Project: Securing a networked system with PKI and configuring a firewall and IDS

Course: CSE 487
Section: 3
Instructor: Rashedul Amin Tuhin
Senior Lecturer

Submitted By
Ramisa Anjum
ID: 2018-3-60-006
Md. Sagor Hossain
ID: 2019-1-60-049
Md. Asif Imtiyaj Chowdhury
ID: 2019-3-60-115

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
EAST WEST UNIVERSITY
(SUMMER-2022)

Table of Contents

Abstraction

This report explored and provided factors which are involved in this mini project of securing a networked system with PKI and configuring a firewall and IDS.

This mini project is completed under the Linux operating system (Lubuntu), which is a very light distribution of Linux because of the usage of VMware, which helps us complete our mini project in a virtual environment. The report provides multiple steps on how to install Xampp, OpenSSL, and other tools for the generation of self-signed certificates (digitally) and how to install those certificates into the browser.

The report will provide knowledge about why a secure network is necessary and how to get a self-signed certificate (digitally) by the visibility of a padlock on a website. Nowadays, a secure network is very important. A secured network helps us to prevent the dangerous sharing of consumer data such as social security numbers, private health information, and financial information. Without a secured network, there will be many trespassers, and the data will be compromised. That leaked data could be harmful to the targeted public. The key difference between HTTPS and HTTP is that HTTPS is HTTP with encryption, while HTTP is without encryption. Using Xampp, which is a free and open-source cross-platform web server solution stack package to host a website, The use of XCA, which is intended for creating and managing X.509 certificates, certificate requests (CR), RSA and DSA private keys, It can also be said that XCA is a GUI version of OpenSSL,
OpenSSL is a terminal/command line interface and it is used to generate self-signed certificates. It is open to all. This report will provide a proper guideline for this requirement to fulfill the use of OpenSSL to generate self-signed certificates using appropriate commands which are given below in derived steps. This report will also provide guidelines to configure firewalls with ports 53, 80, and 443 (only).

Introduction

These days, we want our data to be safe and not get into the hands of bad guys. Because of that, we also want safety in our digital life and while hosting a website, we also want to ensure the verification that the website is secure to visit. Network security is a set of technologies that protect the usability and integrity of a company's infrastructure by preventing the entry or proliferation within a network of a wide variety of potential threats. PKI, which stands for "Public Key Infrastructure", governs the issuance of digital certificates to protect sensitive data, provide unique digital identities for users, devices, and applications, and secure end-to-end communications. There are many steps to getting a secure and verified website (securing a padlock), such as generating a self-signed certificate to prove that it is a secure website to visit. That can be generated by using XCA with OpenSSL, which is a GUI version of OpenSSL, but OpenSSL is a terminal/command line interface, and this report will provide proper guidelines to execute.

HTTPS is not a separate protocol from HTTP. It is simply using TLS/SSL encryption over the HTTP protocol. HTTPS occurs based upon the transmission of TLS/SSL certificates, which verify that a particular provider is who they say they are. Transport Layer Security (TLS) is an Internet Engineering Task Force (IETF) standard protocol that provides authentication, privacy, and data integrity between two communicating computer applications. HTTPS uses Transport Layer Security (TLS)/SSL protocol to encrypt communication between the client and the server. This protocol uses asymmetric encryption to encrypt those communications, which creates private and public keys to secure the communication. The only difference between these two protocols is that HTTPS uses TLS (SSL) to encrypt normal HTTP requests and responses and to digitally sign those requests and responses. As a result, HTTPS is far more secure than HTTP. A website that uses HTTP has http:// in the URL. In other parts, a website that uses HTTPS has https://.

Tools for Implementation:

VMware allows businesses to run multiple operating system workloads on the one server—thus enabling better resource management. By creating a virtual machine that behaves exactly like an actual computer, a computer inside a computer**, OS(**Linux(lubuntu/linux mint/kali linux)**),** **XAMPP** is a software distribution which provides the Apache web server, MySQL database (actually MariaDB), PHP, and Perl as command-line executables and Apache modules all in one package, **OpenSSL** is a software library for applications that secure communications over computer networks against eavesdropping or need to identify the party at the other end. It is widely used by Internet servers, including the majority of HTTPS websites**)**, and a few extra commands for the remaining configurations.

Demonstrations of Key Components

Firstly, a VMware workstation is needed because a virtual environment will be needed to demonstrate the mini project. VMware Workstation was the first product launched by VMware. It is the most popular software that offers the ability to run multiple instances of the operating system on a single physical personal computer.

Installing VMware Workstation

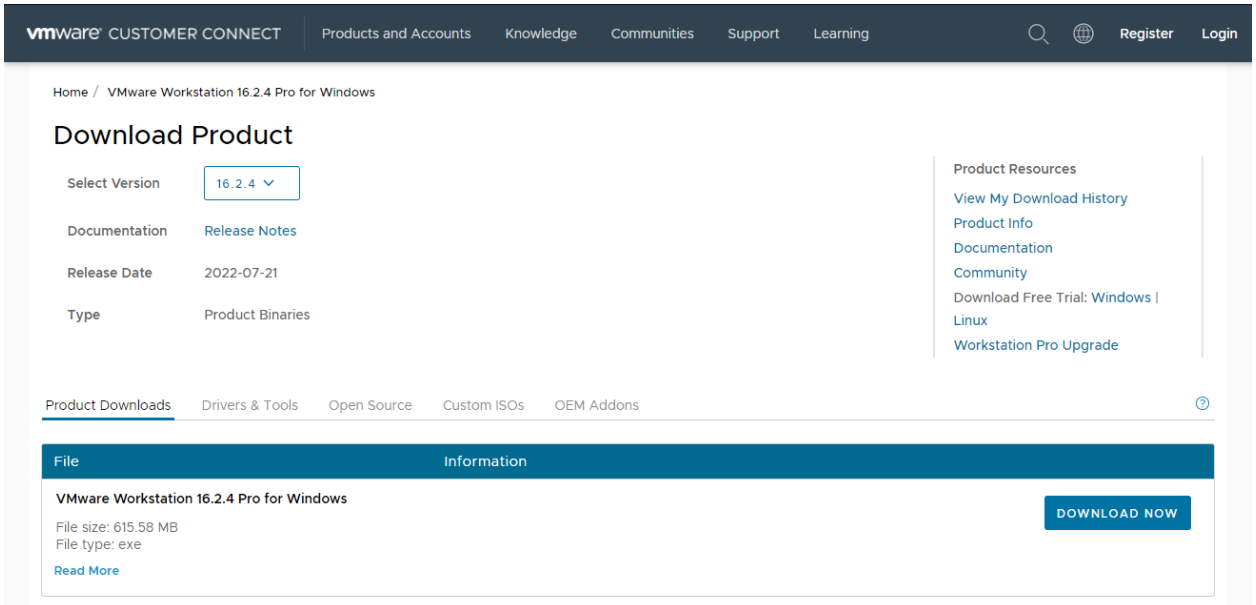
We will use Lubuntu, which is a lightweight OS Linux distribution based on Ubuntu. It provides a fast and easy-to-use user interface. It uses the portable LXDE/LXQT desktop environment, which makes it lightweight as compared to Ubuntu's GNOME desktop. Since it is quite lightweight, it requires less hardware and also provides better efficiency.

Steps to install VMware:

Below are the detailed steps for installing VMware Workstation.

Step 1: To download and install the VMware product, visit the official website of VMware. https://customerconnect.vmware.com/en/downloads/details?downloadGroup=WKST-1624-WIN&productId=1038&rPId=91434

Hover on the Downloads tab. Here you will find various products.

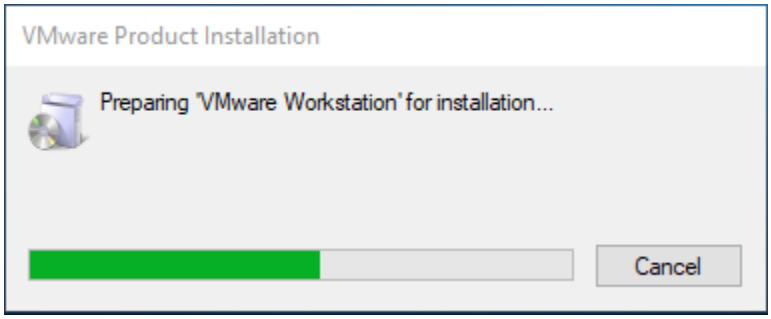


For personal use only, VMware Workstation Pro will be compatible with your machine.

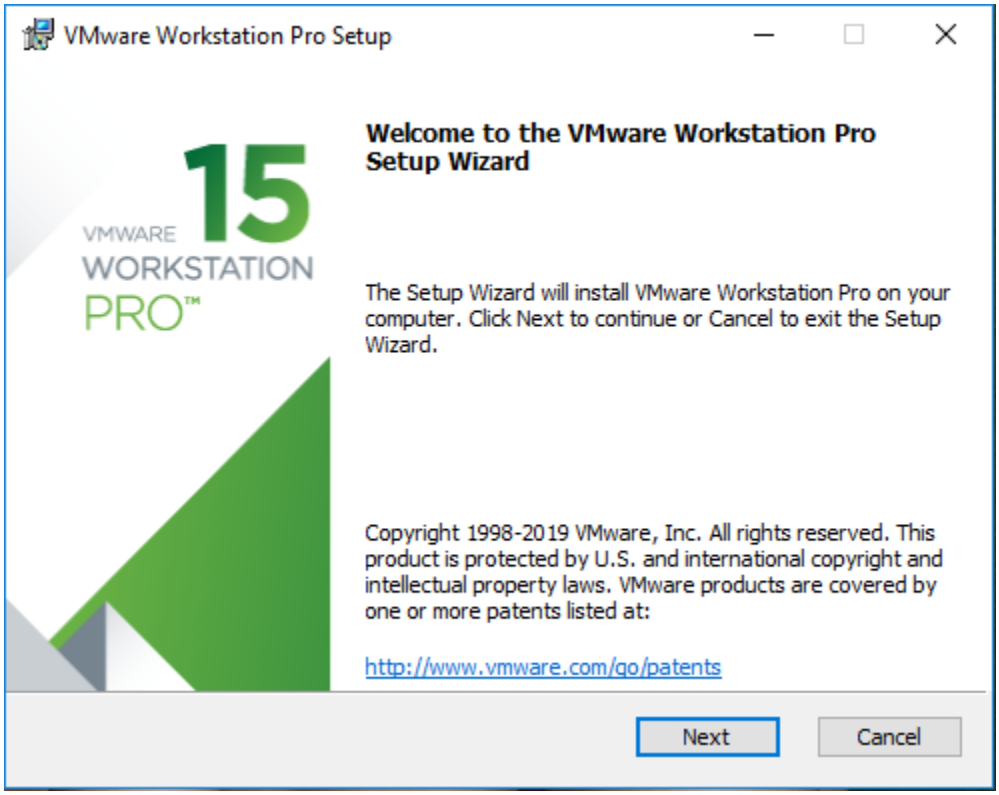
Step 2: Click on "Free Product Trials & Demo" >> Workstation Pro. You will be redirected to the download page. (Similarly, you can select any product that you want to install.)

Click on Download Now according to your operating system. We have chosen Workstation 15 Pro for Windows.
While downloading, make sure you have a proper internet connection as the file may have a large size.

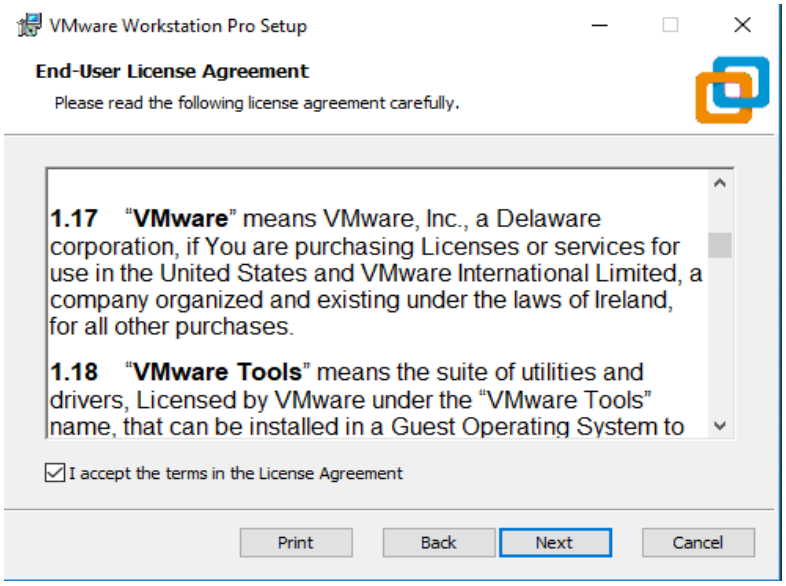
Step 3: Once the download is complete, run the.exe to install VMware Workstation. A popup will appear.



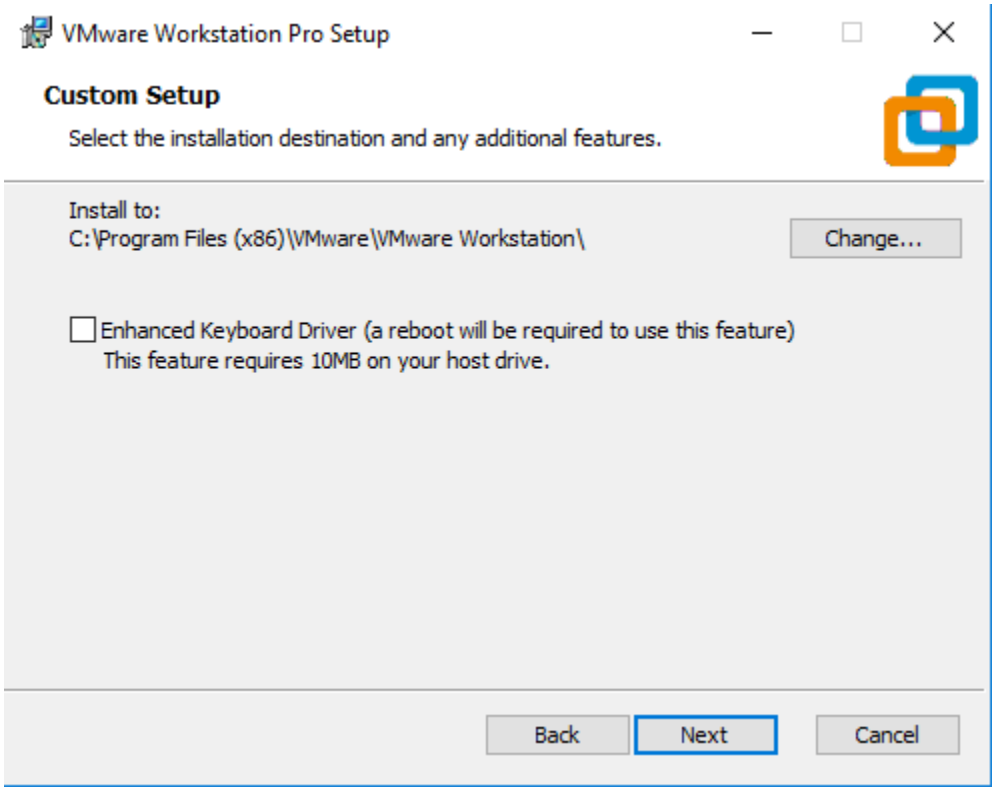
Step 4: Once initialization is completed, click on Next.



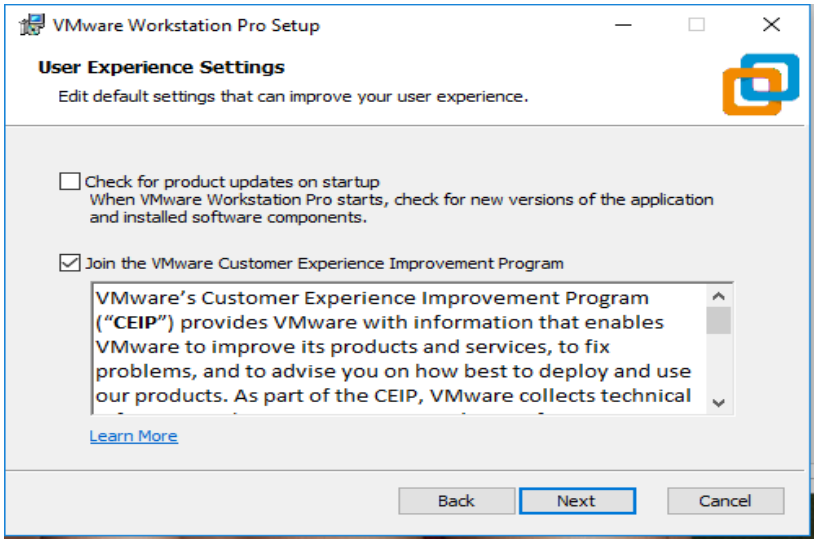
Step 5: Accept the terms and click "Next."



