CYBERSECURITY SEMINARS Google.org, 1st Cycle

Activity 2K: Linux Machine Communication

Full Name: Klajdi Cami

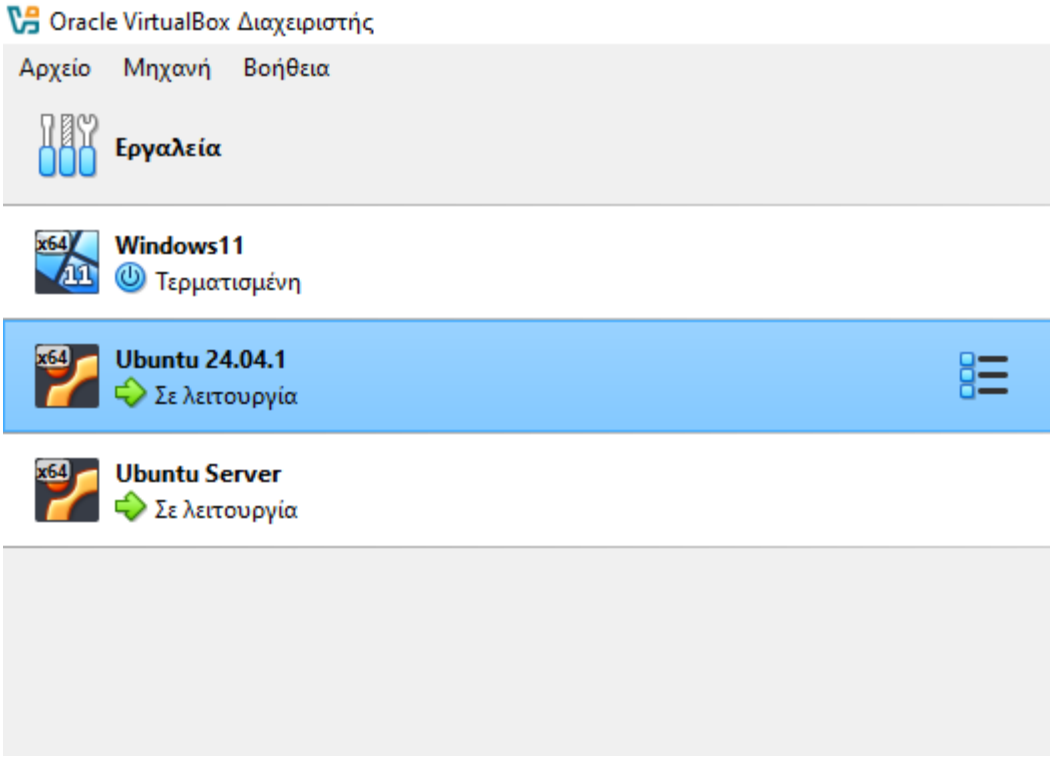
Part A: Machine Communication

Communication Settings

[1] In this part, you will properly configure the two machines to communicate with each other. Take a screenshot for every step.

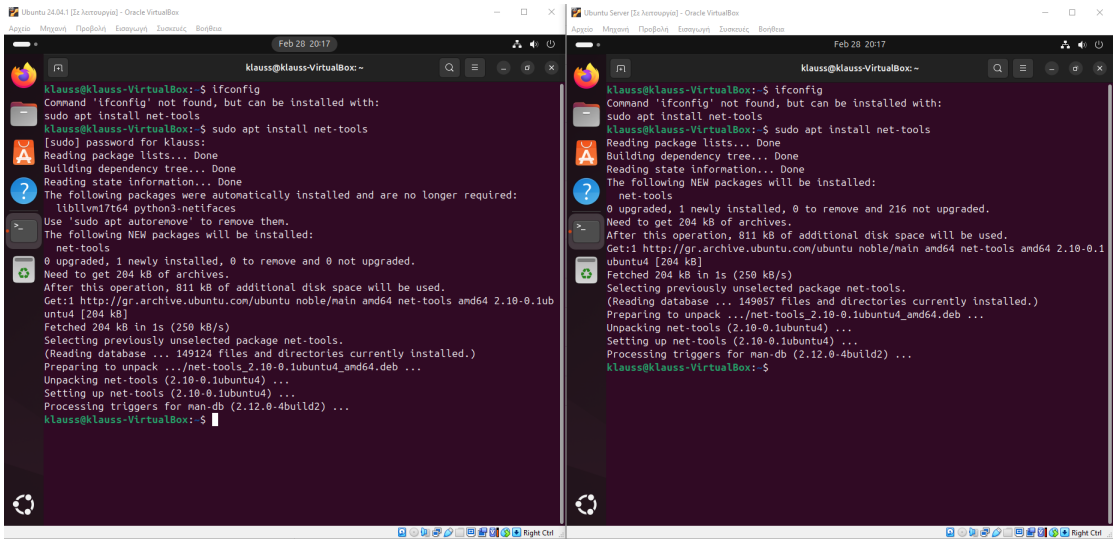
Steps:

First, having installed the second Virtual Machine **Linux-Ubuntu** named **Ubuntu Server**:

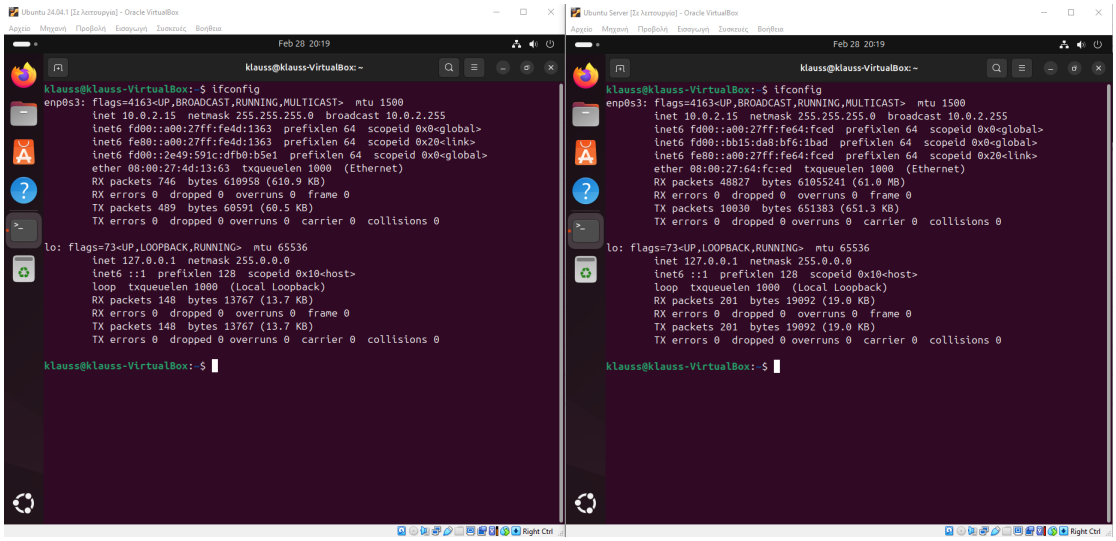


Let’s open a terminal on both machines, check, and install **net-tools** using the following command:

