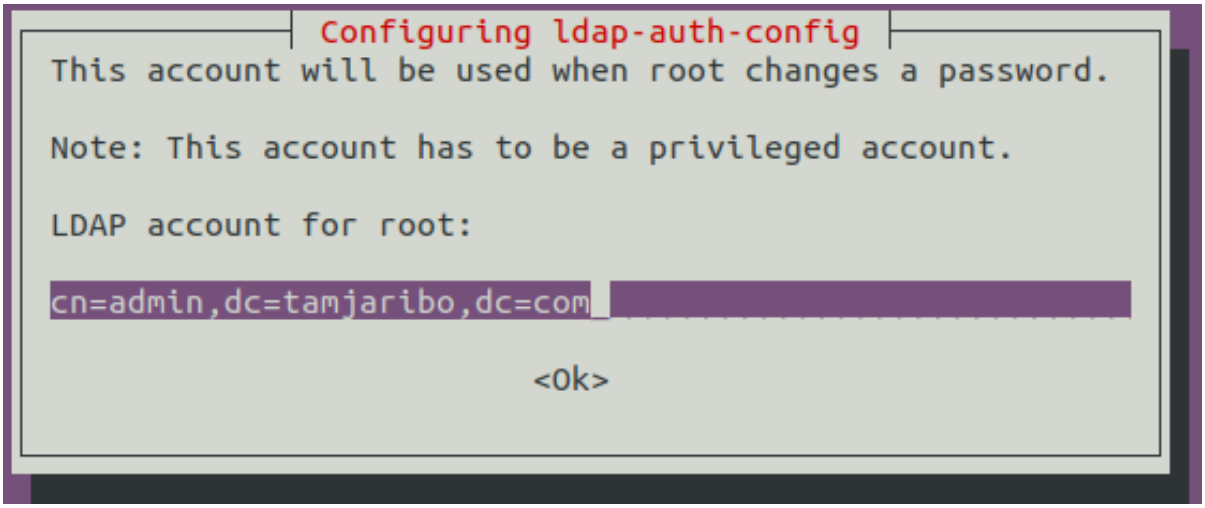
Set LDAP URI:



Set a Distinguished name of the search base:

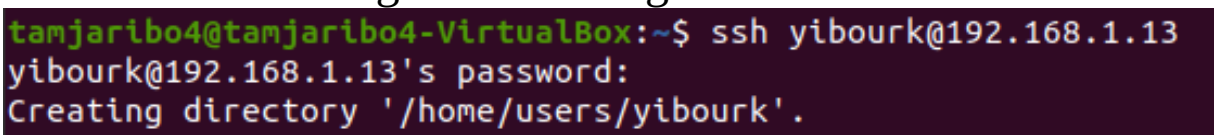


Set LDAP account for root

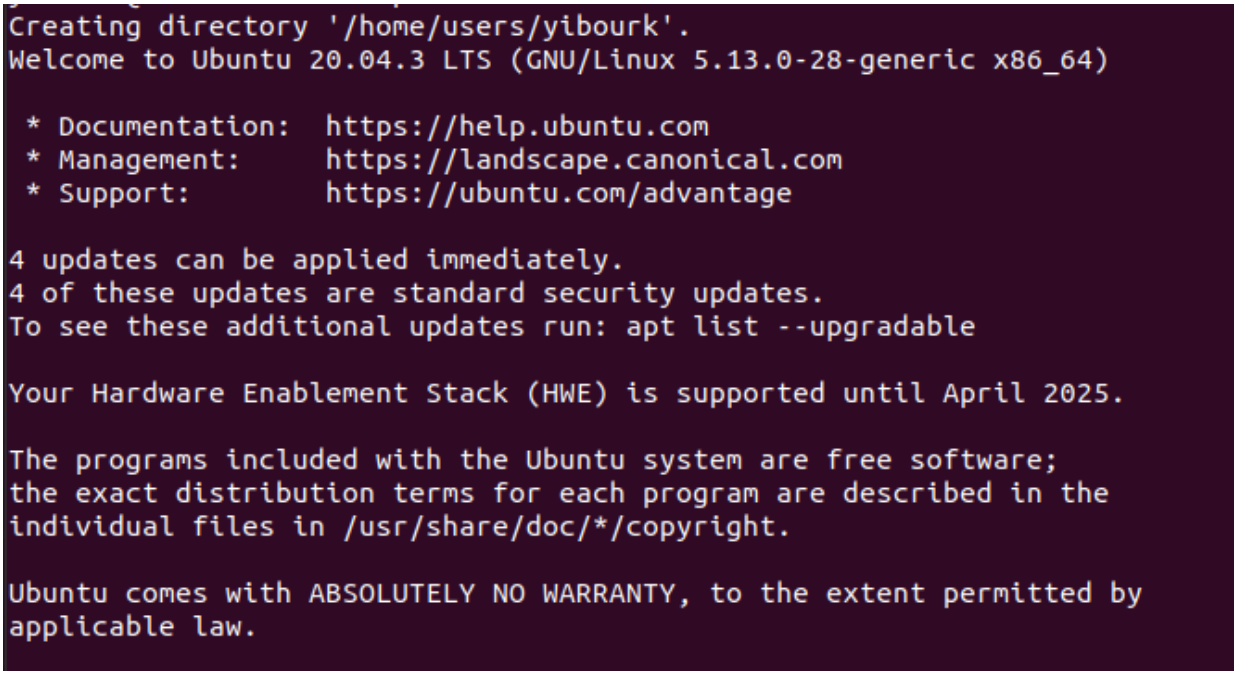


After the installation, edit /etc/nsswitch.confand add ldap authentication to passwd and group lines.

Then restart the system and test by switching to a user account on LDAP using the following:



Result:



We are now connected successfully to the user created in the LDAP server

4. Conclusion

In this project we have used some of the protocols, that we have studied since the beginning of the semester, DHCP, DNS, HTTP, MAIL, ACTIVE DIRECTORY
That is the importance of networking, creating a whole basis of a company (network, websites, secured websites, database, communcationg with other branches of the same company and share data between them)
As we have seen the effeciency of virtual and logical design of a network, and how the various software, hardware and protocols work together to transmit data. When it comes to enterprise networking, every organization has different needs. And besides the concept of creating a network, this project has thaught us, how team work should be, and how we collaborate between each other, in order to complete the given tasks.