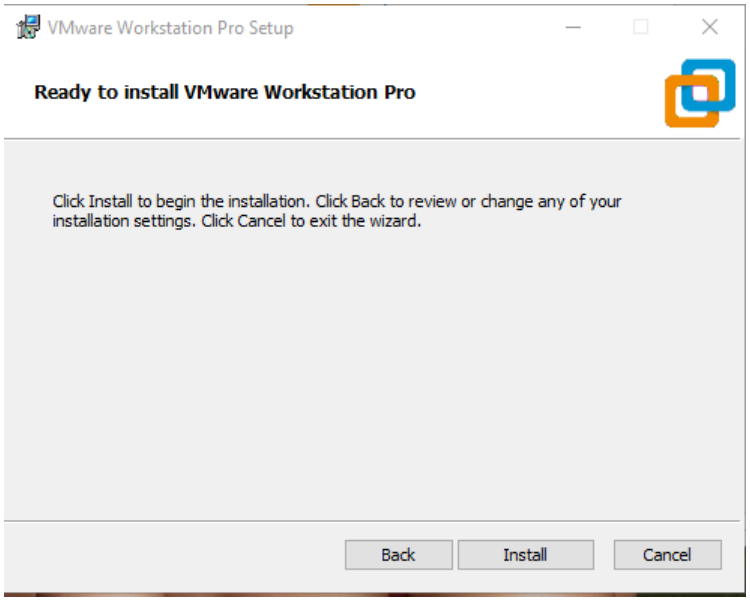
Step 6: On the next screen, it will ask for some additional features. It is not mandatory to check this box. Click on Next.



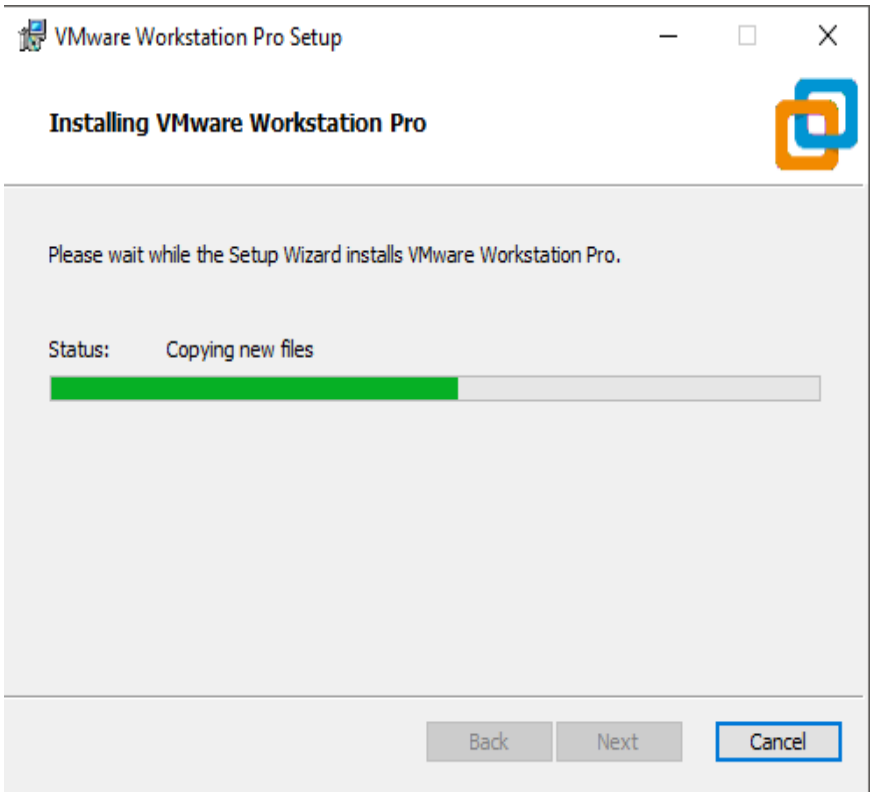
Step 7: On the next screen, some checkboxes are populated, Check them as per your requirement. Click on Next.



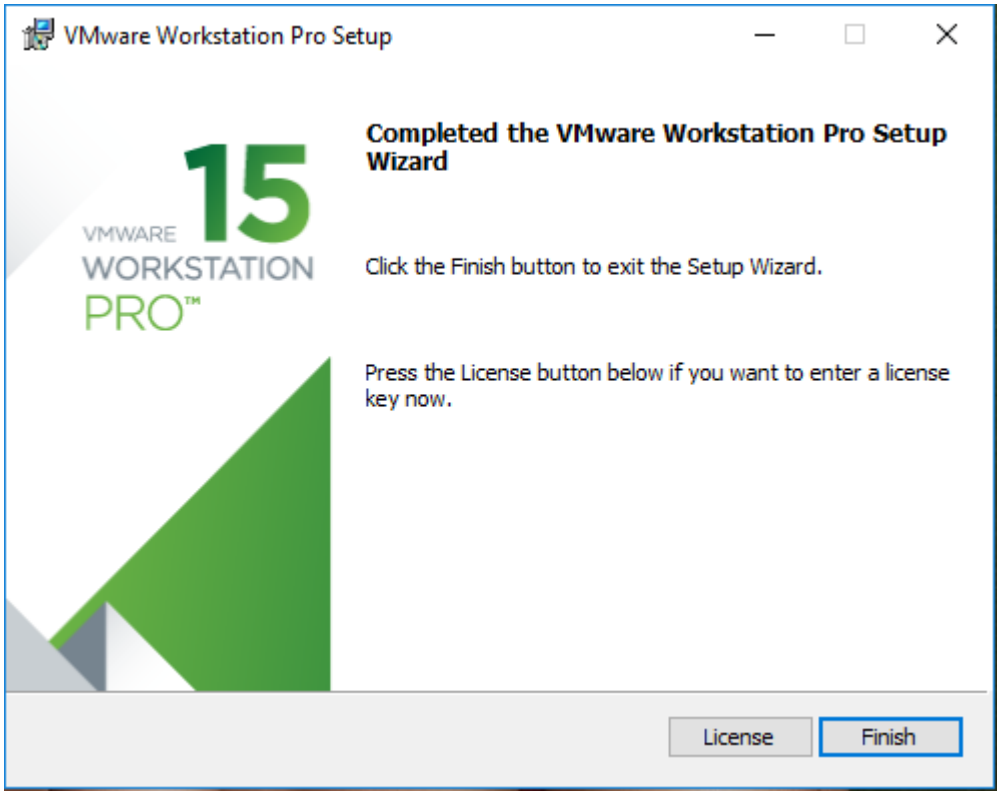
Step 8: At this step, VMware Workstation is ready to install. Click on "Install."



Step 9: At this step, you can see installation taking place. The installation will take some time. Wait for it to properly install.



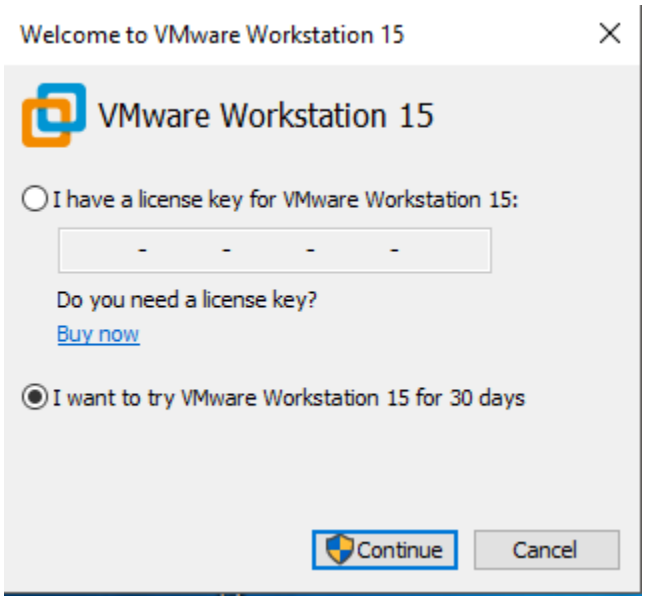
Step 10: Once the installation is completed, you will see the following dialogue box. Click on Finish. If you have purchased the product and have a license key, then you can click on License to enter the key.



Step 11: Upon completion, the window will close, and you will see the VMware Workstation installed icon on your desktop. The icon looks like this,

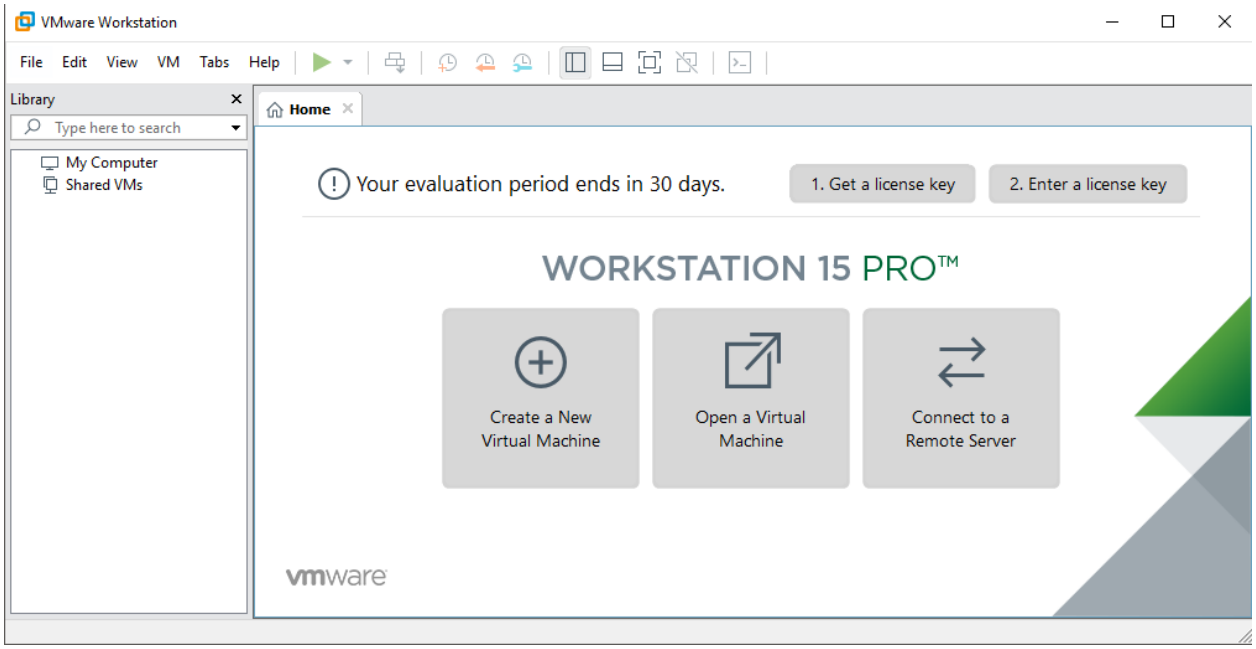
Double click on the icon to open the application.

Step 12: For the first time opening, if you have not entered the license key in step 7, then it will ask for a license key. You can go for the trial version, which is available free for 15 to 30 days. Click on Continue. Make sure you have Admin rights for this in Windows.



At this stage, you will get the final installation message. Click on Finish.

Finally, this will open a window for VMware Workstation Pro.



Now the VMware workstation is ready to use and can easily install a new virtual machine using the .iso file.

Installing Linux in VMware Workstation (steps)

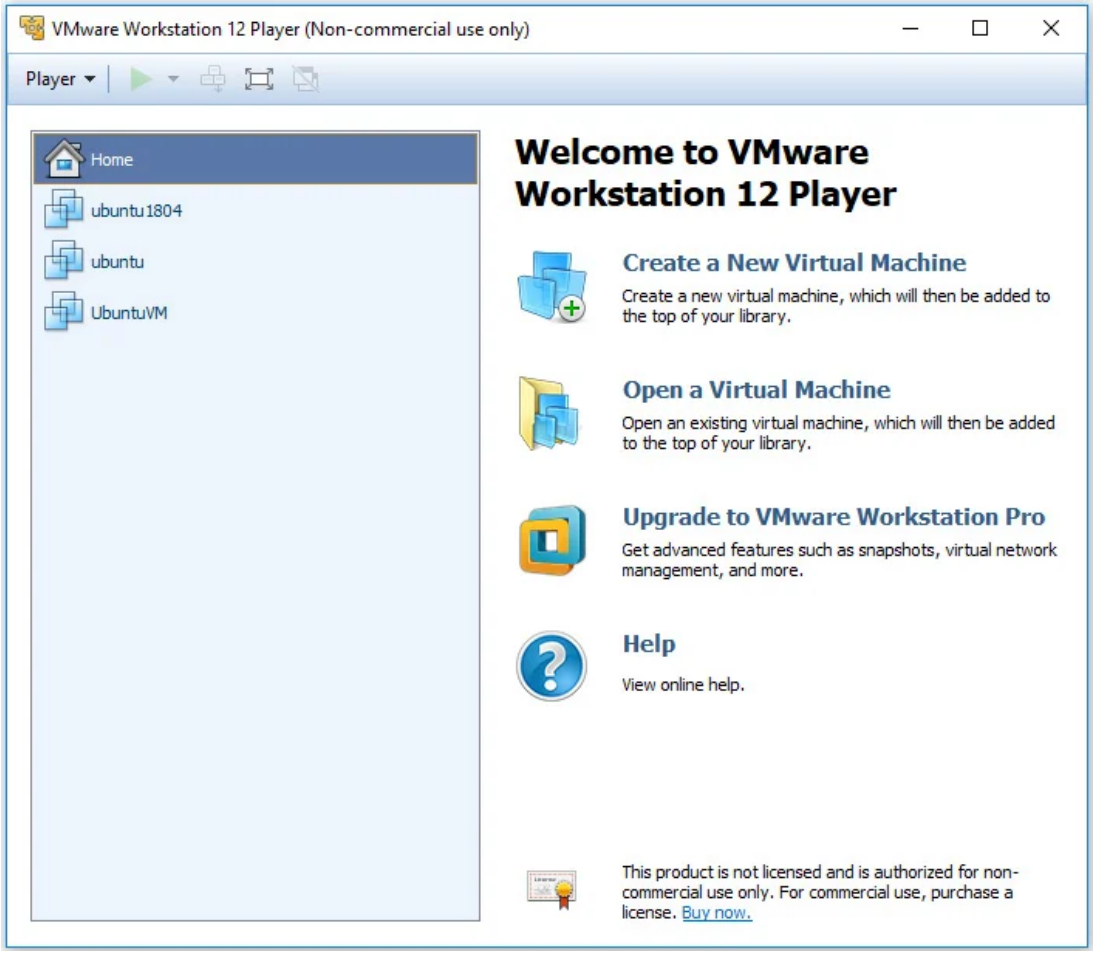
Here we will learn about the process of installing Lubuntu (which is a lightweight version of Ubuntu) on VMware. The steps are easy to demonstrate, and necessary screenshots are also provided to guide you throughout the process of installation.

Steps to Installing Lubuntu on VMware Workstation

After installing VMware, we need to download Lubuntu OS using the link below:

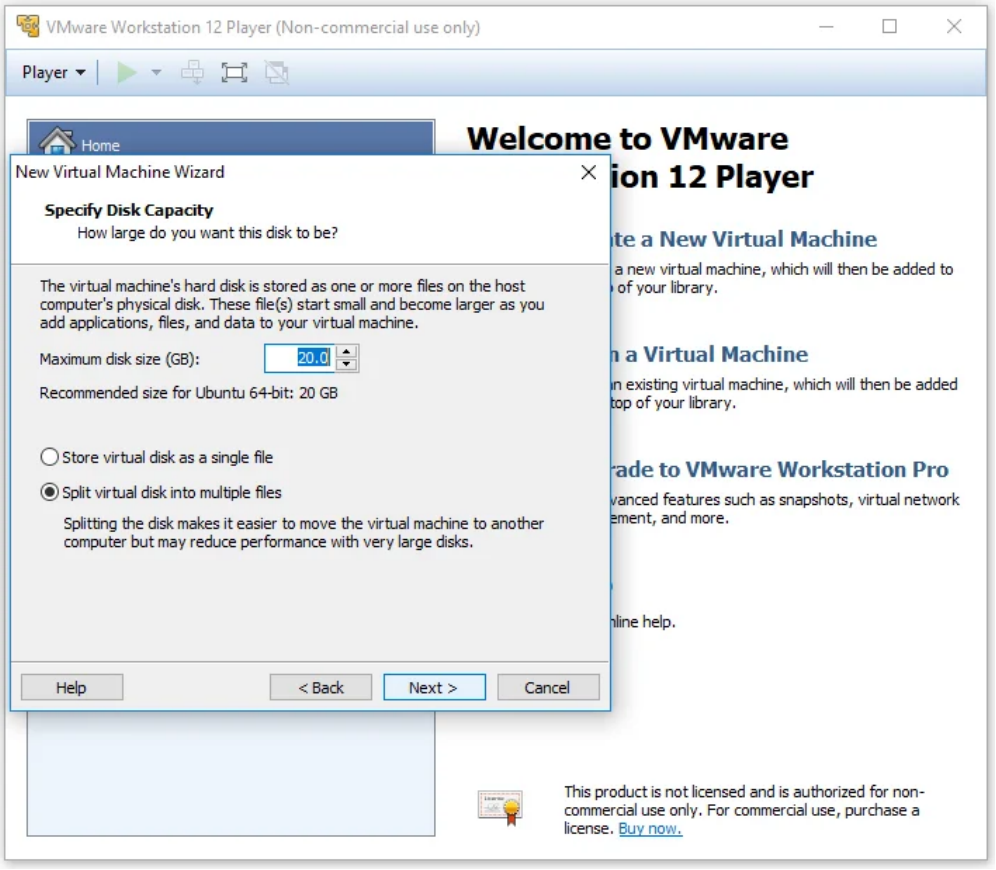
https://lubuntu.me/downloads/ (download the version as per your preferences).