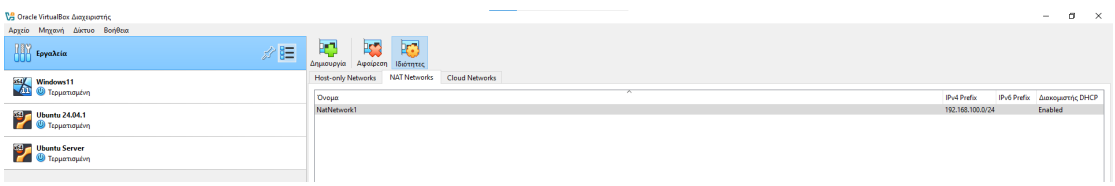
sudo apt install net-tools



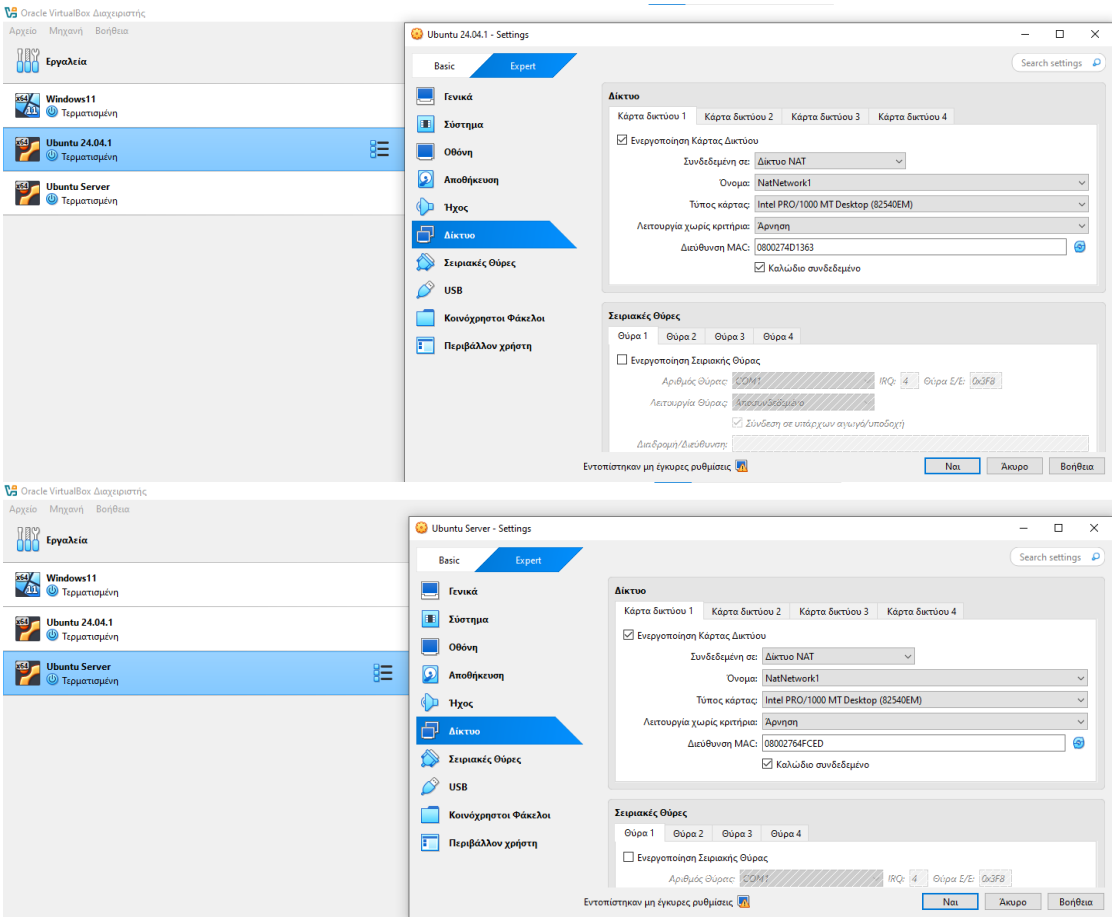
We run the **ifconfig** command again and observe that both virtual machines have the same IP address, meaning they cannot communicate with each other.



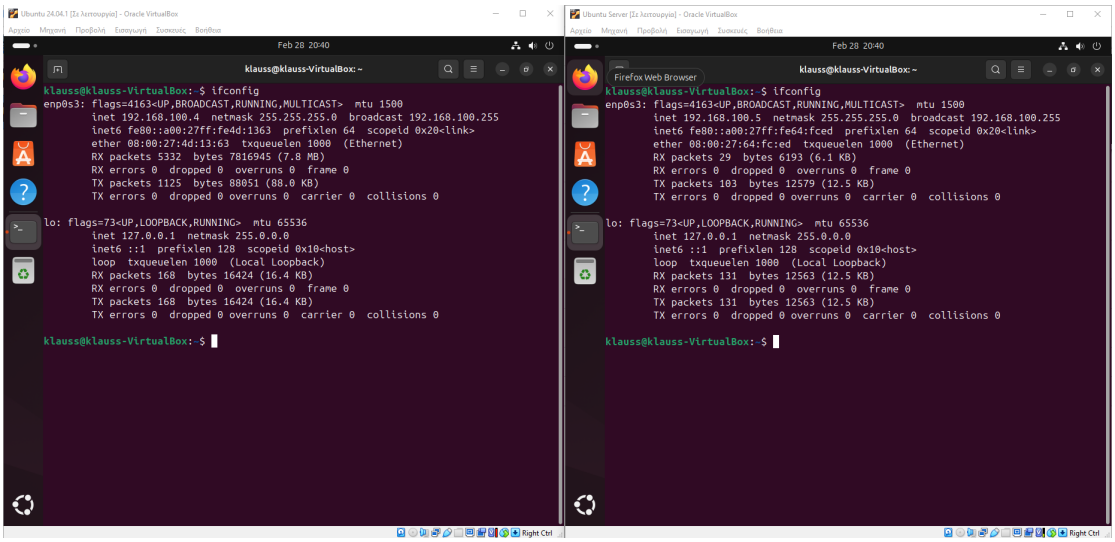
We shut down both virtual machines and navigate to **Tools -> Network -> Nat Networks**, where we create a new network.



Then, we add this network to both virtual machines.

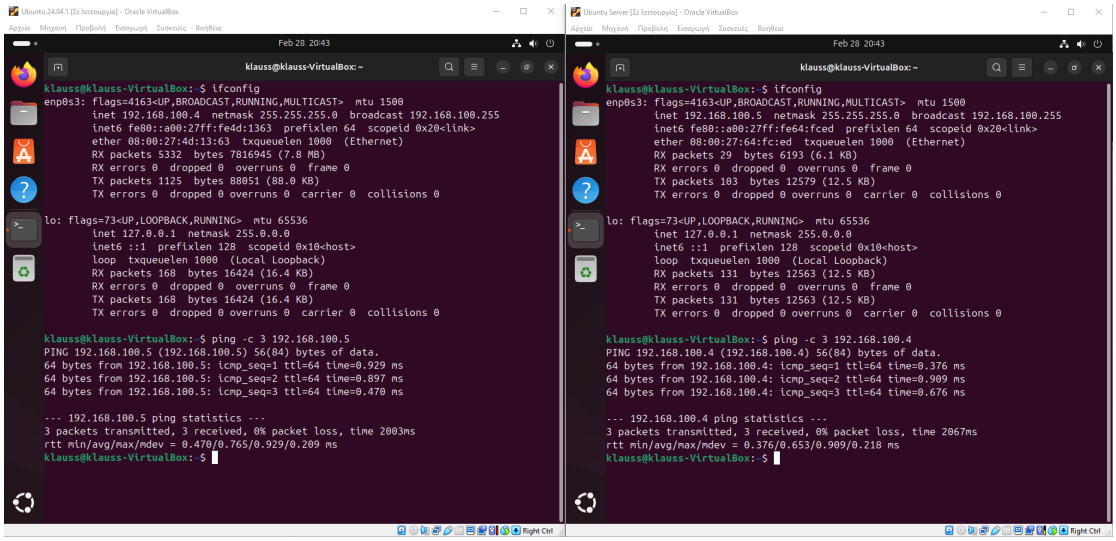


We restart the machines and run the **ifconfig** command again in both terminals. We observe that they have been assigned different IP addresses.



We confirm the connectivity between the two virtual machines by executing the following command:

ping –c 3 <IP_ADDRESS>:



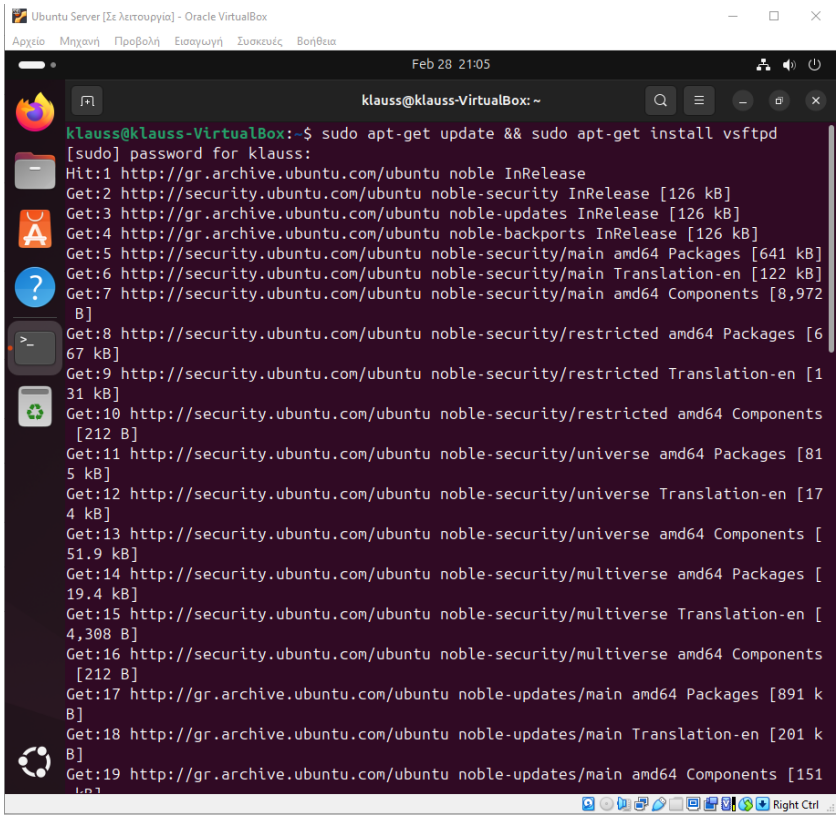
Part B: Setting Up FTP/SCP Servers

[1] In this part, you will configure the processes to enable machine-to-machine communication using **FTP** and **SCP**.

Steps:

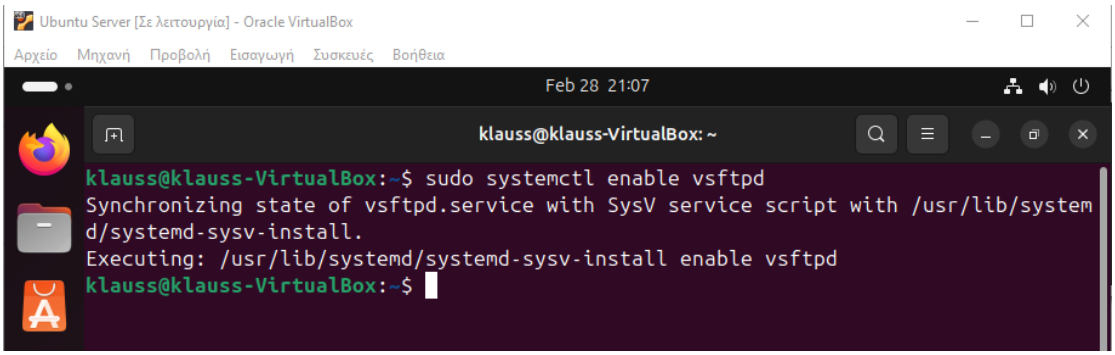
First, we install and update the **vsftpd** application on the **Ubuntu Server** (the virtual machine that will function as the server) using the following command:

sudo apt-get update && sudo apt-get install vsftpd



We activate the service using the following command:

sudo systemctl enable vsftpd

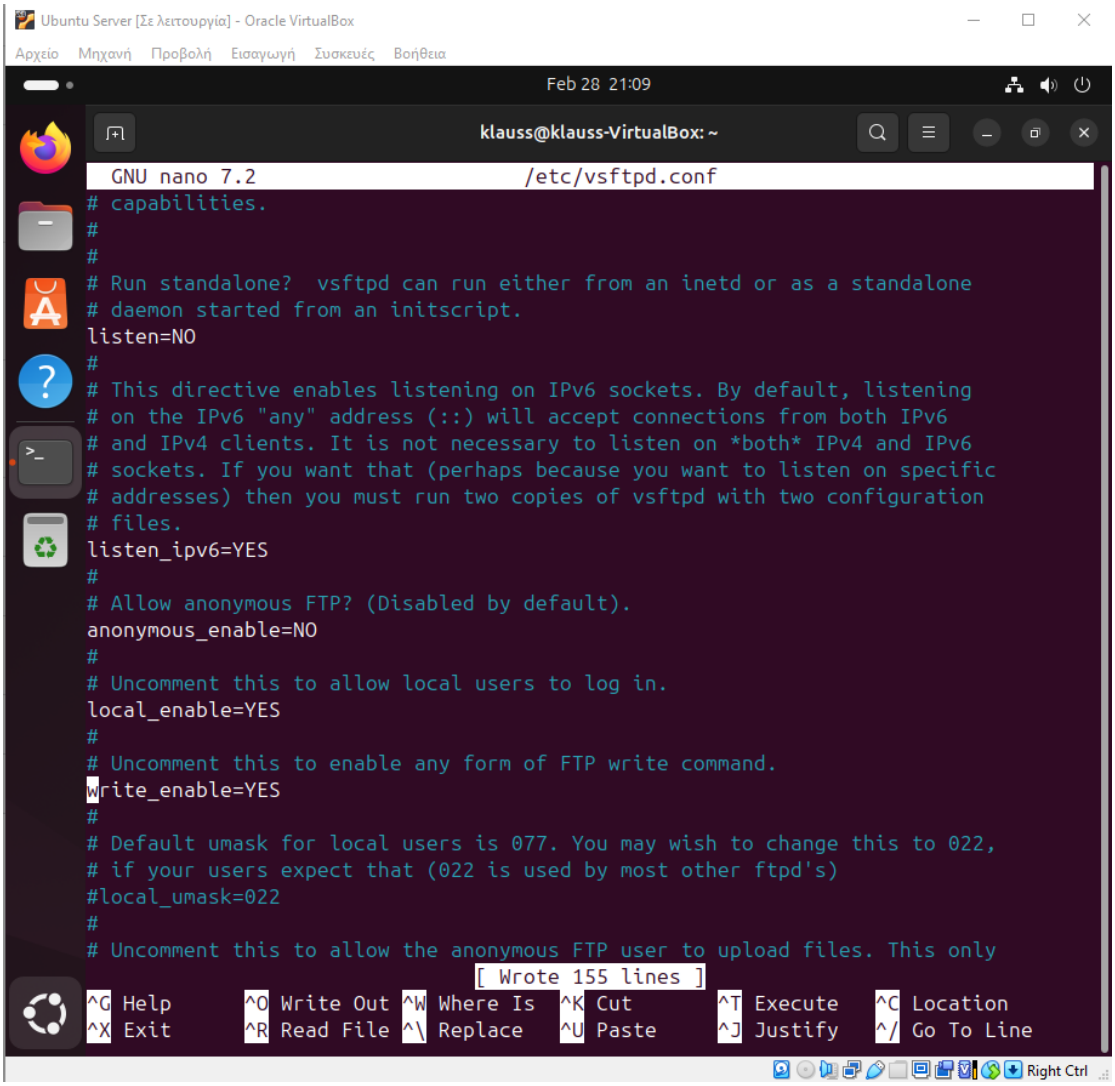


We modify the **vsftpd** service settings by running the following command:

sudo pico /etc/vsftpd.conf

Then, in the editor, we remove the comment (#) from the following line:

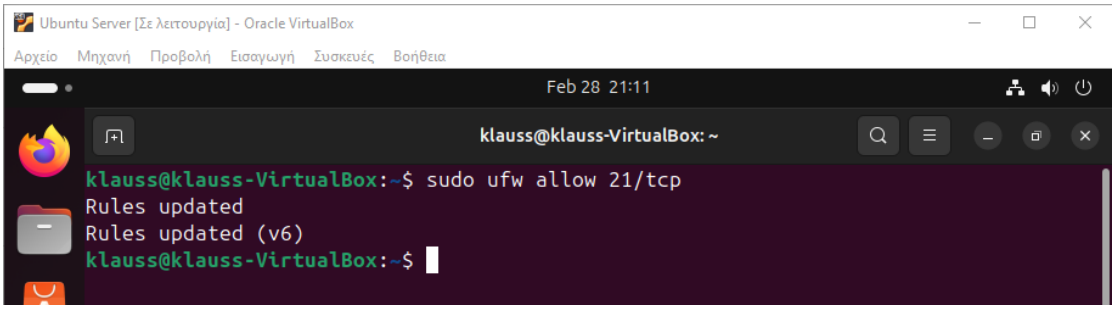
write_enable=YES



We allow access to the **FTP Server** by executing the following command:

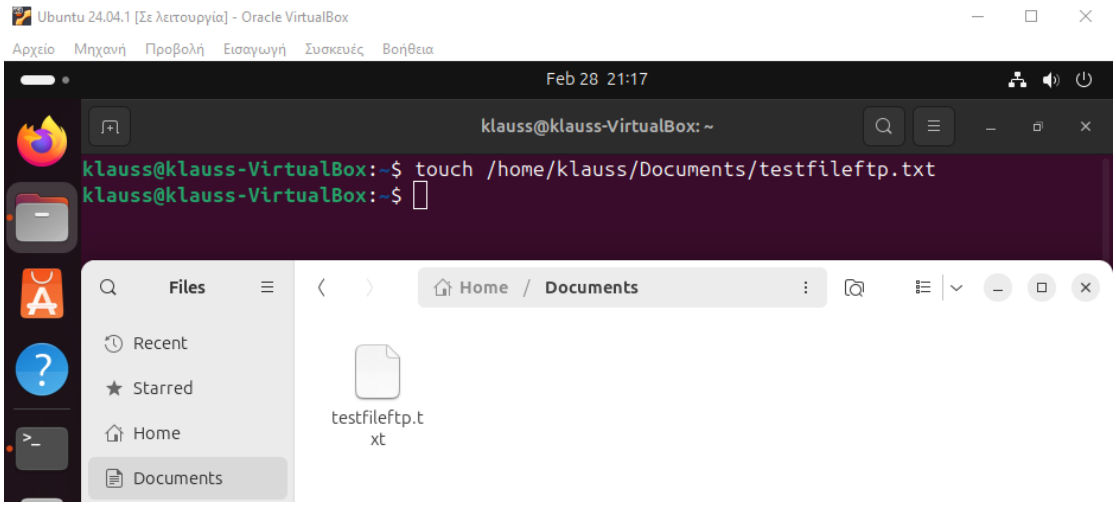
sudo ufw allow 21/tcp

Then, we observe that the access rules related to the server have been updated.