Step 1: To start with the installation process, open the VMware Player that you have downloaded using the above link. When you open the VMware Player, it looks something like this.

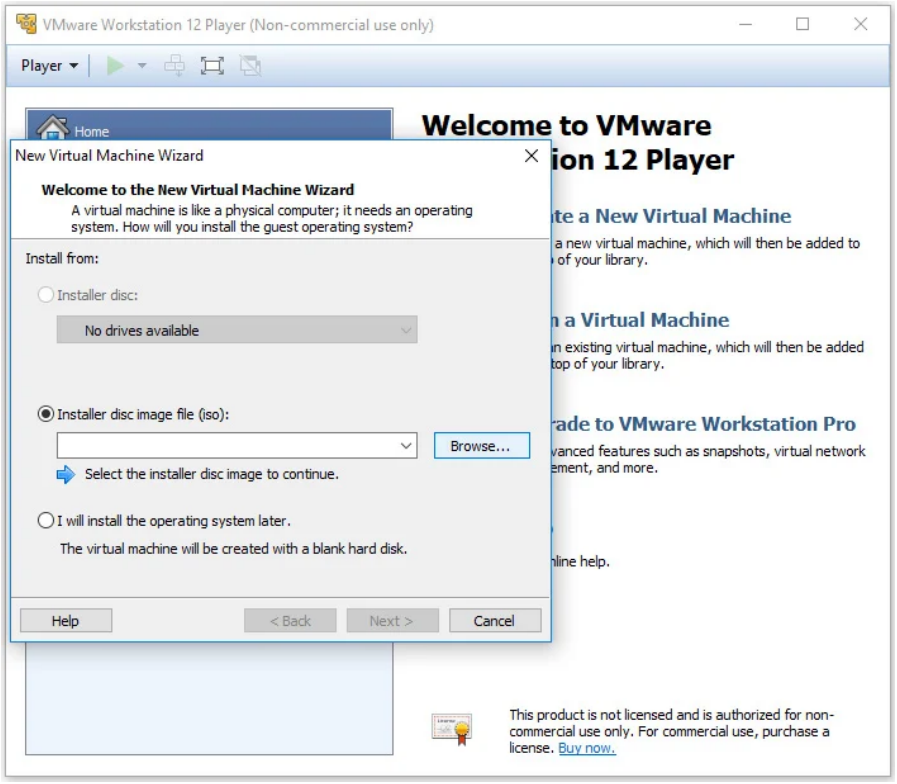


Click on the Create a New Virtual Machine option to begin the installation process.

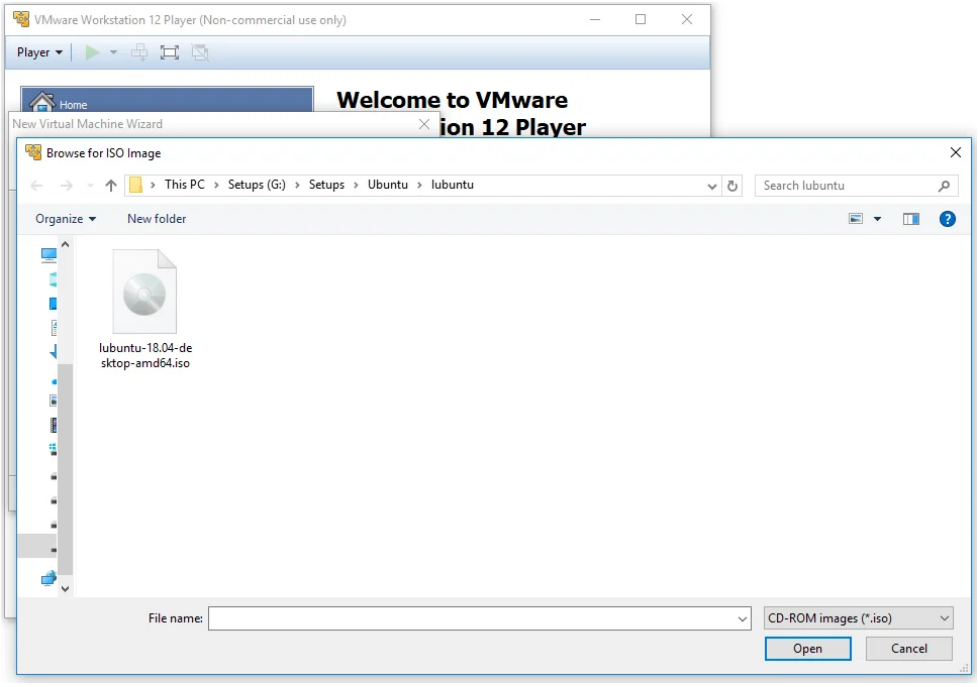
Step 2: Specify the Maximum Disk Size (20GB recommended) and select the "Split virtual disk into multiple files" option in the new window. Click the Next button and after.



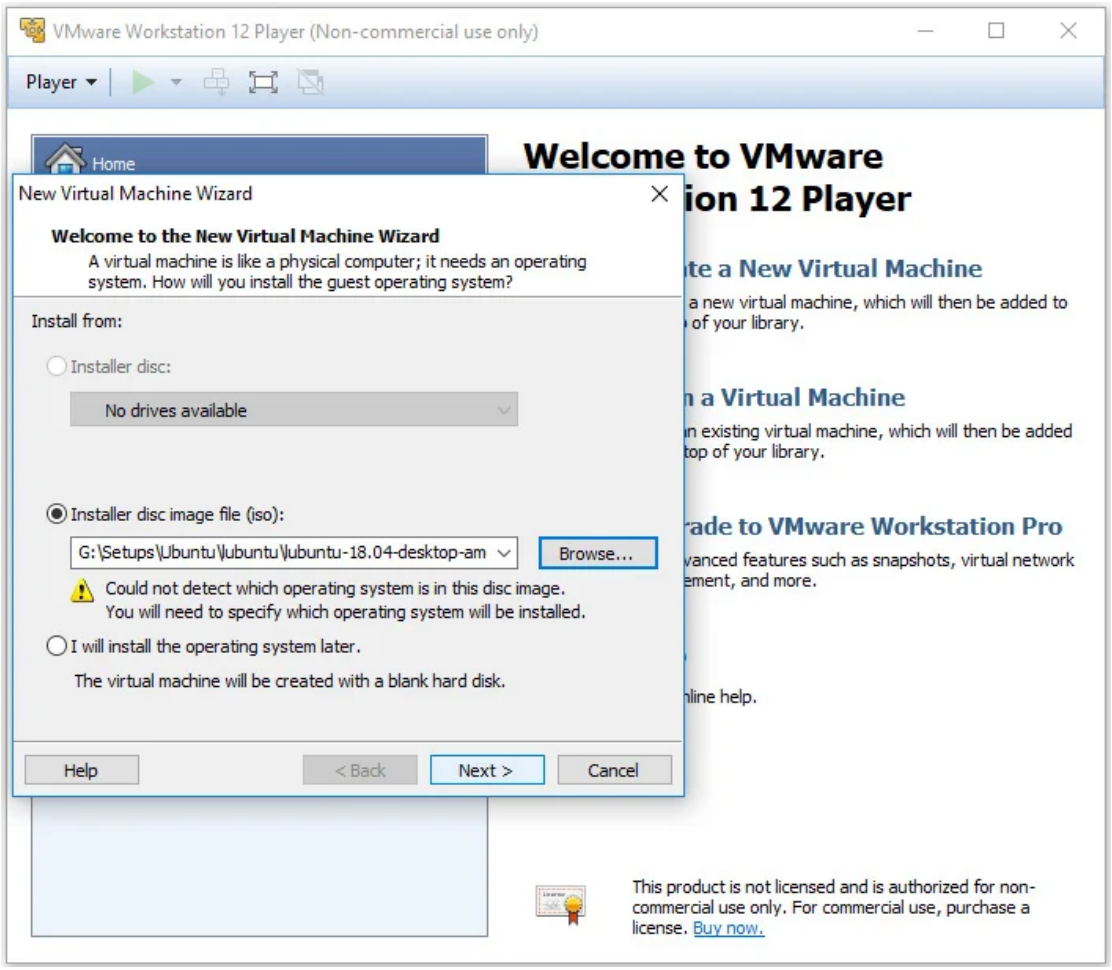
Step 3: Select the Installer disc image file (iso) option in the new window that appears on your screen.



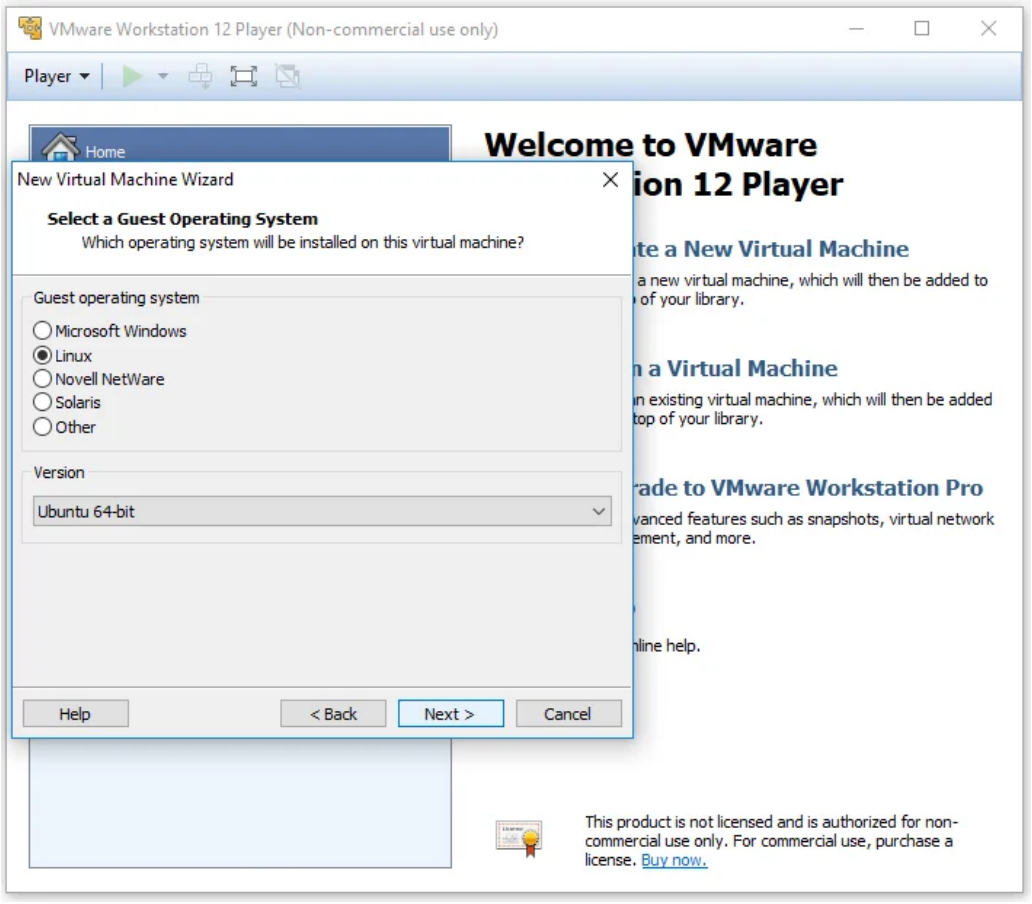
Step 4: Click on the "Browse" button to locate your ISO image.



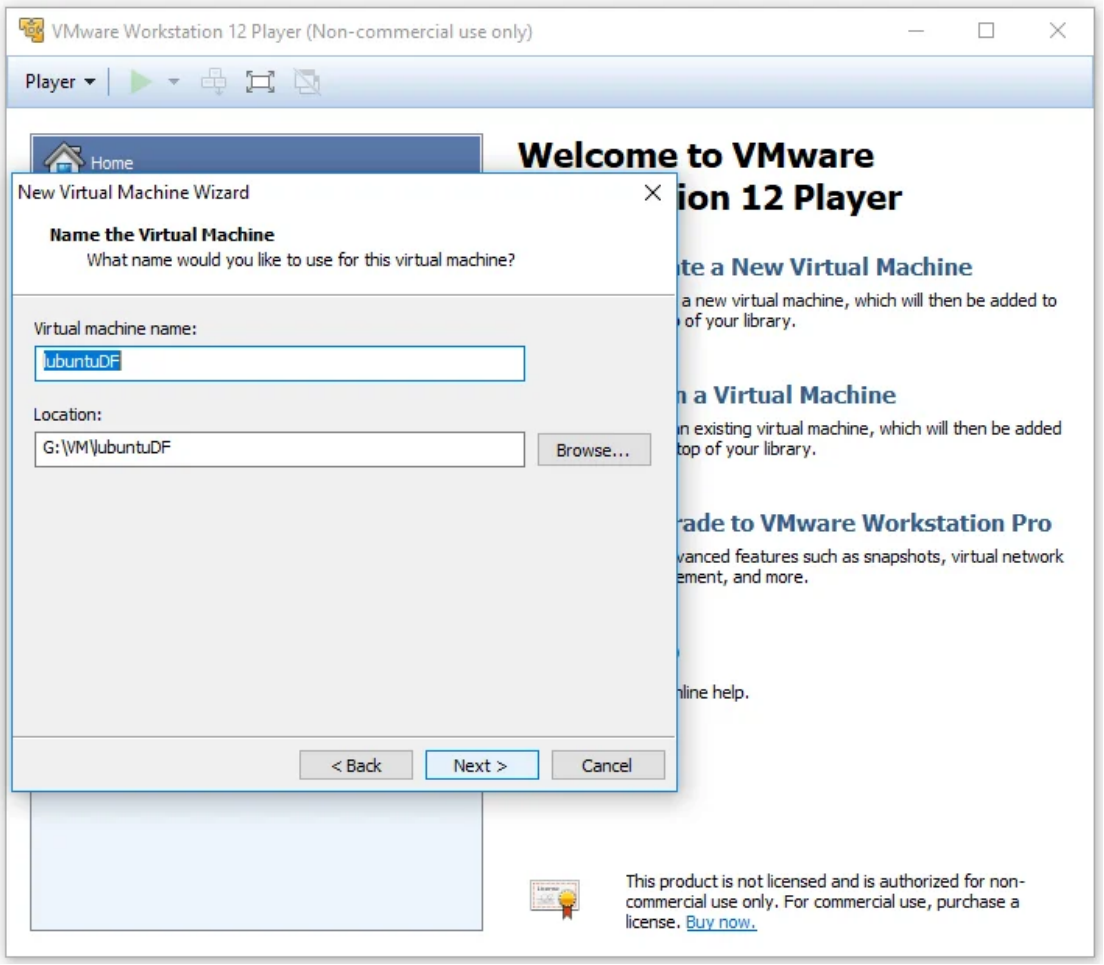
Step 5: Once you are done with selecting the ISO image, click on the Next button.



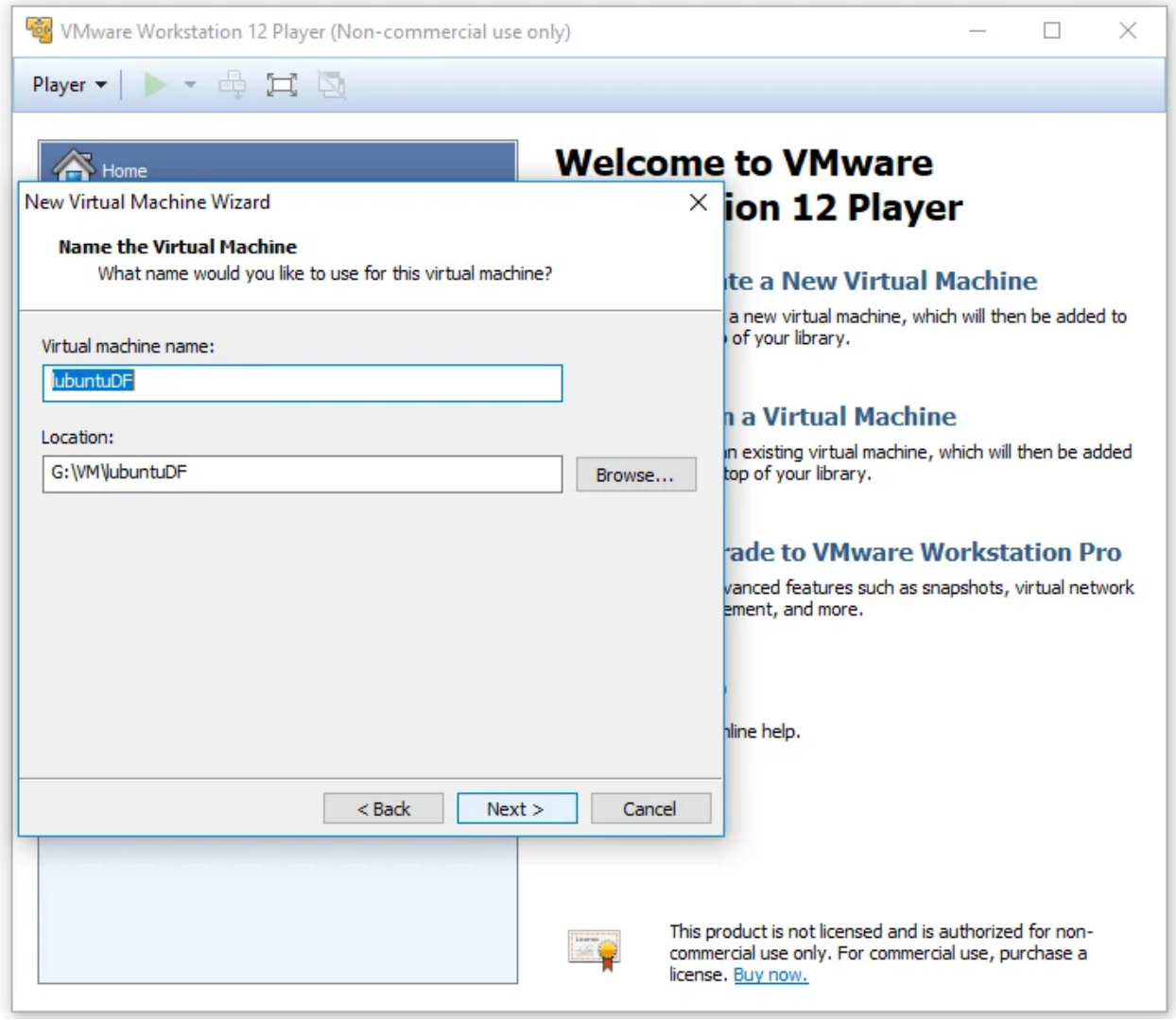
Step 6: Now, select the guest operating system that you want to install on the virtual machine. As of now, select the Linux option from the list and the version as ubuntu-64 bit. Click the Next button.