We create a **.txt** file on the **client machine** and place it in the following directory:

home/<user_name directory>/Documents

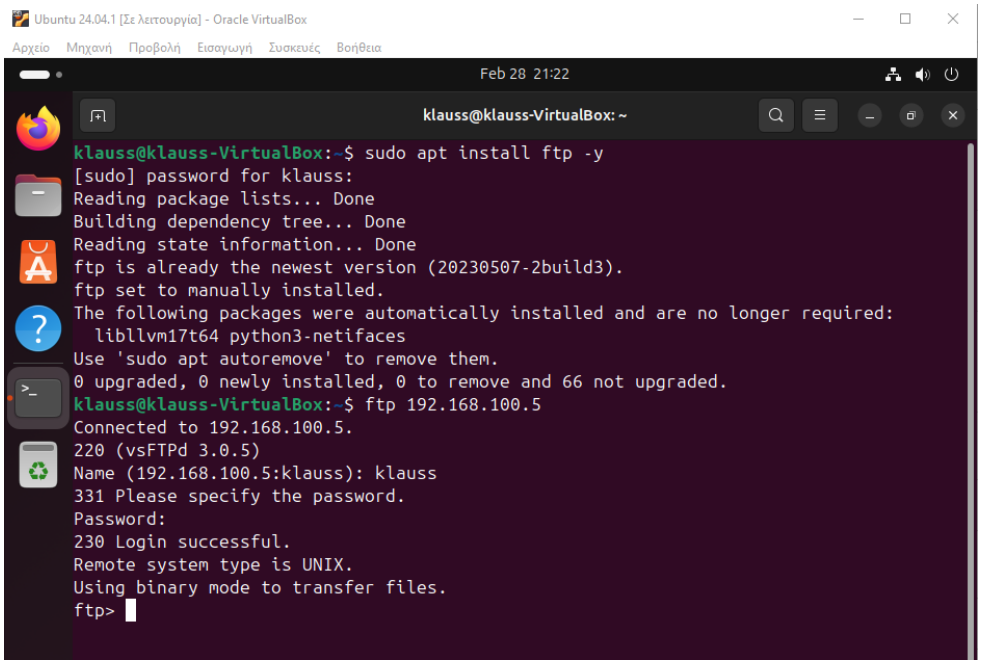


On the **Linux machine acting as the FTP client**, we install the FTP client by executing the following command:

sudo apt install ftp –y

To connect to the **FTP Server** that we set up earlier, we run the following command from the **second Linux machine (client):**

ftp <SERVER'S IP>

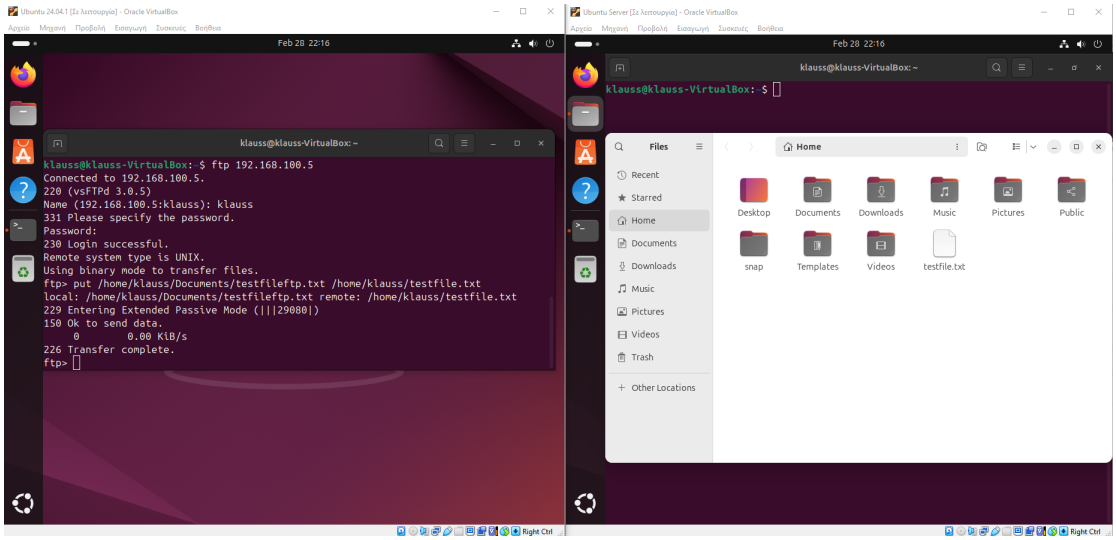


To **upload files** from the client to the server, follow these steps:

Run the following command:

put /home/<user_name directory>/Documents/<filename> /home/<user_name directory>/<filename>

Now, the file **testfileftp.txt** is successfully transferred to the machine acting as the **FTP Server**.



Activity 1:

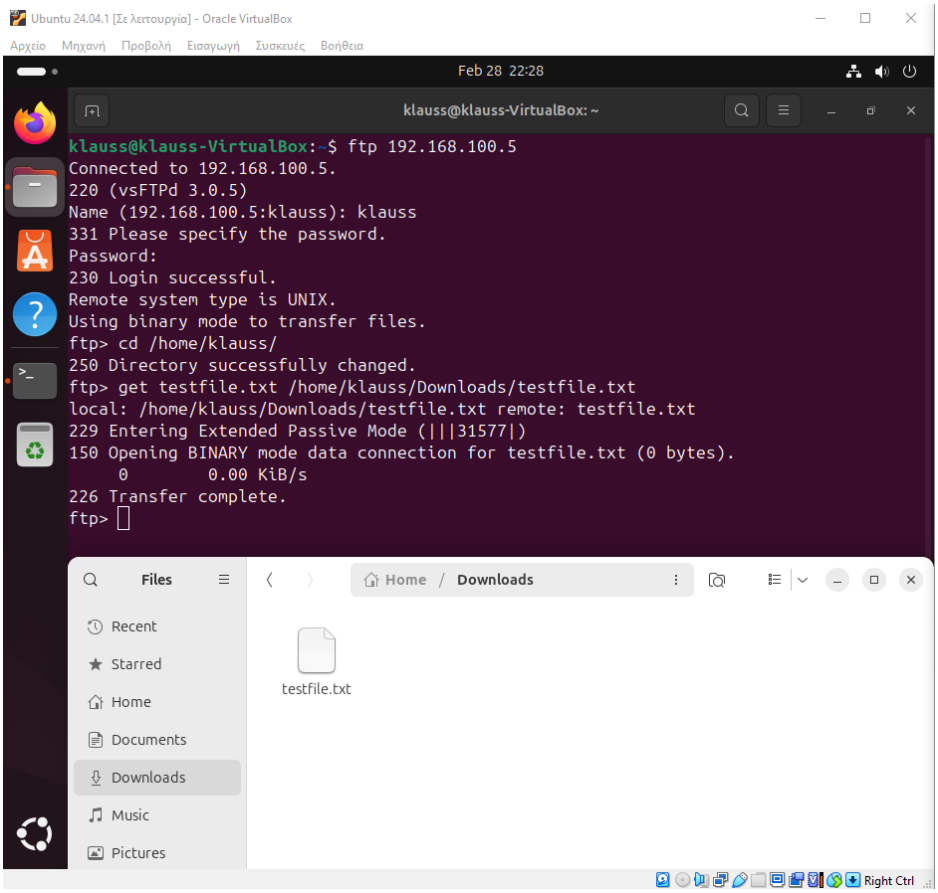
The file created in the previous step, located in **/home/<user_name>**, should be downloaded by the **client** and placed in the directory **/home/<user_name>/Downloads**.

Answer:

To achieve this, we go to the **client** machine and, while connected to the **FTP server** (ftp 192.168.100.5), execute the following commands:

cd /home/klauss/

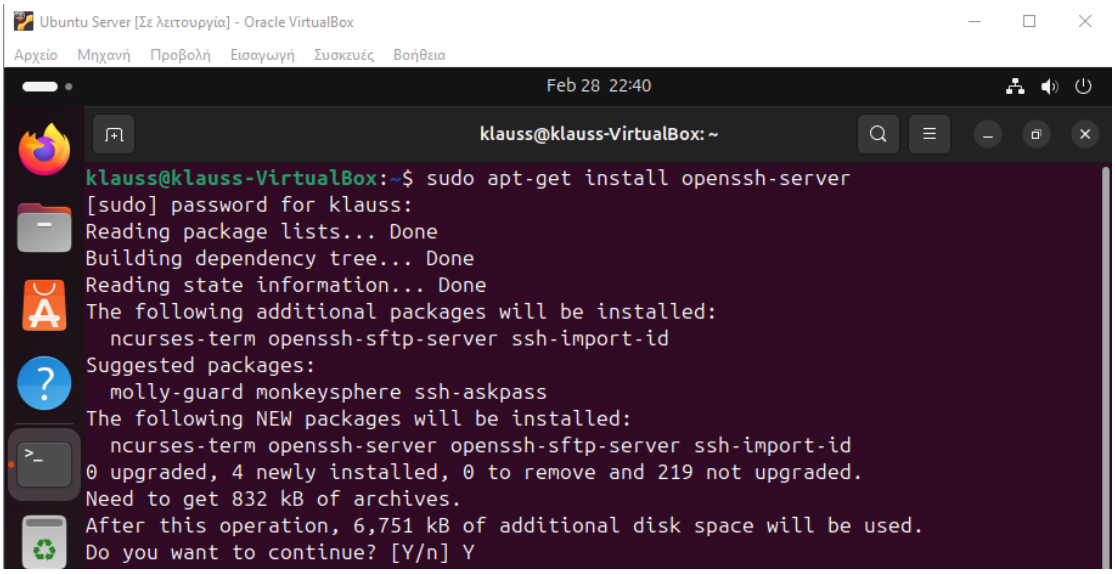
get testfile.txt /home/klauss/Downloads/testfile.txt