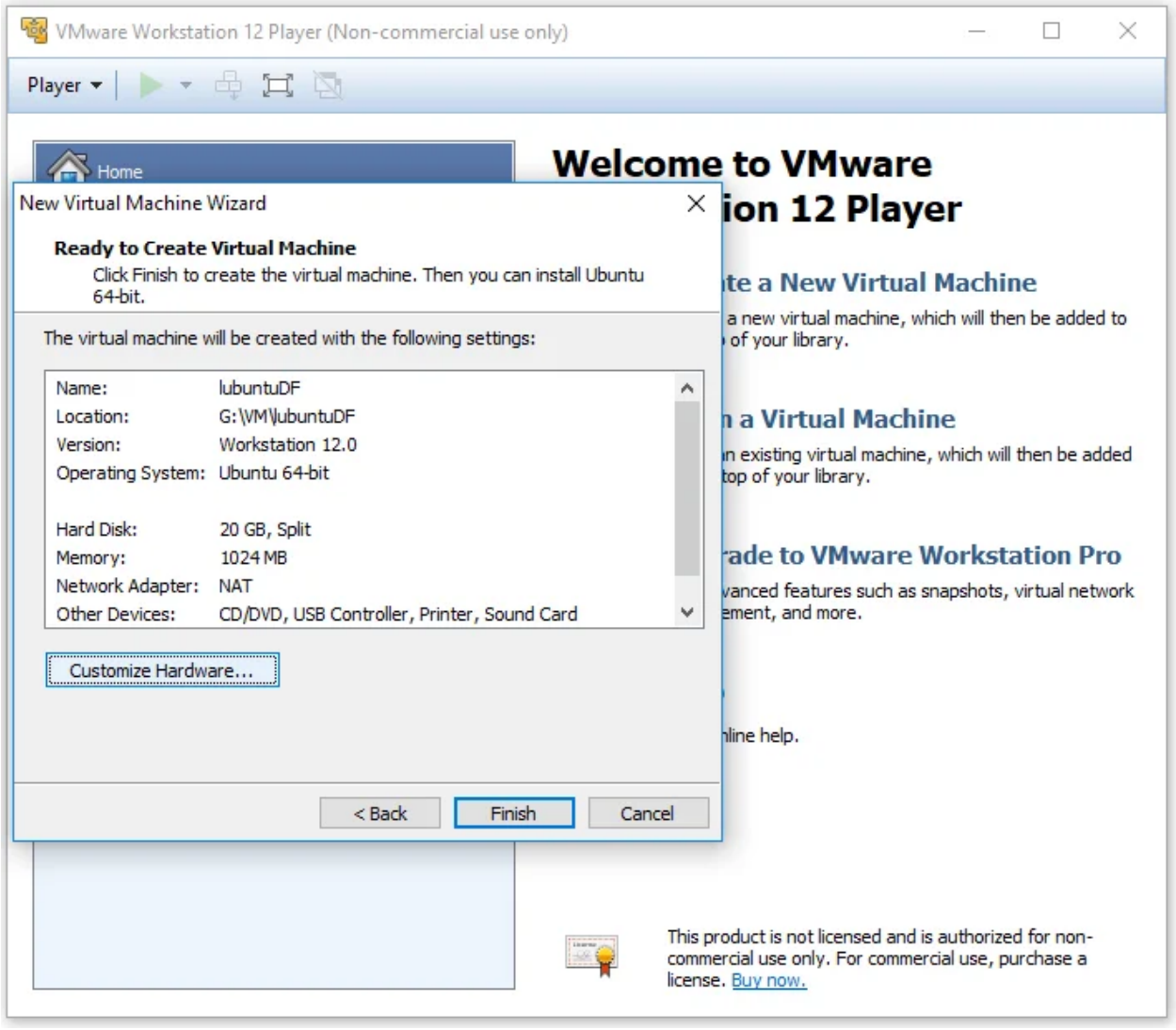
Step 7: Supply an appropriate name for the virtual machine as well as the location in which you want it to be saved. After doing so, click the Next button.



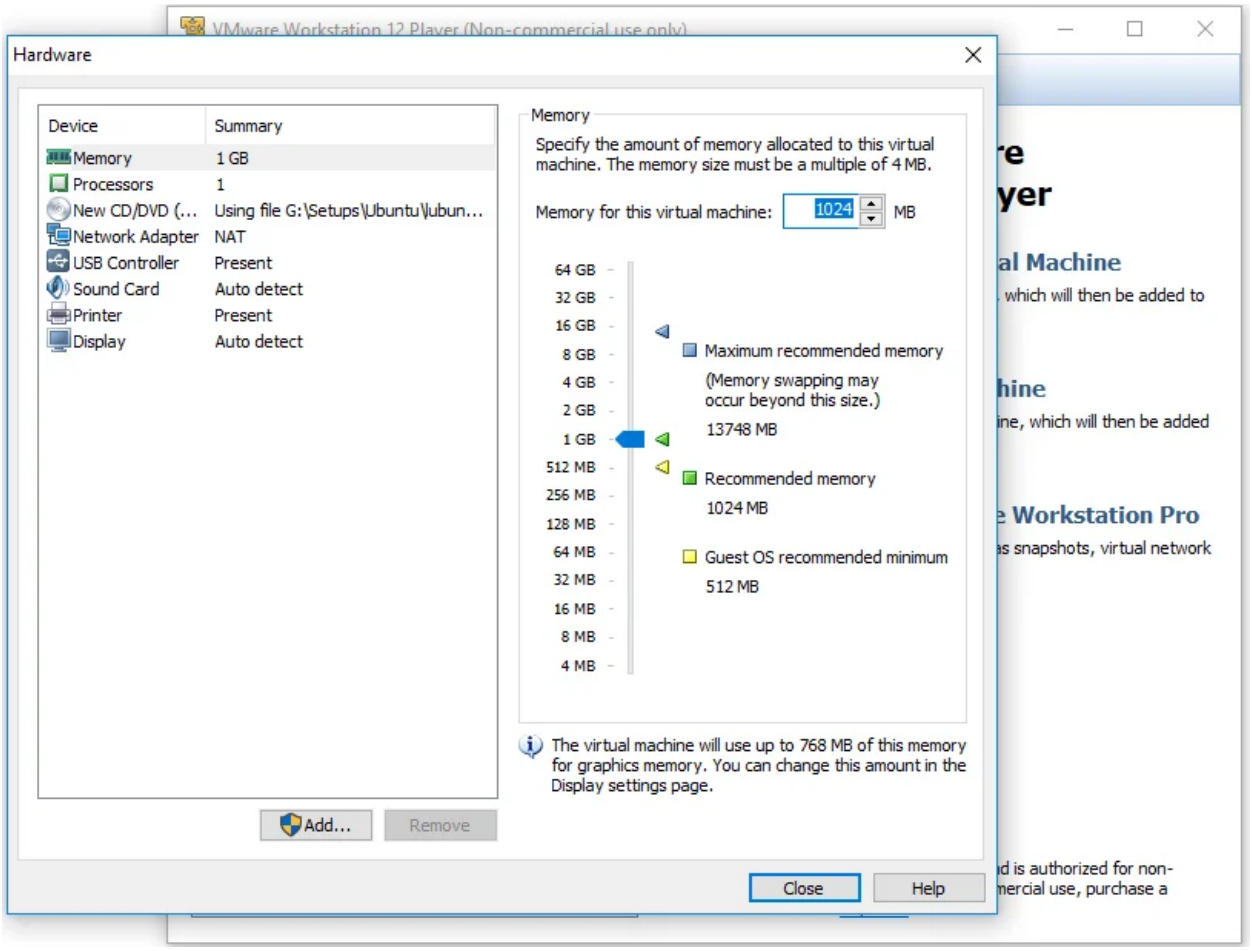
Step 8: Specify the Maximum Disk Size (20GB recommended) and select the "Split virtual disk into multiple files" option in the new window. Click the Next button and after that



Step 9: Once you click the Next button, a window appears which displays all the settings that you have chosen for the virtual machine.

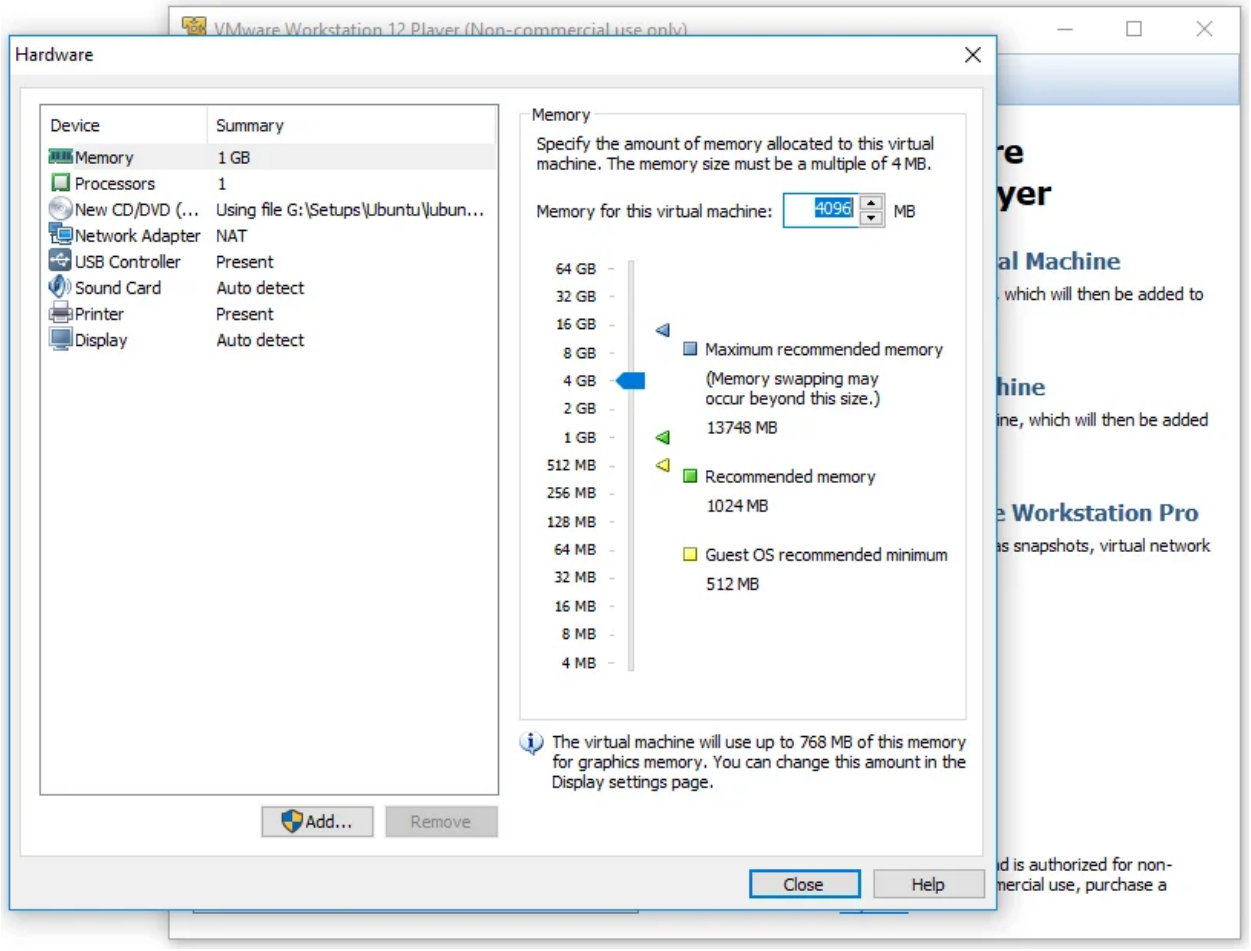


Step 10: Since you haven't mentioned the RAM for the virtual machine, click on the Customize Hardware button to select the RAM for the VM.



Step 11: Now you need to specify the amount of RAM that should be assigned to the VM.

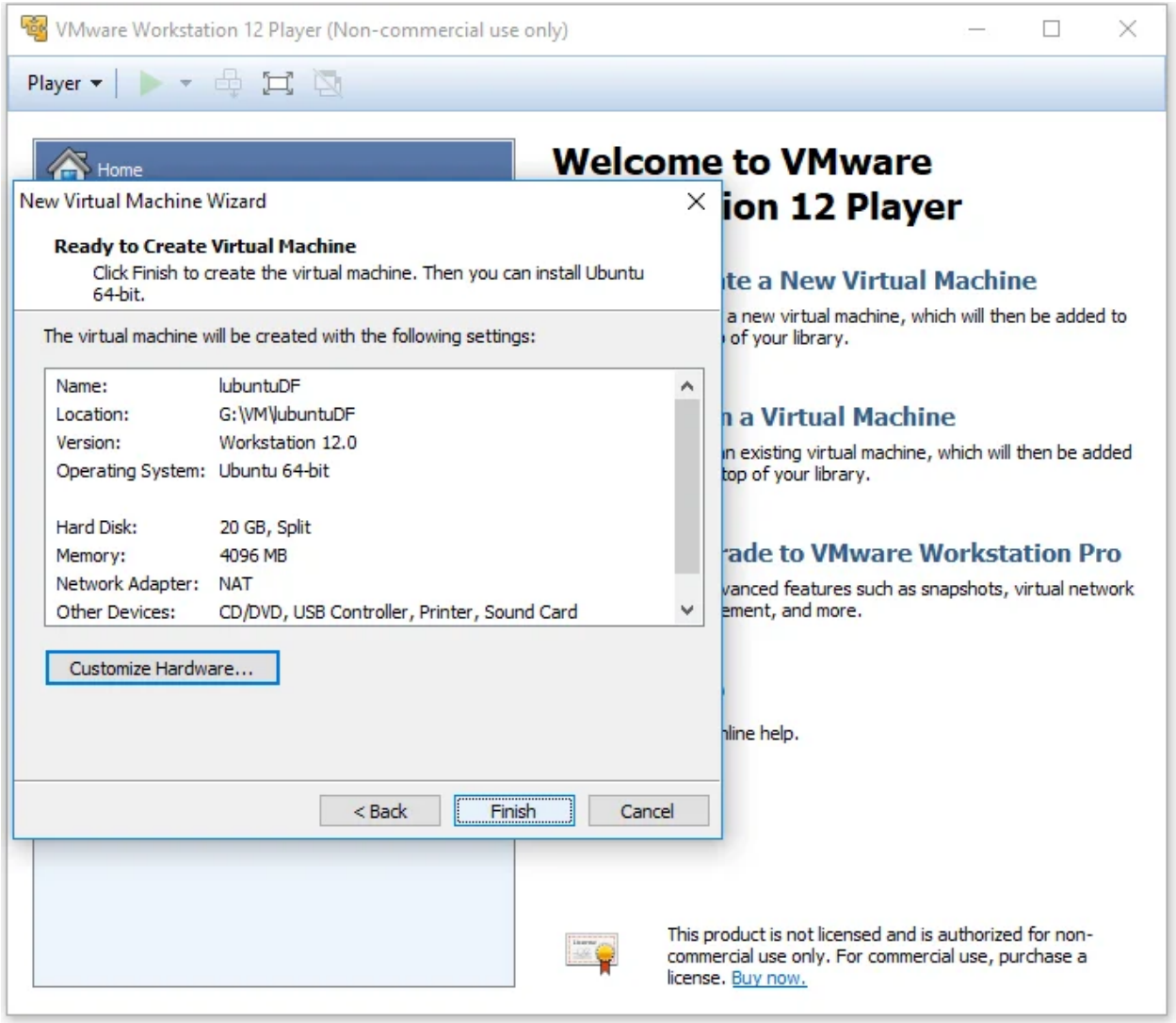
Note: It is highly recommended that the RAM assigned to the VM should be kept at half of the actual RAM of your system to reduce the load on your system.