File Transfer with SCP

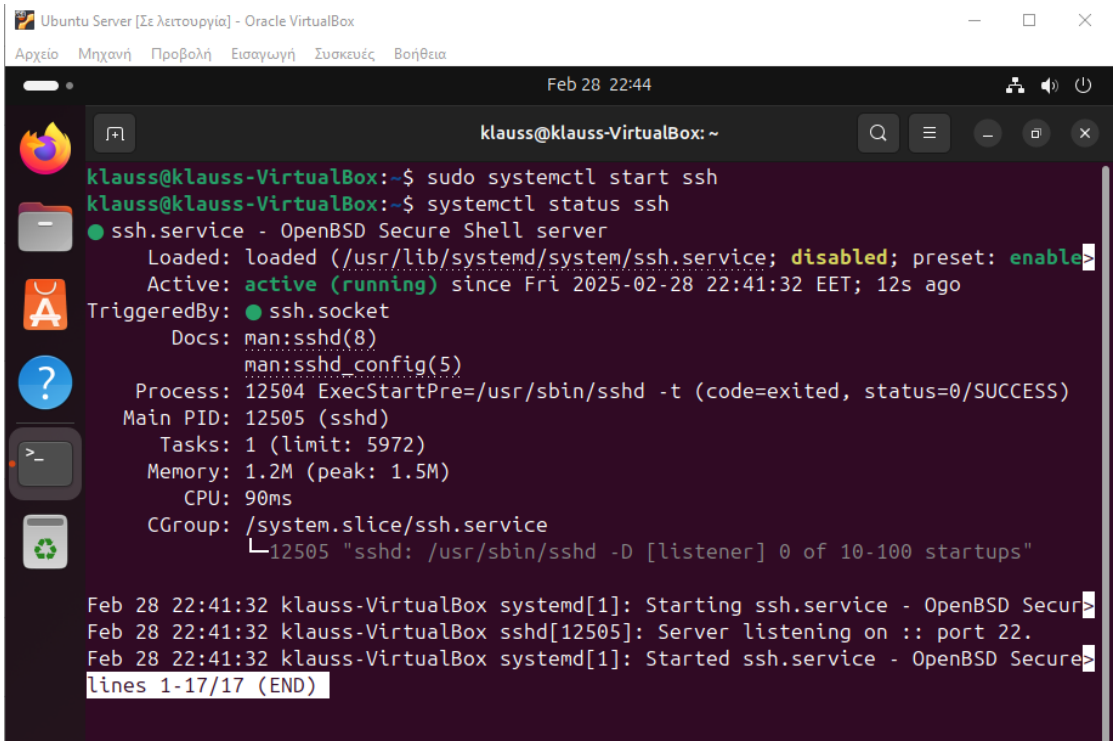
First, we install and update the **OpenSSH server** on the **Ubuntu Server** by executing the following command:

sudo apt-get install openssh-server



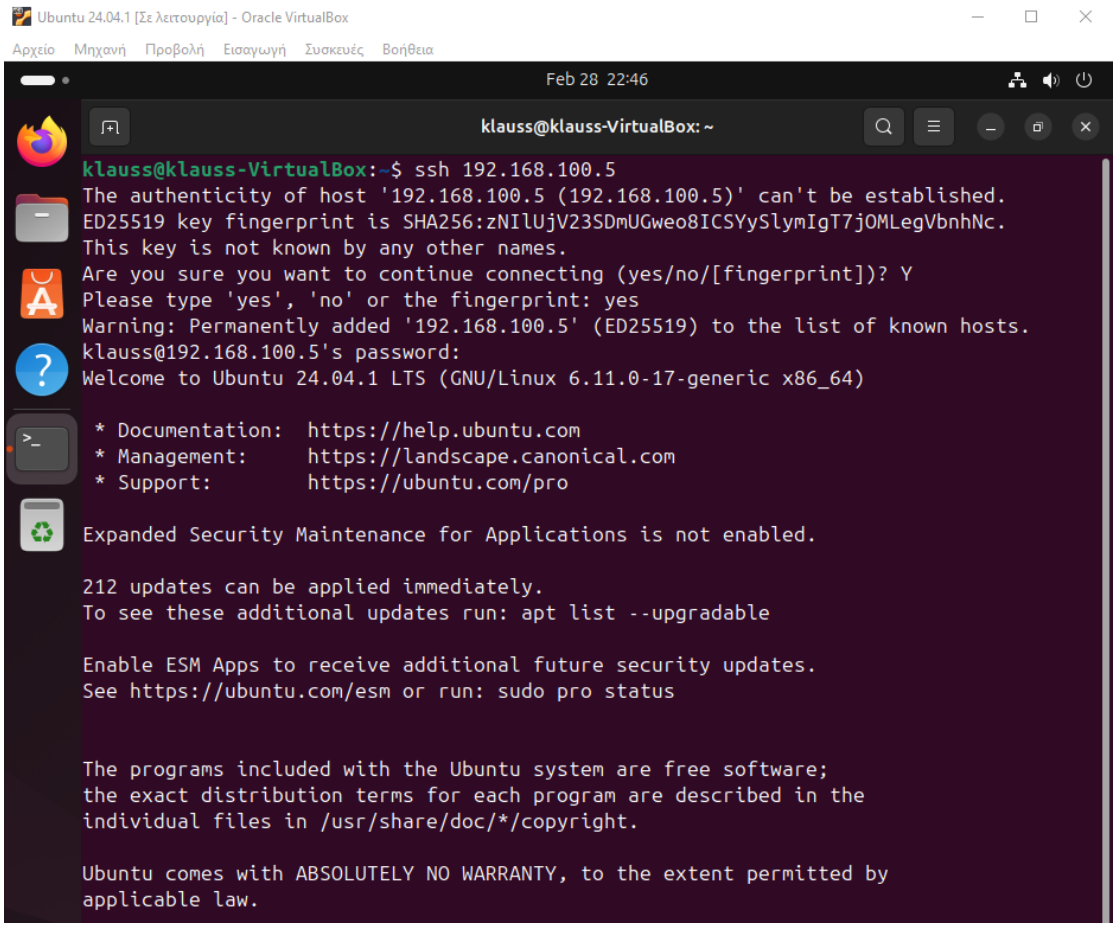
We activate and check if the **OpenSSH service** is running by executing the following commands:

sudo systemctl start ssh
systemctl status ssh

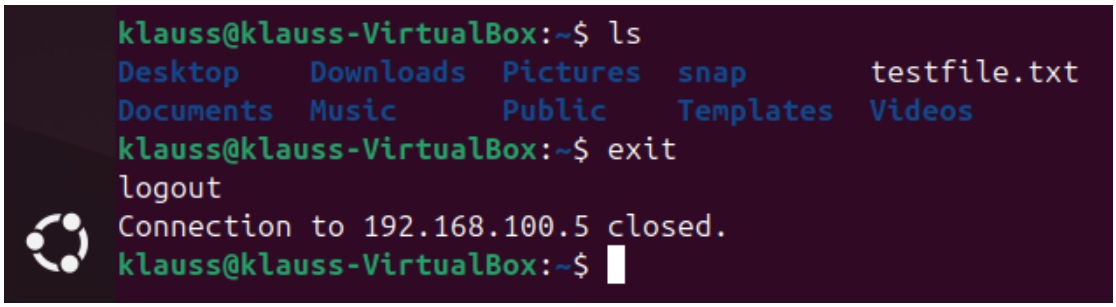


We connect to the **server via SSH** from the second **Linux machine (client)** by executing the following command:

ssh <SERVER'S IP>



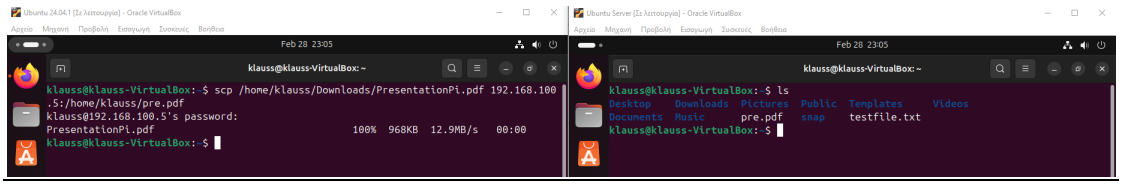
On the **client side**, we observe that we are now inside the **server's directory**. The file **testfile.txt** is **not present** on the **client machine**. We can disconnect from the **remote server** by executing the following command: exit



To send the file **"PresentationPi.pdf"** to the **server**, we execute the following command on the **client machine**:

scp <source_path> <Server's IP>:<destination_path>

To confirm that the file has successfully arrived on the **server**, we switch to the **server's terminal** and run: **ls**



Activity 2:

Download the file **"CyberSecurity Economics for Emerging Markets"** from **eLearning** to the **server**. Then, from the **client side**, download the file using **SCP**.