Step 12: After rechecking all the settings, click on the Finish button.

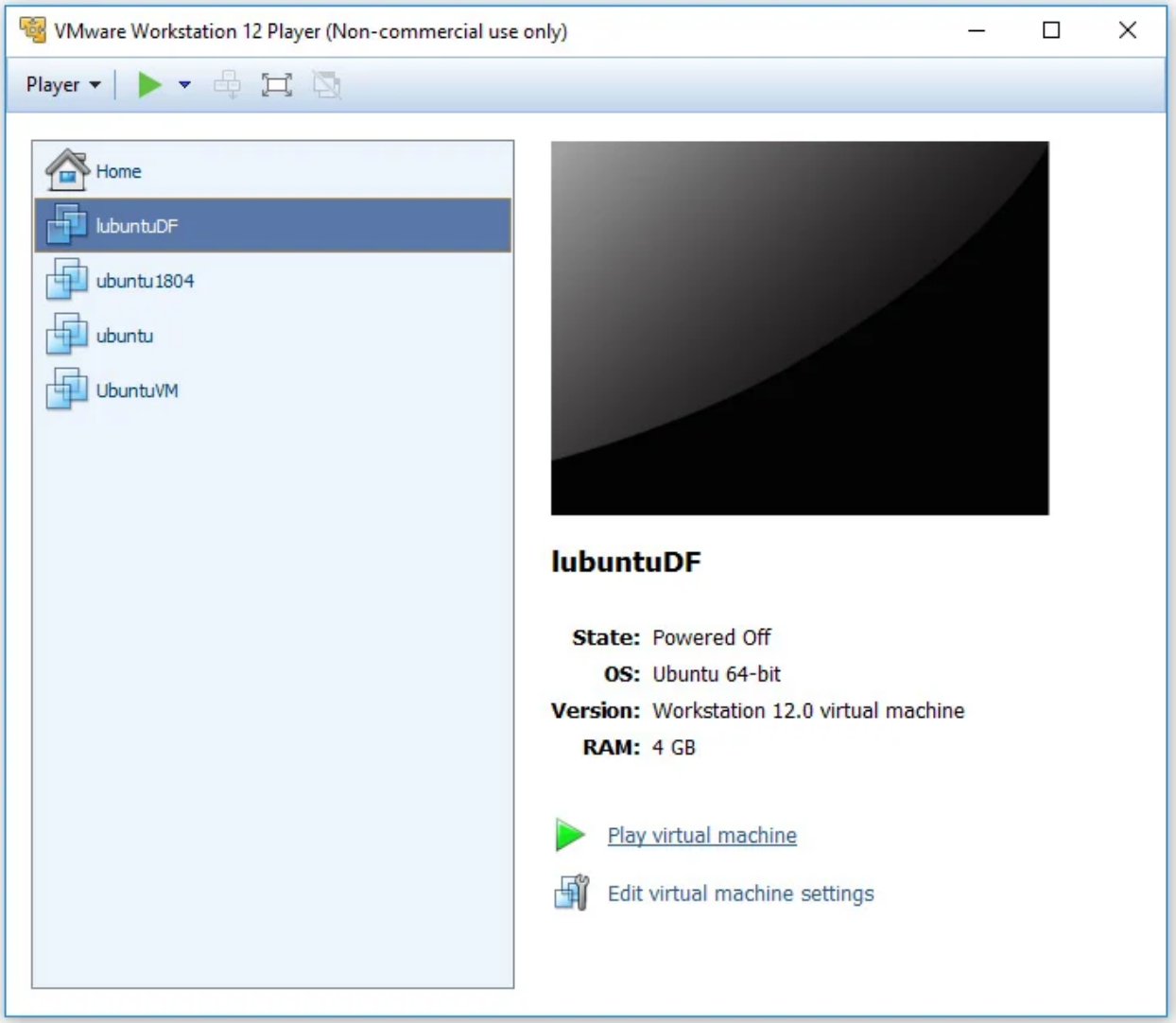
Your VM is now ready to be created.

Note: If you still want to change the settings of the VM, you can do that by clicking on the Customize Hardware button again.



Step 13: After you click on the Finish button, your VM is now ready to be installed with your specified settings. The new window looks something like this.

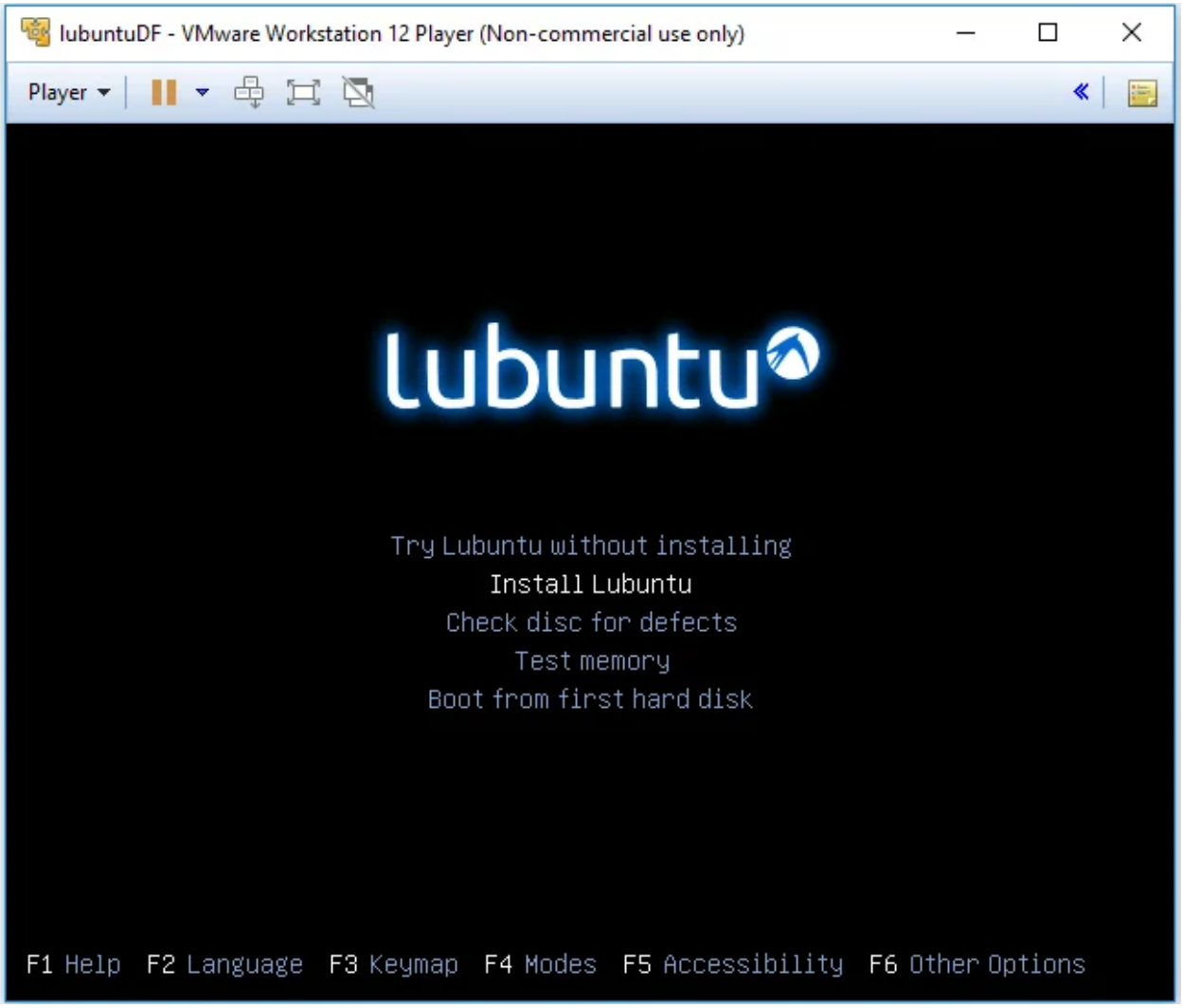
Click on the "Play virtual machine option" to proceed further.



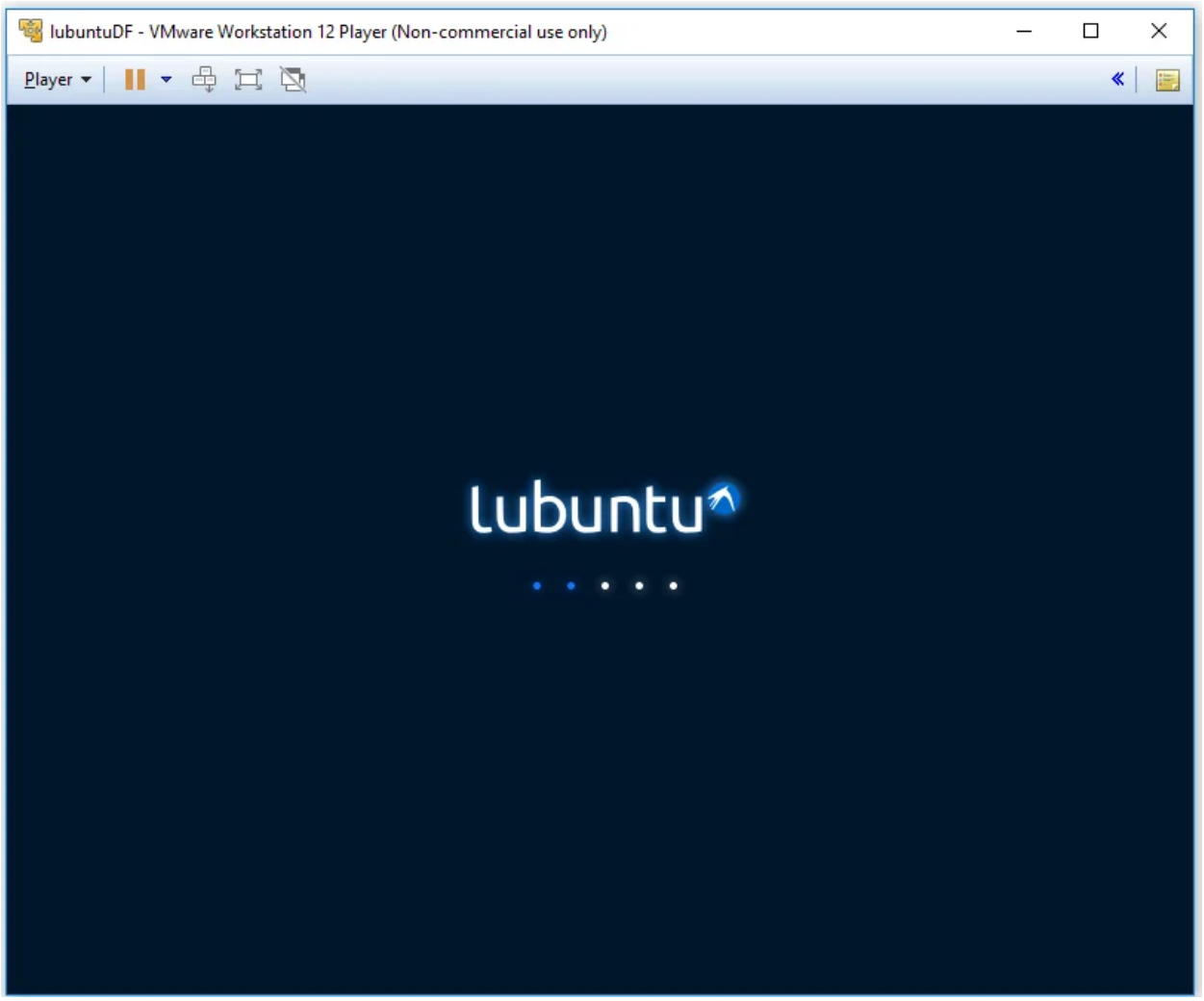
Note: From here onwards, your only input will be the keyboard, as your mouse won't work after this.

Step 14: Now, with the help of your keyboard, select your language as English and then press Enter.

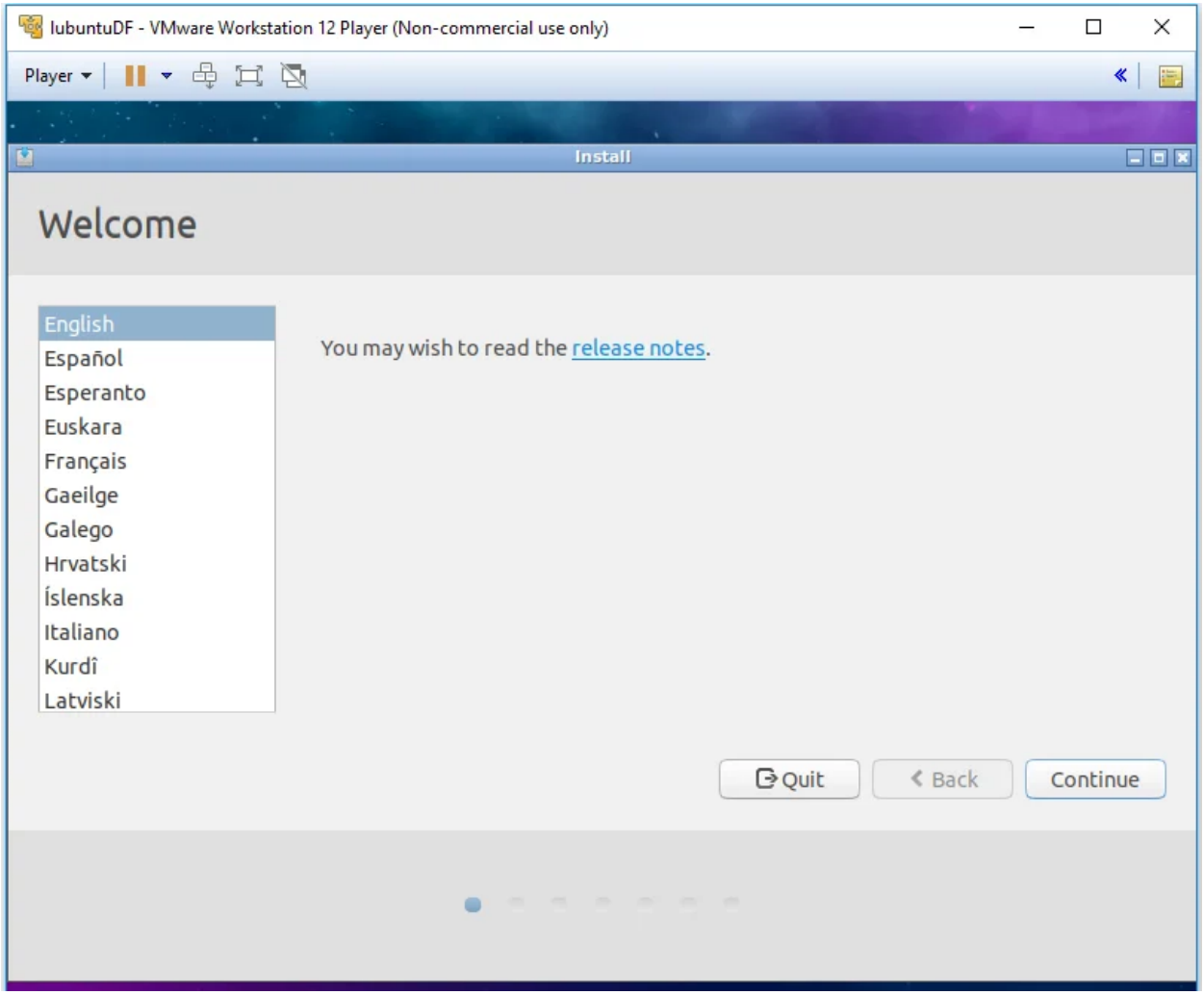
Step 15: Now select the Install lubuntu option in the new window that appears to start the installation of the lubuntu guest OS on your VM.