Answer:

After downloading the file to the **server**, on the **client side**, we execute the following command:
scp <username>@<Server's IP>:/home/<user_name directory>/Downloads/CyberEcoForEmeMar.pdf /home/<local_user>/

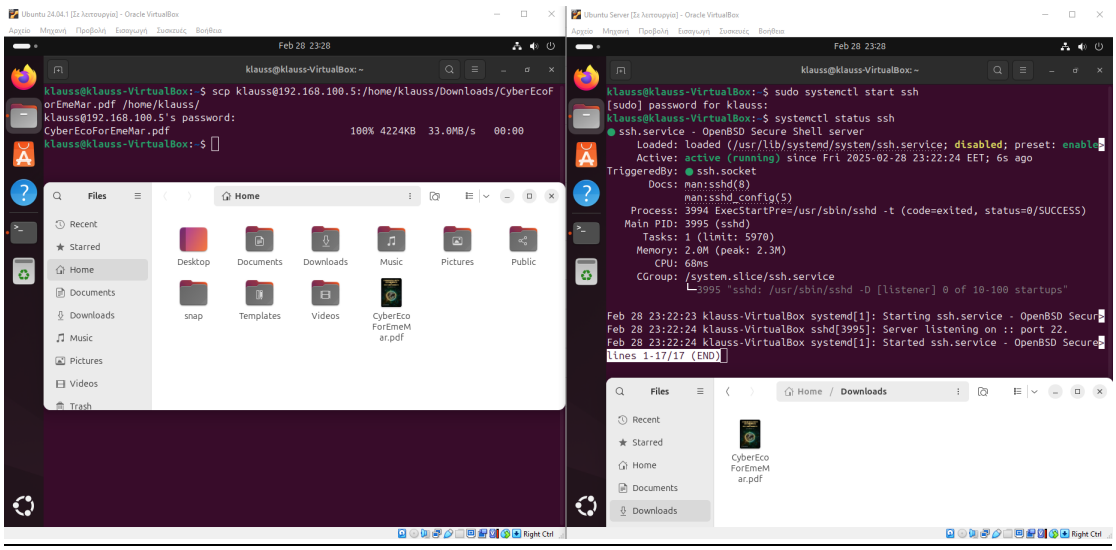
Where:

• **<username>** → The username on the **server** (e.g., user).

• **<Server's IP>** → The **IP address** of the **server**.

• **/home/<user_name>/Downloads/CyberEcoForEmeMar.pdf** → The **file path** on the **server**.

• **/home/<local_user>/** → The **local folder** where the file will be saved on the **client**.



[2] Encrypt a Directory Using PGP

Step 1: Create a Directory

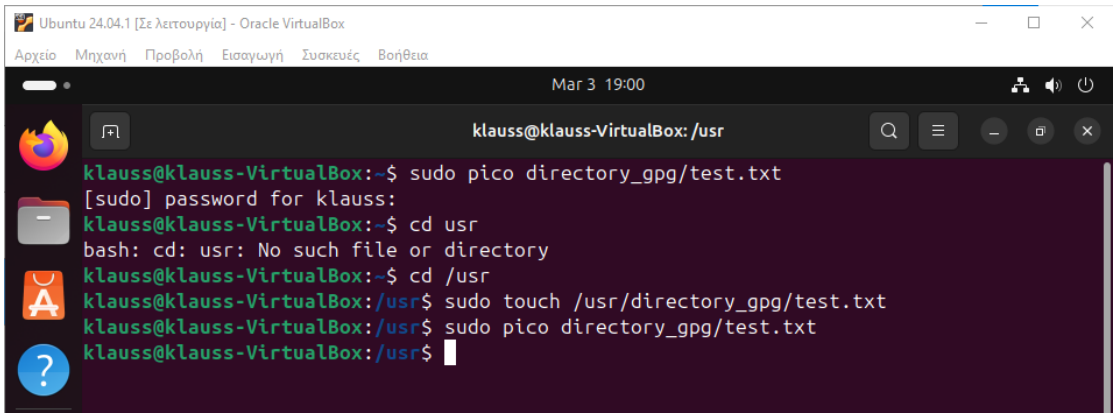
First, we create a directory named **directory_gpg** as the **root user** in the **/usr/** path using the following command:

sudo mkdir /usr/directory_gpg

Next, we create a **test file** inside **directory_gpg** by executing:

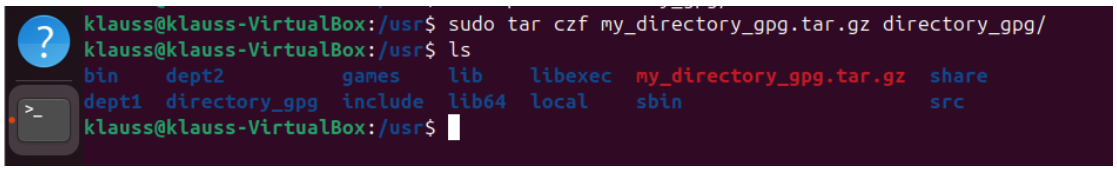
sudo touch /usr/directory_gpg/test.txt

We open the file **test.txt** and write the message **“This is a file to be encrypted!!”**.



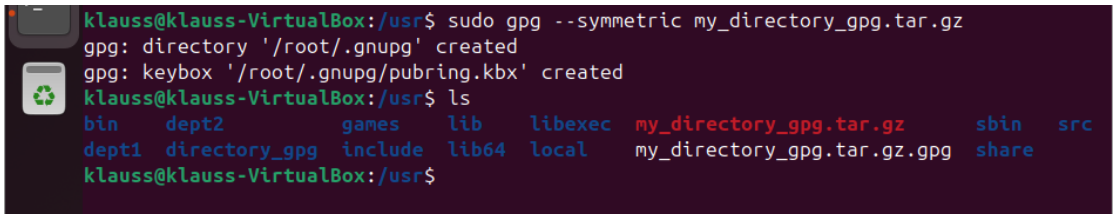
To convert the **directory** into a **file**, we use the following command:

sudo tar czf my_directory_gpg.tar.gz directory_gpg/

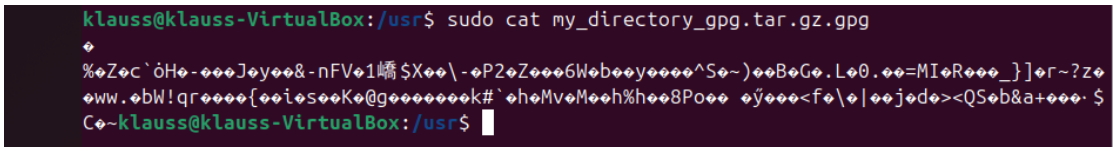


To encrypt the file **my_directory_gpg.tar.gz**, we execute the following command and choose a password (we must remember it because it is the key required for decryption).

sudo gpg - -symmetric my_directory_gpg.tar.gz

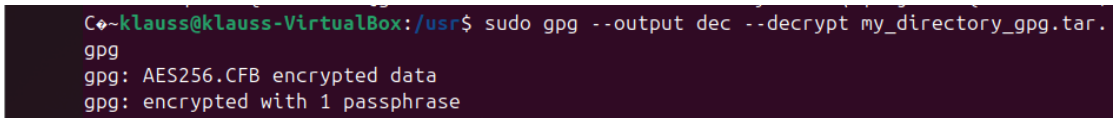


The encrypted file looks something like this…



To decrypt, we execute the following command and enter the key we set in the previous step.

sudo gpg - -output <desired_name_of_decrypted_file> -d my_directory_gpg.tar.gz.gpg



In our case, no password was requested because it was stored on the machine. Additionally, the file is an **archive file**, which means that to view and verify the correct functionality of the **encryption/decryption**, we must first extract it. This is done using the following command:

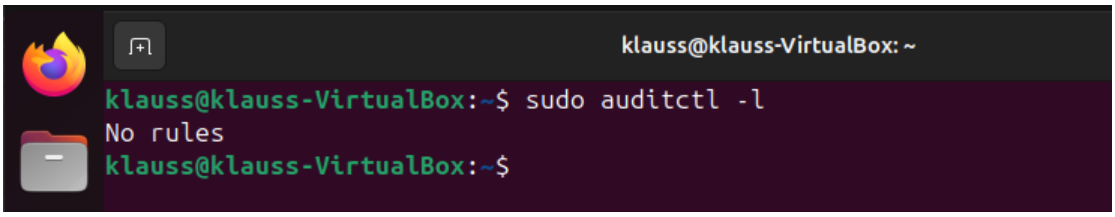
[2] Configure auditd for Event Logging

First, we install the **Auditd** application using the following command:

sudo apt-get install auditd

Then, we check if it was installed correctly and verify that no event logging rules have been set by executing:

sudo auditdctl –l



We create a file using the following command:

touch /tmp/test_auditd.txt

Next, we grant full access permissions to the file, making it freely accessible to all users, by executing:

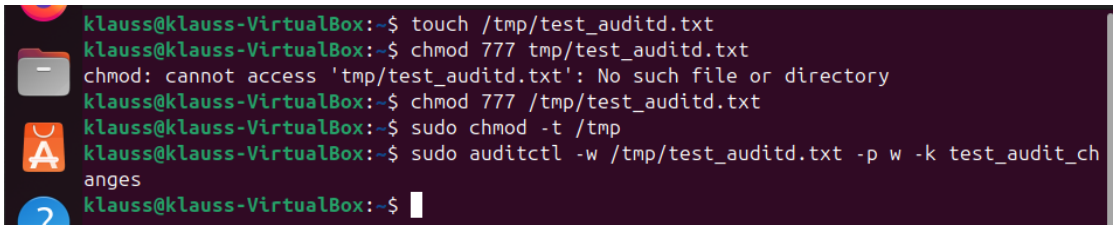
chmod 777 /tmp /test_auditd.txt

We then allow modifications to the **/tmp** directory using the following command:

sudo chmod +t /tmp

Finally, we set up a **monitoring rule** for the file using:

sudo auditctl –w /tmp/test_auditd.txt –p w –k test_audit_changes_changes



Using the **aureport** command, we get an overview of the logs recorded by the **audit** tool on our system.