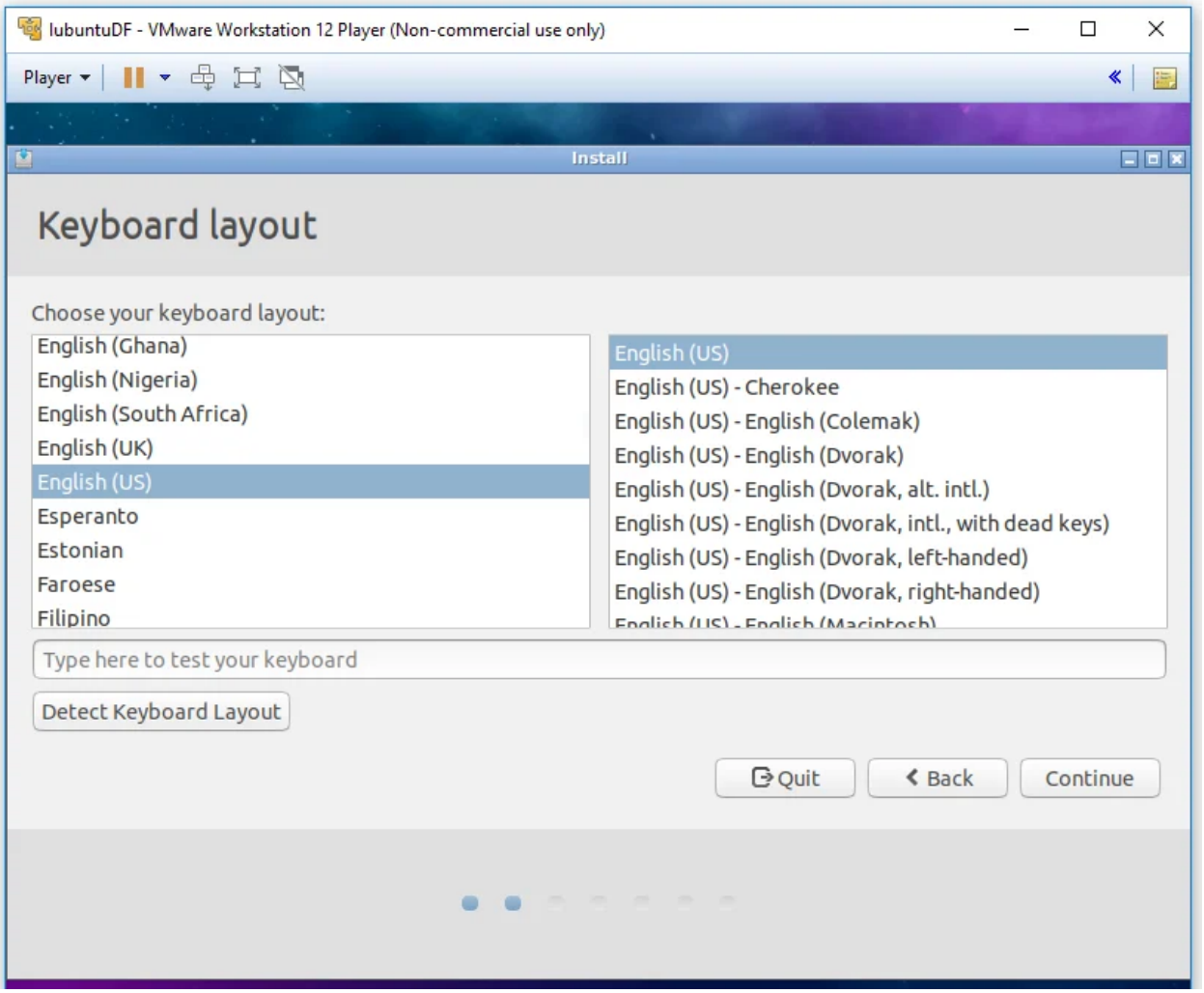
Please wait for a few seconds (depending on your machine) for the installation process to be completed.


lubuntuDF - VMware Workstation 12 Player (Non-commercial use only)
Player
lubuntu

Step 16: Select English as the default language of the VM and then click on the Continue button.

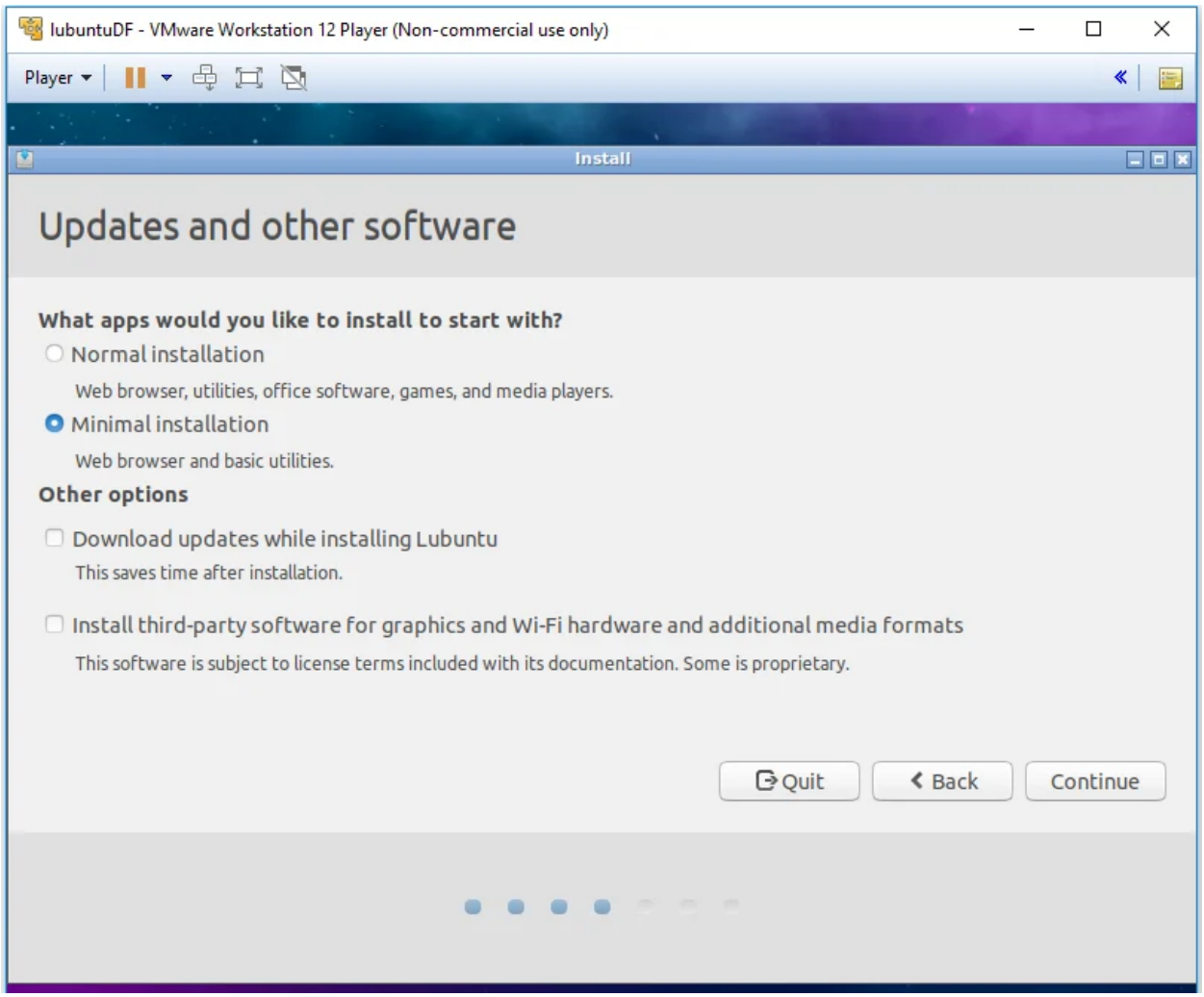


Step 17: Select an appropriate keyboard layout and then click on the Continue button.



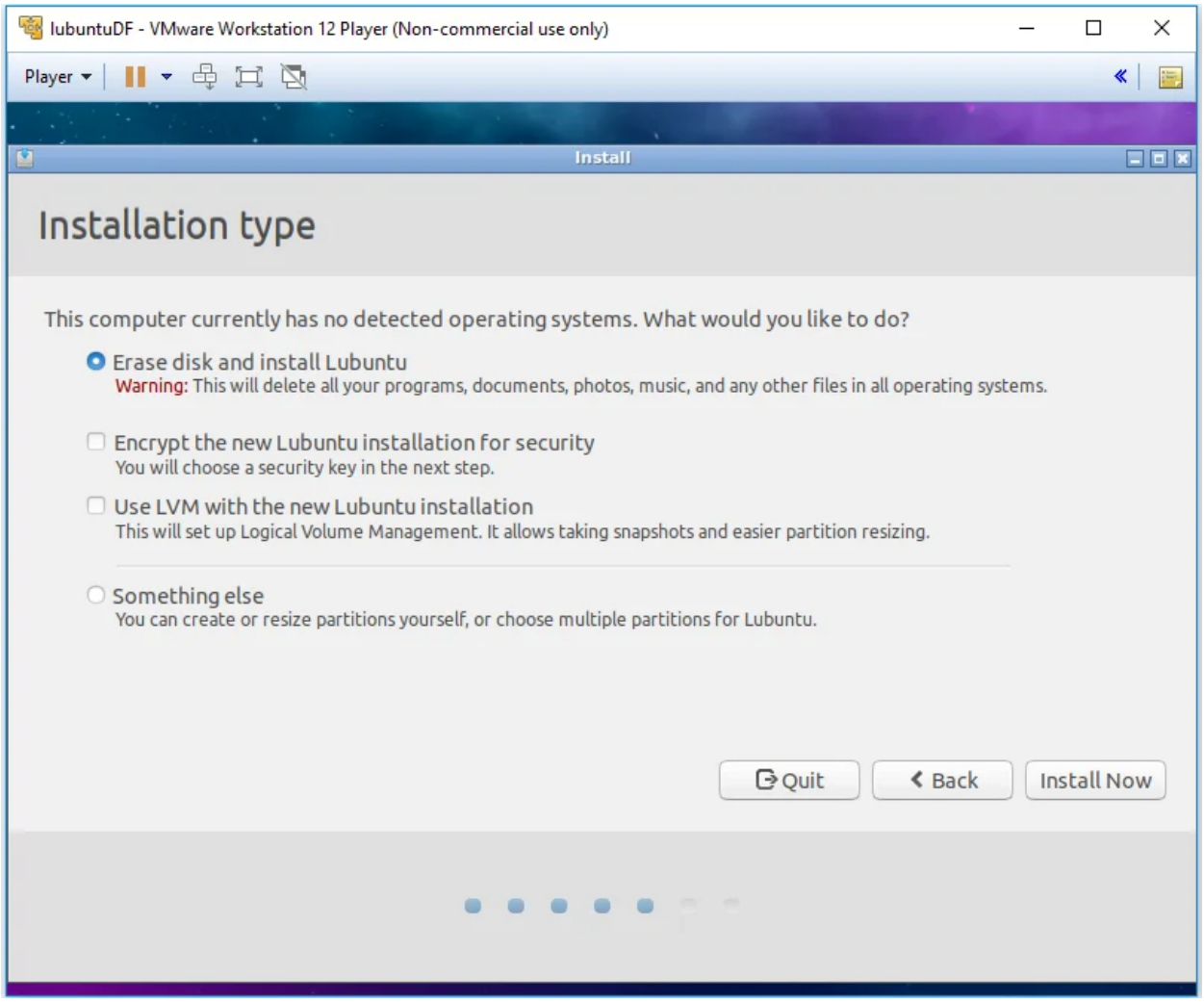
Step 18: In the Updates and other software window, select the Minimal installation option. This reduces the load on your system and enhances the performance of your VM.

Now click on the Continue button.

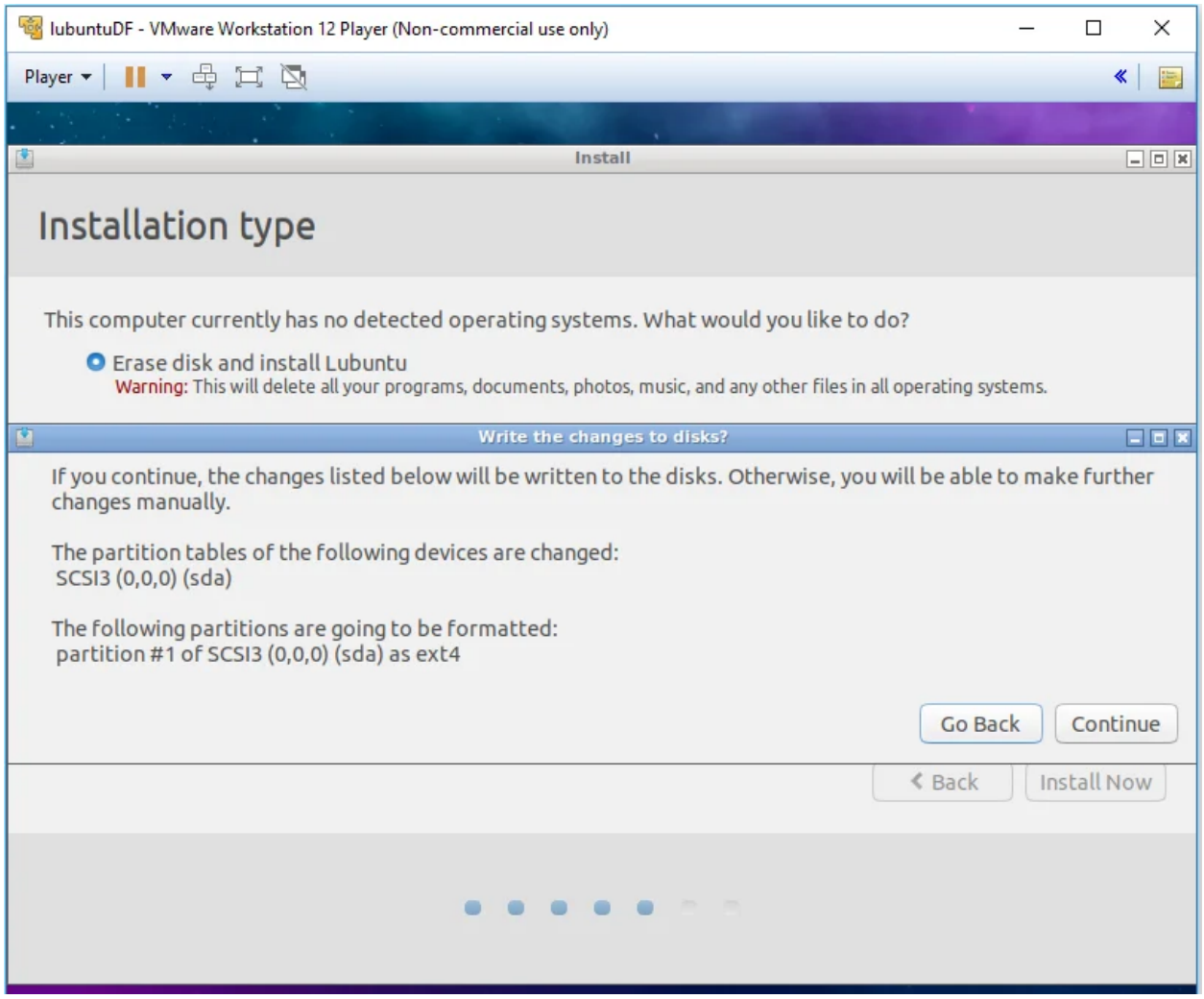


Step 19: Now you need to choose the installation type from the new window. Select the Erase disk and install Lubuntu option from the list and then click on the Install Now button.

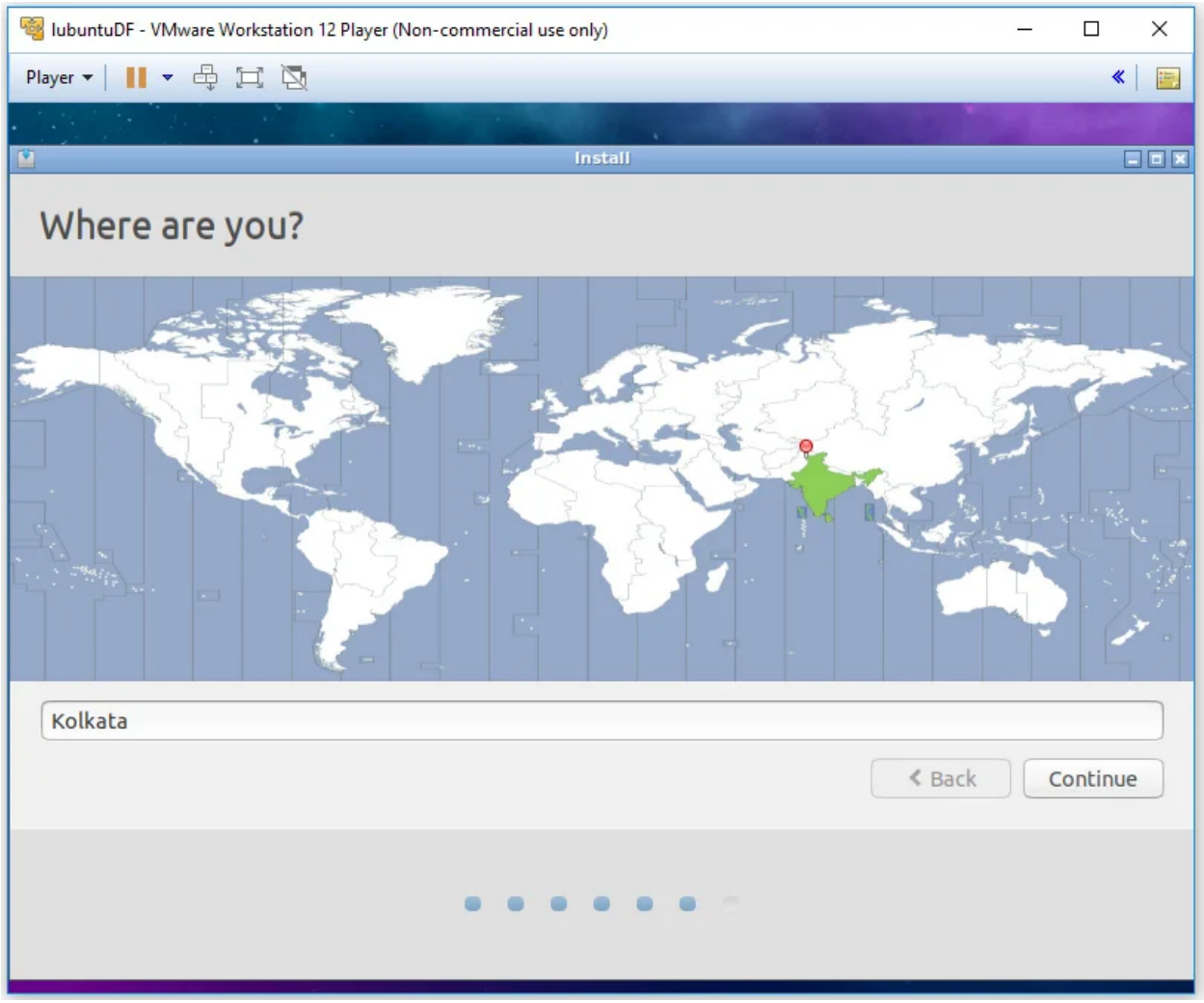
Note: Selecting this option won't erase the hard disk of your system, in fact, it will erase the disk of your VM. So relax, your system is safe, no harm can be caused.



Step 20: Once you click on the Install Now option, a new warning window appears stating the consequences of selecting that option. Click on the Continue button to proceed further with the installation.

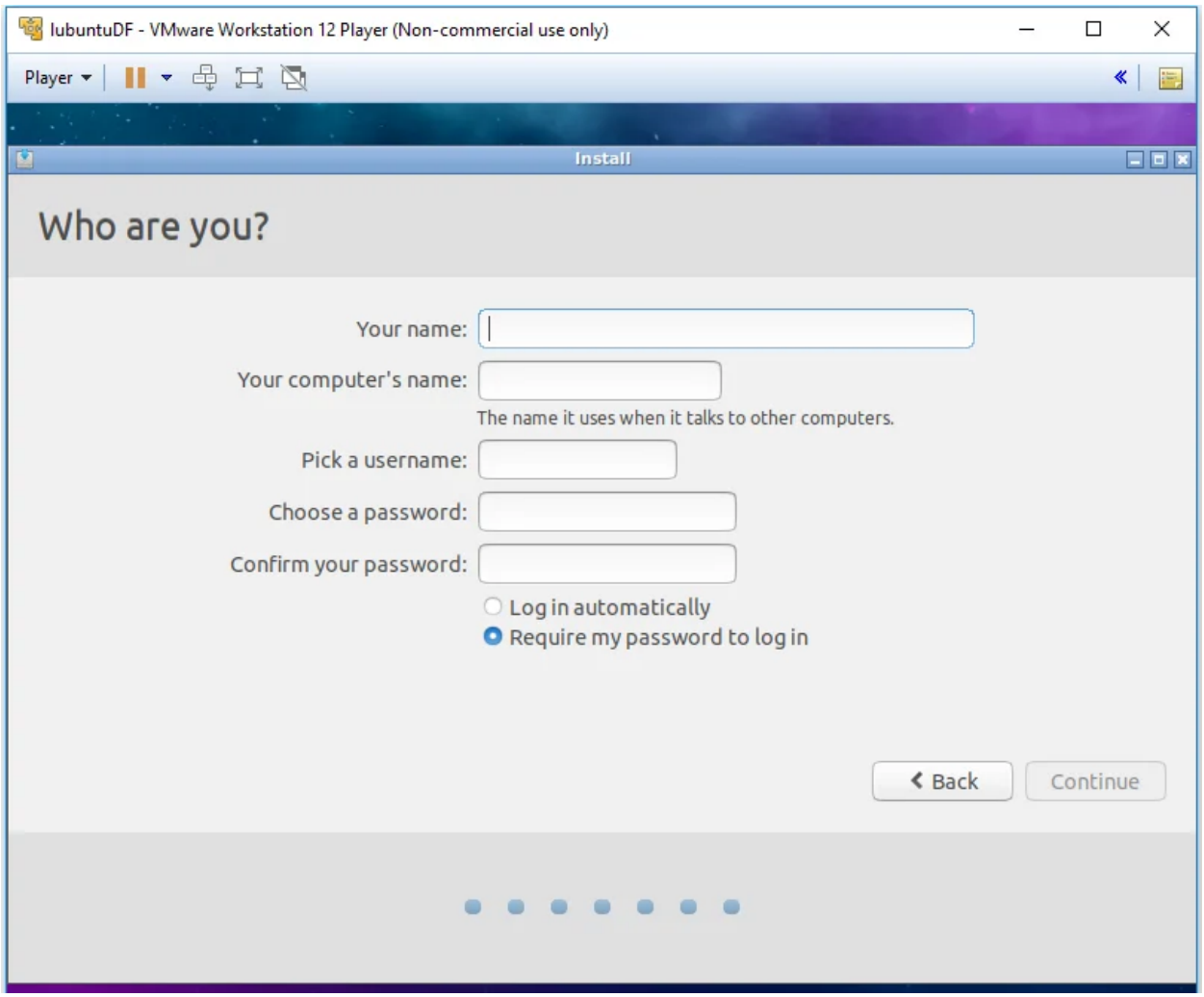


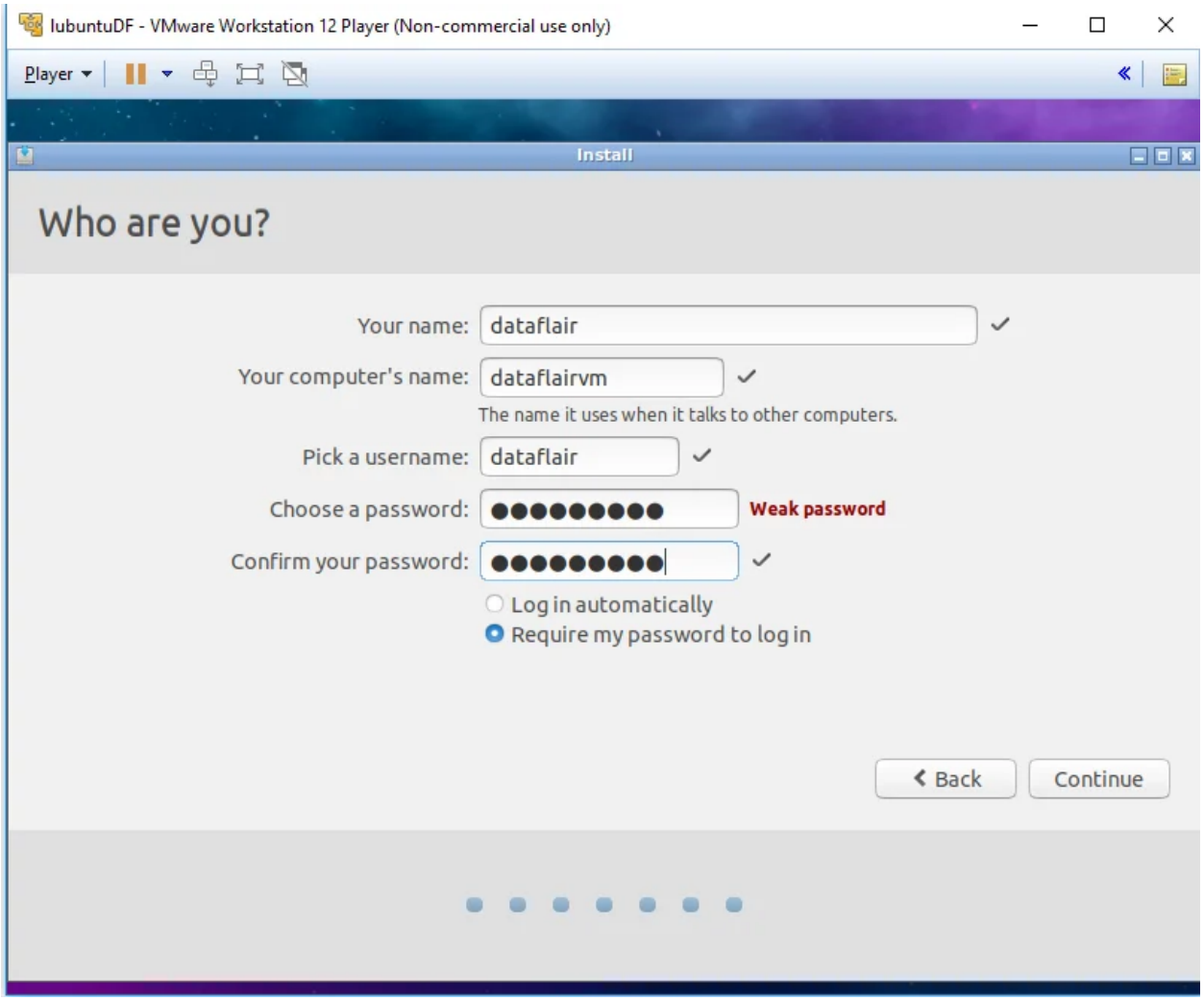
Step 21: Now in the next window, select your current location and then click on the Continue button.



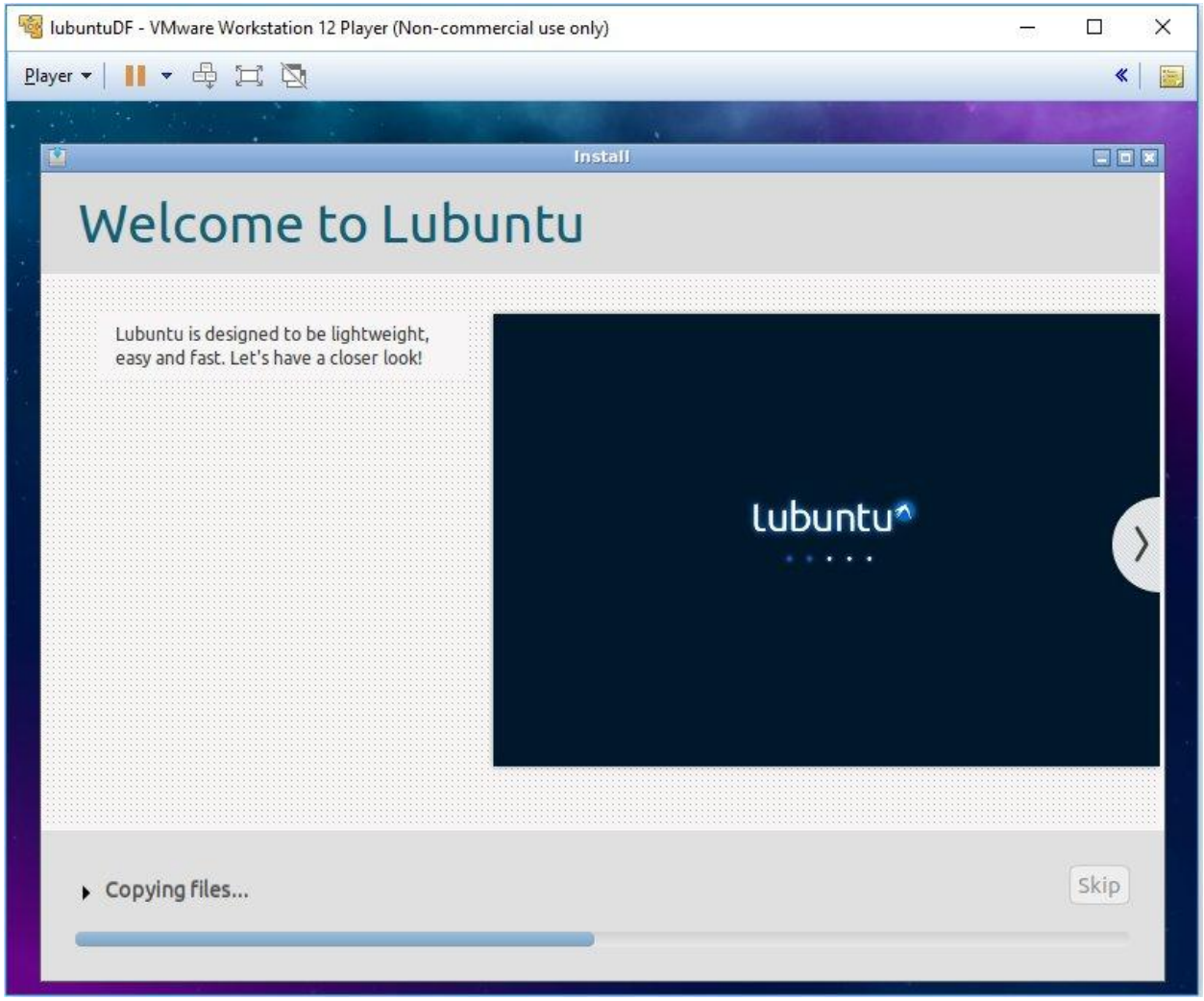
Step 22: Fill in your appropriate details in the form that appears in the next window and then click Continue.

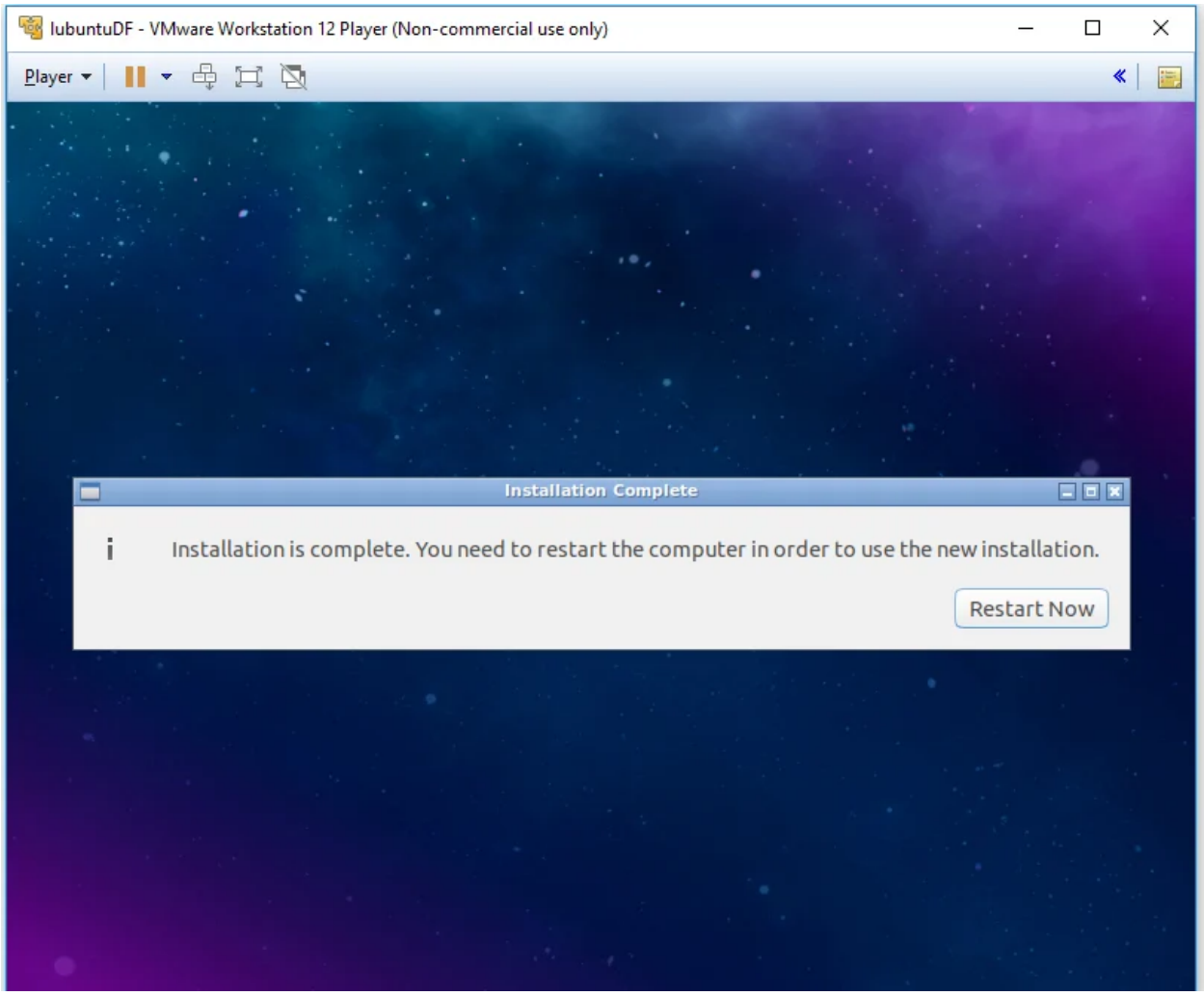
Note: Do remember your username and password for later use.