By executing:

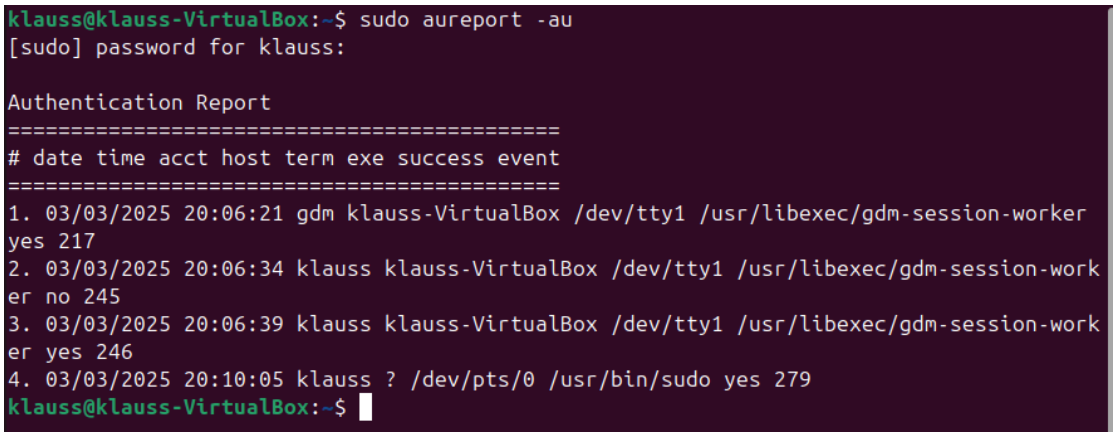
sudo aureport –au

we obtain a report of **all authentication-related events** on our system.

For example, we log out of our account and attempt to log back in, making **one incorrect password attempt** before entering the **correct password**.

Then, we run:

sudo aureport –au and observe that **both login attempts** have been recorded.



Generally, the **rules set in auditd** are deleted after the system shuts down. To make these **rules permanent**, we need to **modify** the following configuration file:

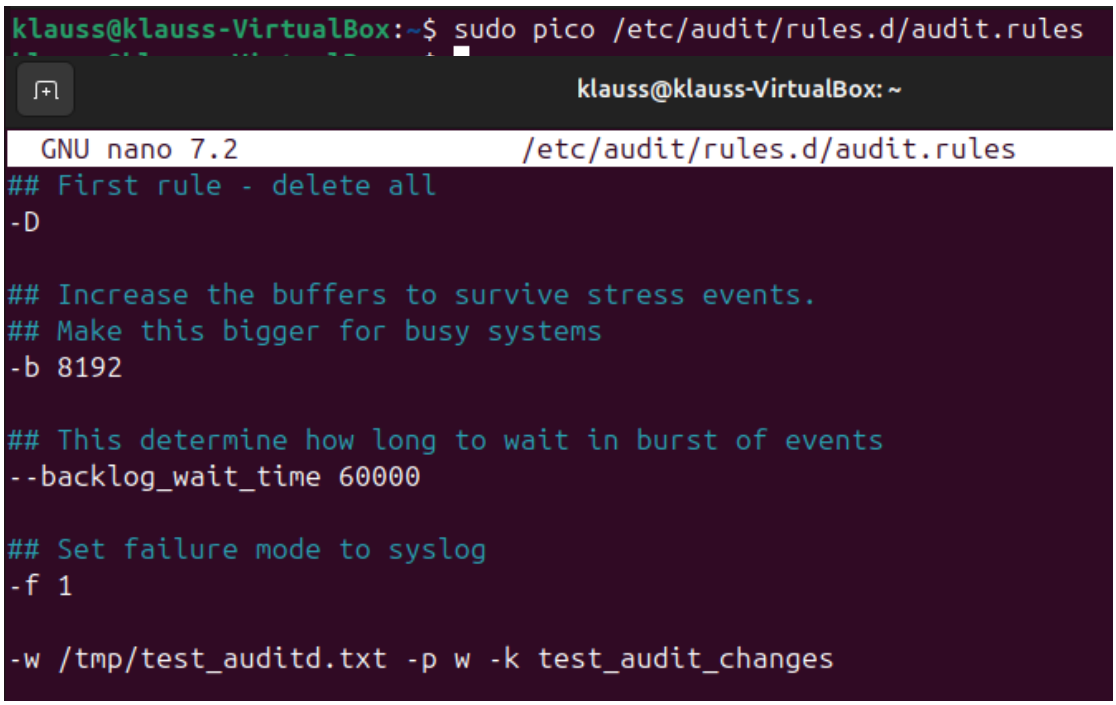
/etc/audit/rules.d/audit.rules

We execute the command:

sudo pico /etc/audit/rules.d/audit.rules

and add the rule we want to **persist** at the end of the file:

-w /tmp/test_auditd.txt -p w -k test_audit_changes



Part C: Creating a Backup

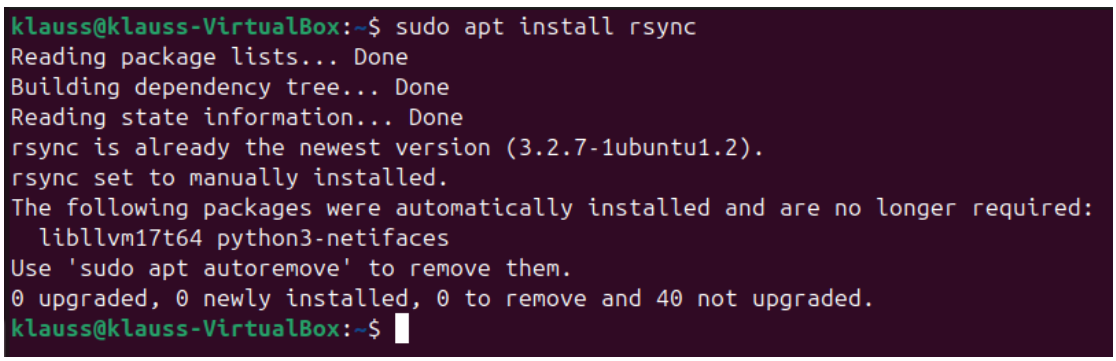
[1] Backup for Immediate Recovery

You will define a **directory** and use **Rsync** to back it up between the two machines for immediate recovery in case of an attack or data loss.

Answer:

First, we install **Rsync** (a fast and flexible command-line tool for synchronizing files and directories between two locations. It ensures fast file transfer by only copying the differences between the source and destination) using the following command:

sudo apt install rsync



Next, we create the directory that we want to **back up** on the **Desktop** and name it **"Vasi"** by executing the following command:

```
klauss@klauss-VirtualBox:~$ cd Desktop
klauss@klauss-VirtualBox:~/Desktop$ mkdir Vasi
klauss@klauss-VirtualBox:~/Desktop$ ls
Vasi
klauss@klauss-VirtualBox:~/Desktop$
```

Next, we create **10 text files** inside the **"Vasi"** directory by executing the following command:

touch /Vasi/file{1..10}.txt

To **confirm** that the files have been created, we list the contents of the directory:

ls

```
klauss@klauss-VirtualBox:~/Desktop$ touch Vasi/file{1..10}.txt
klauss@klauss-VirtualBox:~/Desktop$ cd Vasi/
klauss@klauss-VirtualBox:~/Desktop/Vasi$ ls
file10.txt  file2.txt  file4.txt  file6.txt  file8.txt
file1.txt   file3.txt  file5.txt  file7.txt  file9.txt
klauss@klauss-VirtualBox:~/Desktop/Vasi$
```

We create a second directory **"Vasi_cp"** and synchronize/copy the **"Vasi"** directory to **"Vasi_cp"** by executing the following command. Then, we confirm the result.

```
klauss@klauss-VirtualBox:~/Desktop$ mkdir Vasi_cp
klauss@klauss-VirtualBox:~/Desktop$ rsync -a Vasi Vasi_cp
klauss@klauss-VirtualBox:~/Desktop$ ls Vasi_cp/
Vasi
klauss@klauss-VirtualBox:~/Desktop$ ls Vasi_cp/Vasi/
file10.txt  file2.txt  file4.txt  file6.txt  file8.txt
file1.txt   file3.txt  file5.txt  file7.txt  file9.txt
klauss@klauss-VirtualBox:~/Desktop$
```

The backup can be set to run automatically at the frequency we desire. We add **5 .pdf files** to the **Vasi** directory by executing the following command:

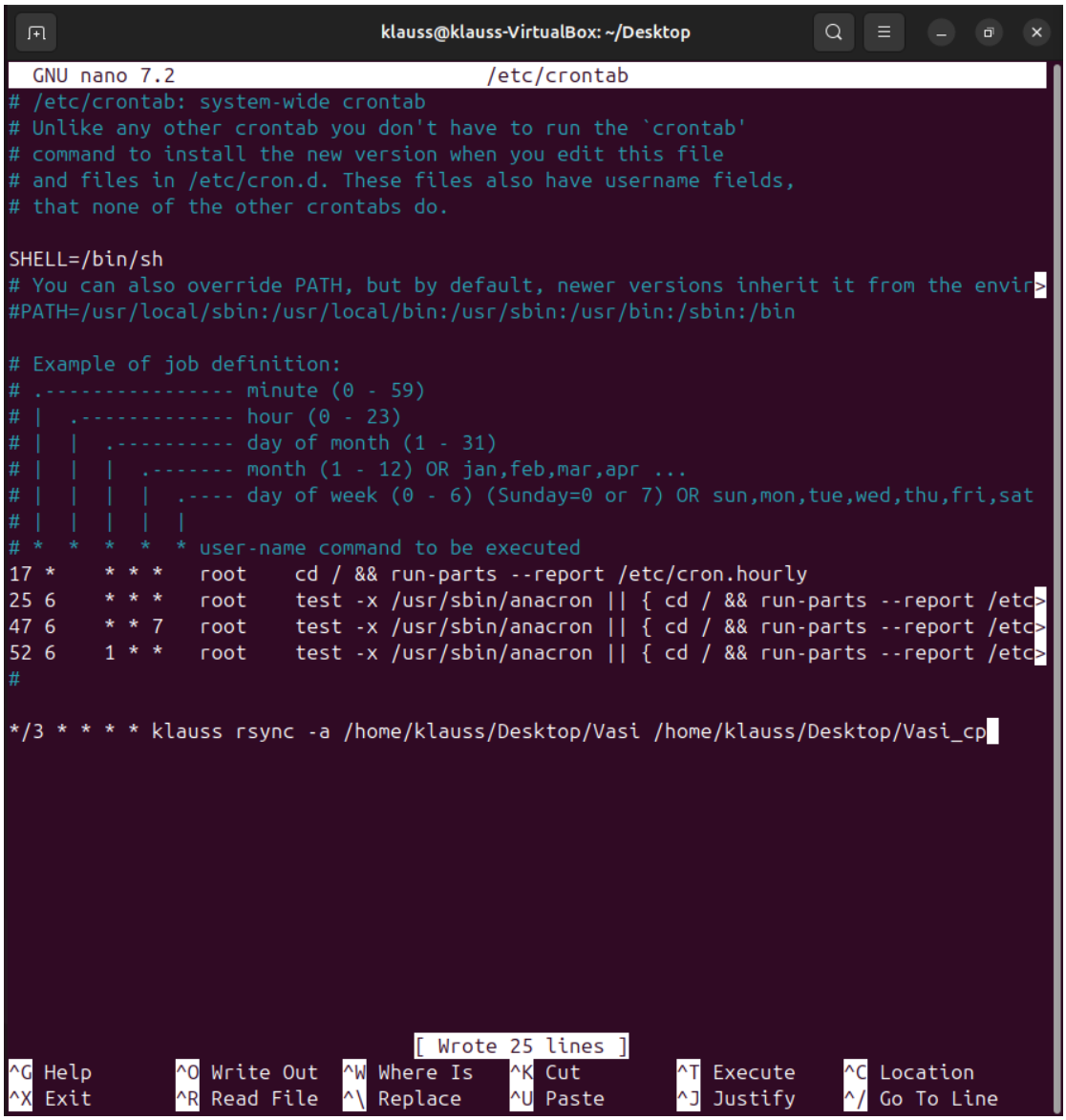
touch Vasi/file{1..5}.pdf

```
klauss@klauss-VirtualBox:~/Desktop$ touch Vasi/file{1..5}.pdf
klauss@klauss-VirtualBox:~/Desktop$ cd Vasi/
klauss@klauss-VirtualBox:~/Desktop/Vasi$ ls
file10.txt  file2.pdf  file3.txt  file5.pdf  file7.txt
file1.pdf   file2.txt  file4.pdf  file5.txt  file8.txt
file1.txt   file3.pdf  file4.txt  file6.txt  file9.txt
klauss@klauss-VirtualBox:~/Desktop/Vasi$
```

To set the backup frequency, we need to modify the configuration file by executing the following command:

sudo pico /etc/crontab

With the rule shown at the end of the file, the backup is performed every **3 minutes**.



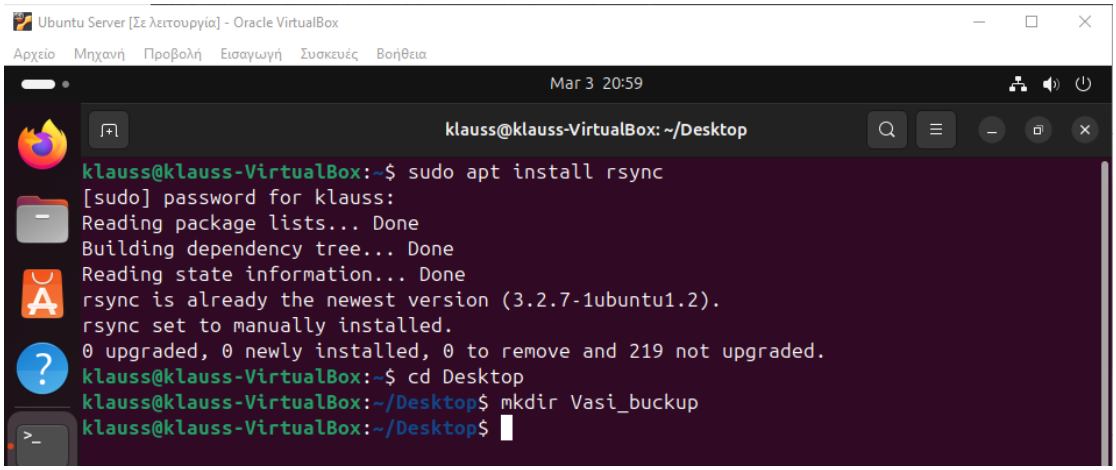
The backup can be set to run automatically at the frequency we desire for specific file types. In this scenario, we want to back up only the **.pdf** files located in the **Vasi** directory.

```
klauss@klauss-VirtualBox:~$ rsync -av --exclude '*.txt' /home/klauss/Desktop/Vasi/ /hom
e/klauss/Desktop/Vasi_pdf
sending incremental file list
created directory /home/klauss/Desktop/Vasi_pdf
./
file1.pdf
file2.pdf
file3.pdf
file4.pdf
file5.pdf

sent 351 bytes  received 166 bytes  1,034.00 bytes/sec
total size is 0  speedup is 0.00
klauss@klauss-VirtualBox:~$
```

Scenario: We want to keep a backup of the **"Vasi"** directory on the second machine we have set up, which functions as a server.

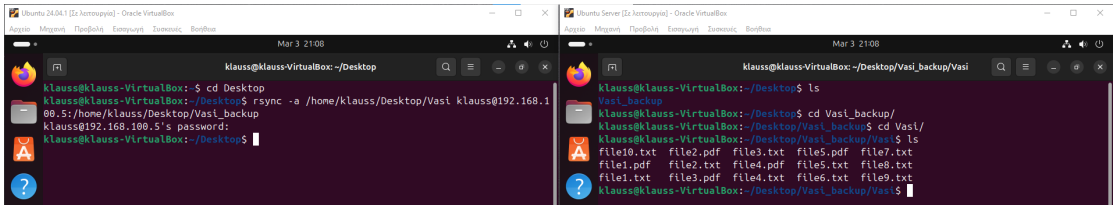
Step 1: On the machine functioning as a server, we create a directory **"Vasi_backup"** on the desktop, where the backup will be stored.



Step 2: From the first machine, which contains the **"Vasi"** directory that we want to back up, we execute the following command:

Rsync -a /home/klauss/Desktop/Vasi
klauss@192.168.100.5:/home/klauss/Desktop/Vasi_backup

And we confirm that the backup was successful by typing the following in the **server's terminal**:


Ubuntu 24.04.1 [Σε λειτουργία] - Oracle VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοηθεια
Mar 3 21:08
klauss@klauss-VirtualBox: ~/Desktop
klauss@klauss-VirtualBox:~$ cd Desktop
klauss@klauss-VirtualBox:~/Desktop$ rsync -a /home/klauss/Desktop/Vasi klauss@192.168.100.5:/home/klauss/Desktop/Vasi_backup
klauss@192.168.100.5's password:
klauss@klauss-VirtualBox:~/Desktop$
Ubuntu Server [Σε λειτουργία] - Oracle VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοηθεια
Mar 3 21:08
klauss@klauss-VirtualBox: ~/Desktop/Vasi_backup/Vasi
klauss@klauss-VirtualBox:~/Desktop$ ls
Vasi_backup
klauss@klauss-VirtualBox:~/Desktop$ cd Vasi_backup/
klauss@klauss-VirtualBox:~/Desktop/Vasi_backup$ cd Vasi/
klauss@klauss-VirtualBox:~/Desktop/Vasi_backup/Vasi$ ls
file10.txt file2.pdf file3.txt file5.pdf file7.txt
file1.pdf file2.txt file4.pdf file5.txt file8.txt
file1.txt file3.pdf file4.txt file6.txt file9.txt
klauss@klauss-VirtualBox:~/Desktop/Vasi_backup/Vasi$