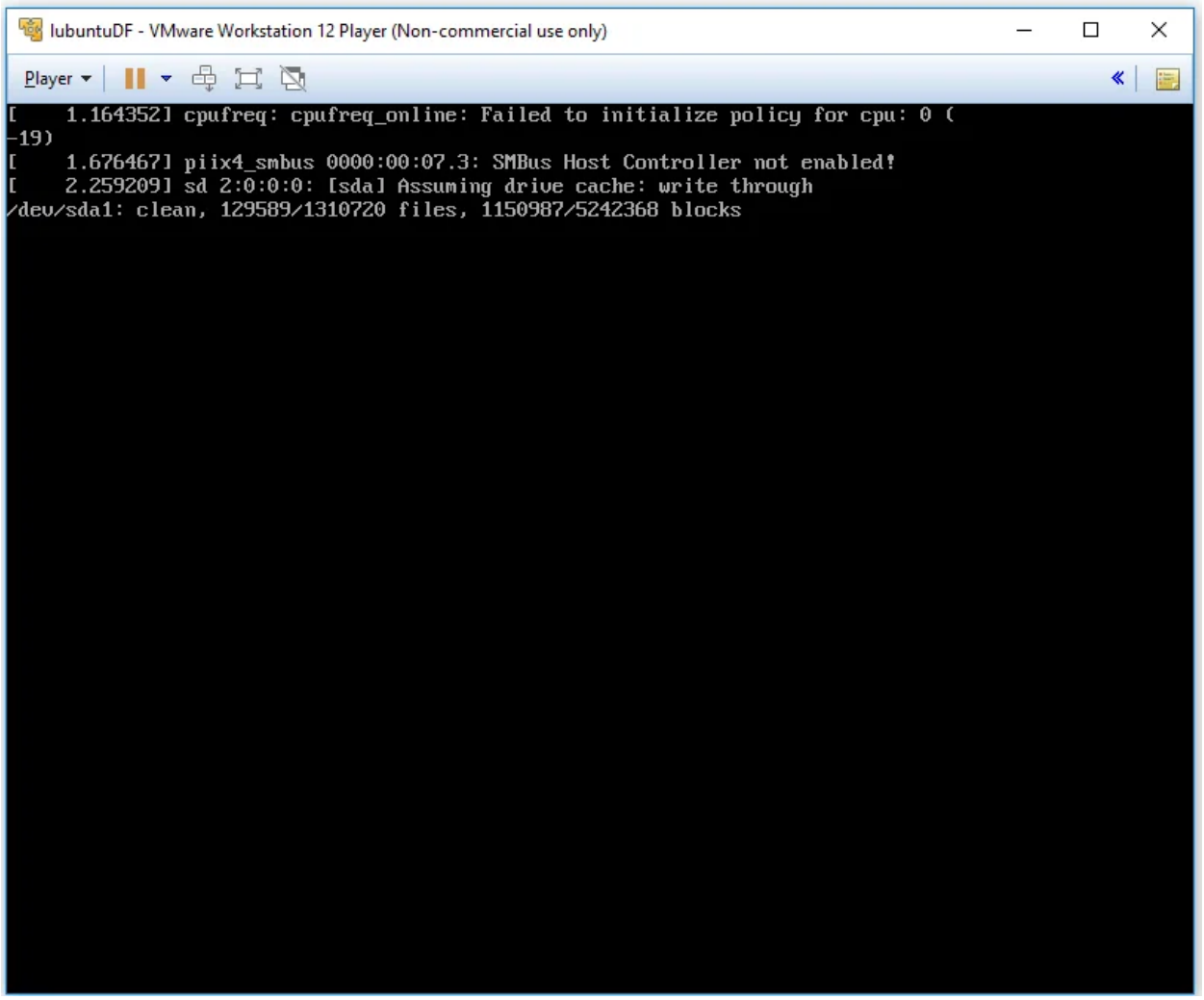

lubuntuDF - VMware Workstation 12 Player (Non-commercial use only)
Player
Install
Who are you?
Your name:
dataflair
Your computer's name:
dataflairvm
The name it uses when it talks to other computers.
Pick a username:
dataflair
Choose a password:
Weak password
Confirm your password:
Log in automatically
Require my password to log in
Back
Continue

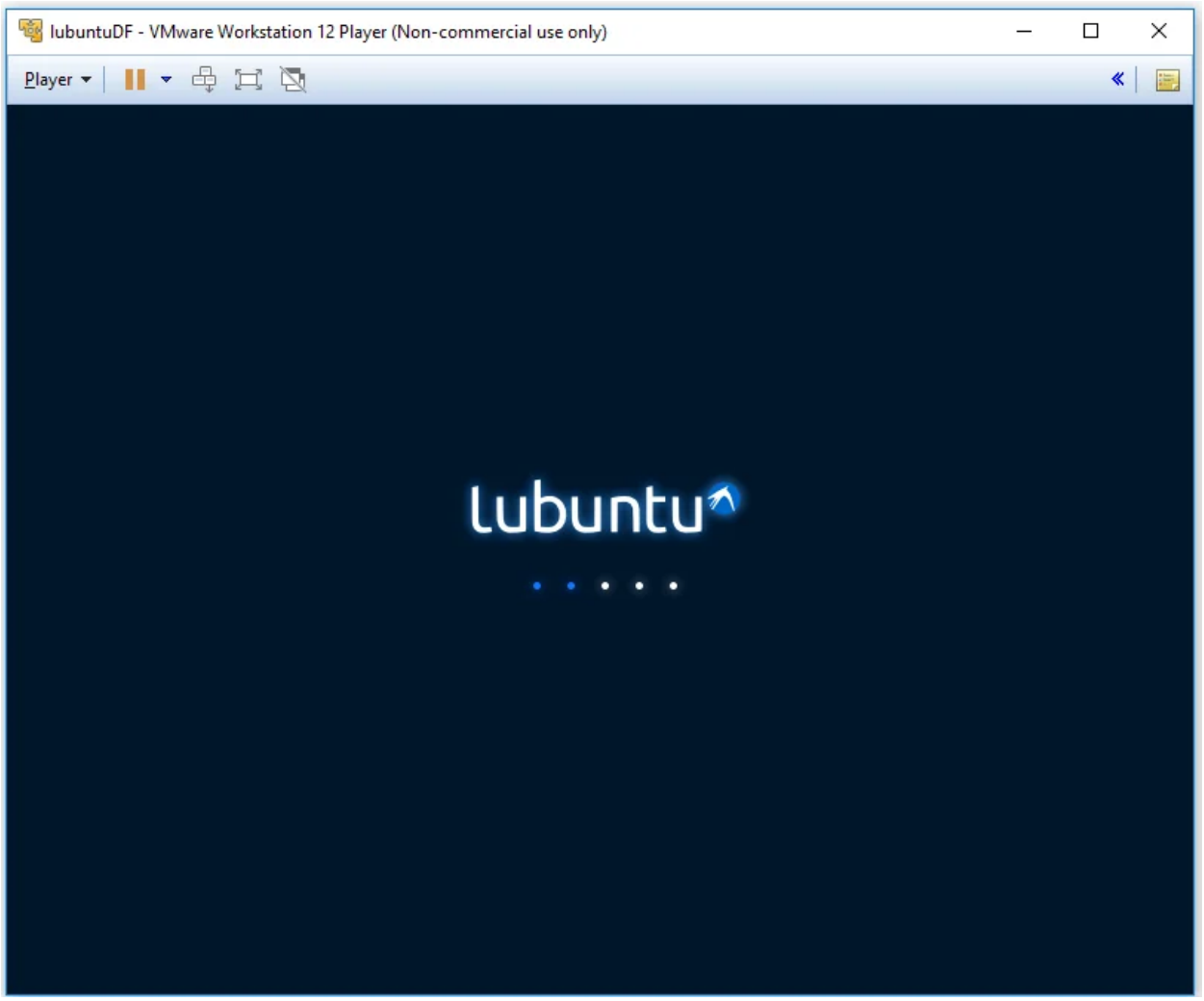
Step 23: Wait for a few minutes until the installation process is completed.



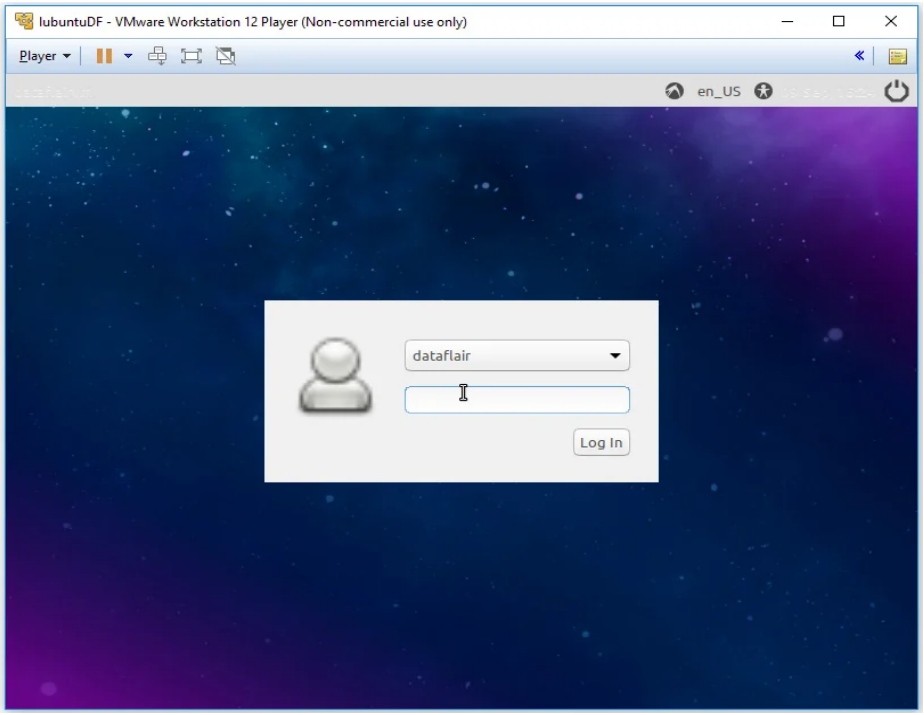

lubuntuDF - VMware Workstation 12 Player (Non-commercial use only)
Player
Installation Complete
Installation is complete. You need to restart the computer in order to use the new installation.
Restart Now

Now to start your Virtual Machine with lubuntu OS, turn the power off and restart your system.

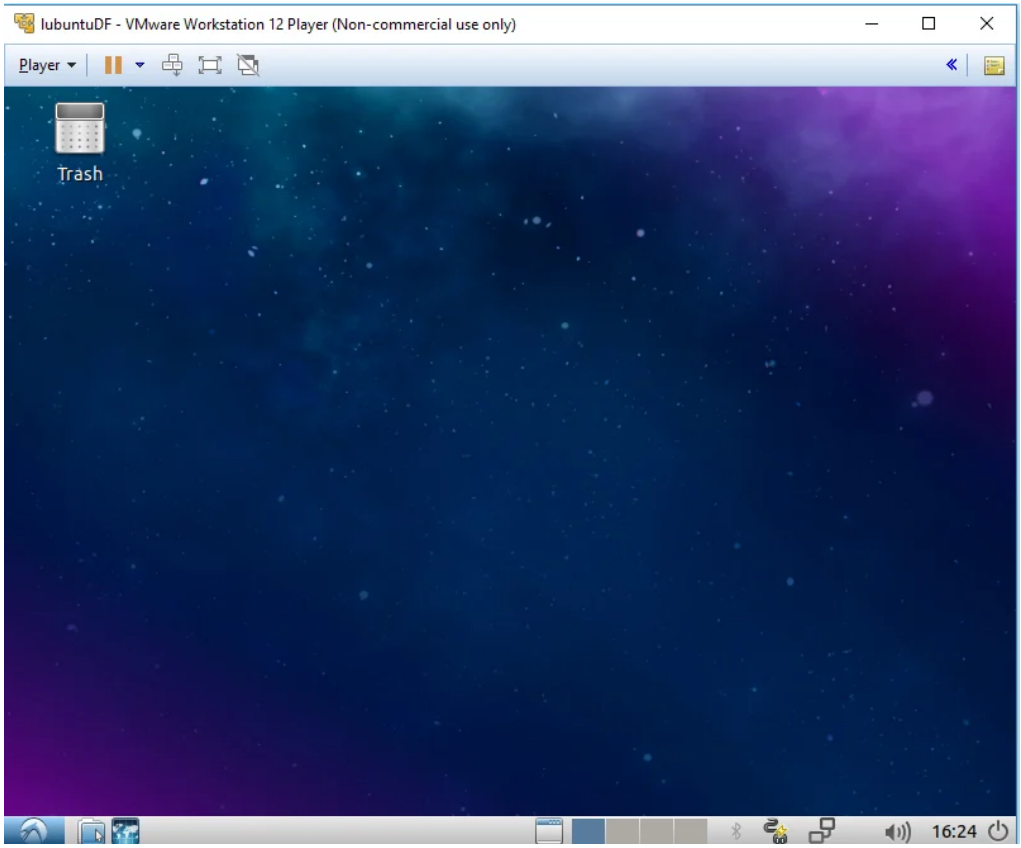


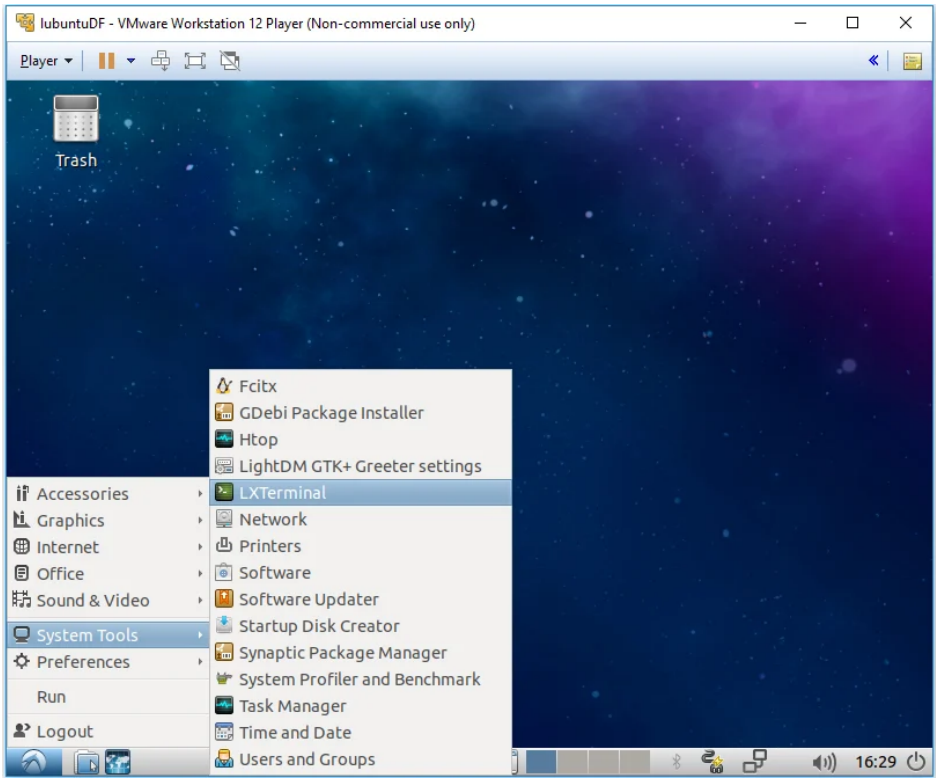

lubuntuDF - VMware Workstation 12 Player (Non-commercial use only)
Player
lubuntu

Log-in into the VM using the username and password that you mentioned in the detail form earlier. Click on the Login button.



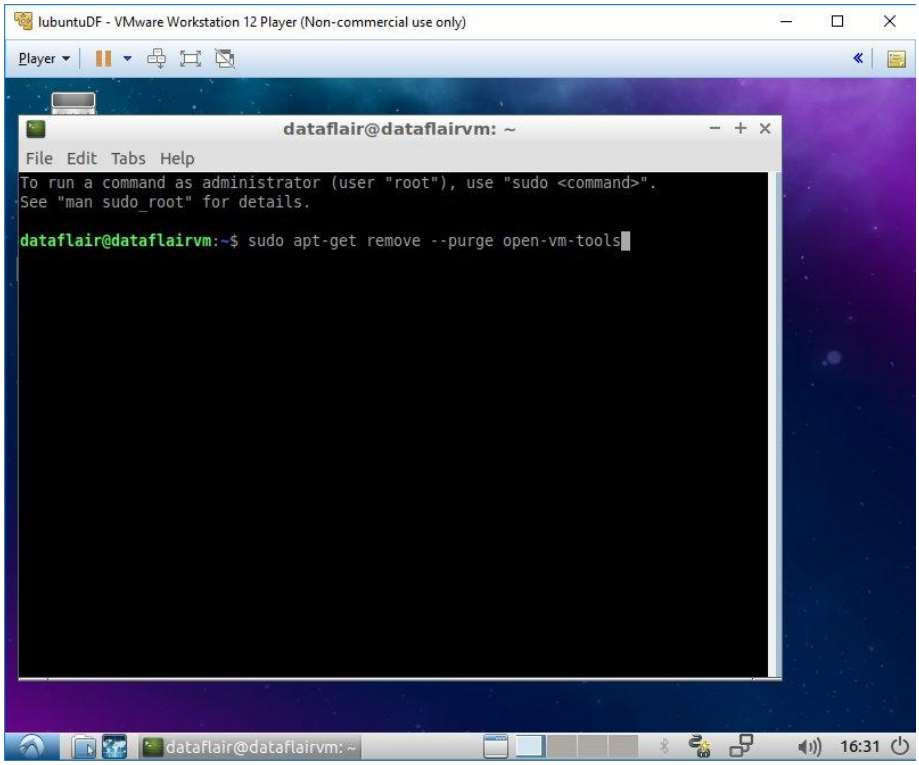
Now to move into the full-screen mode, open a terminal in the Lubuntu OS.



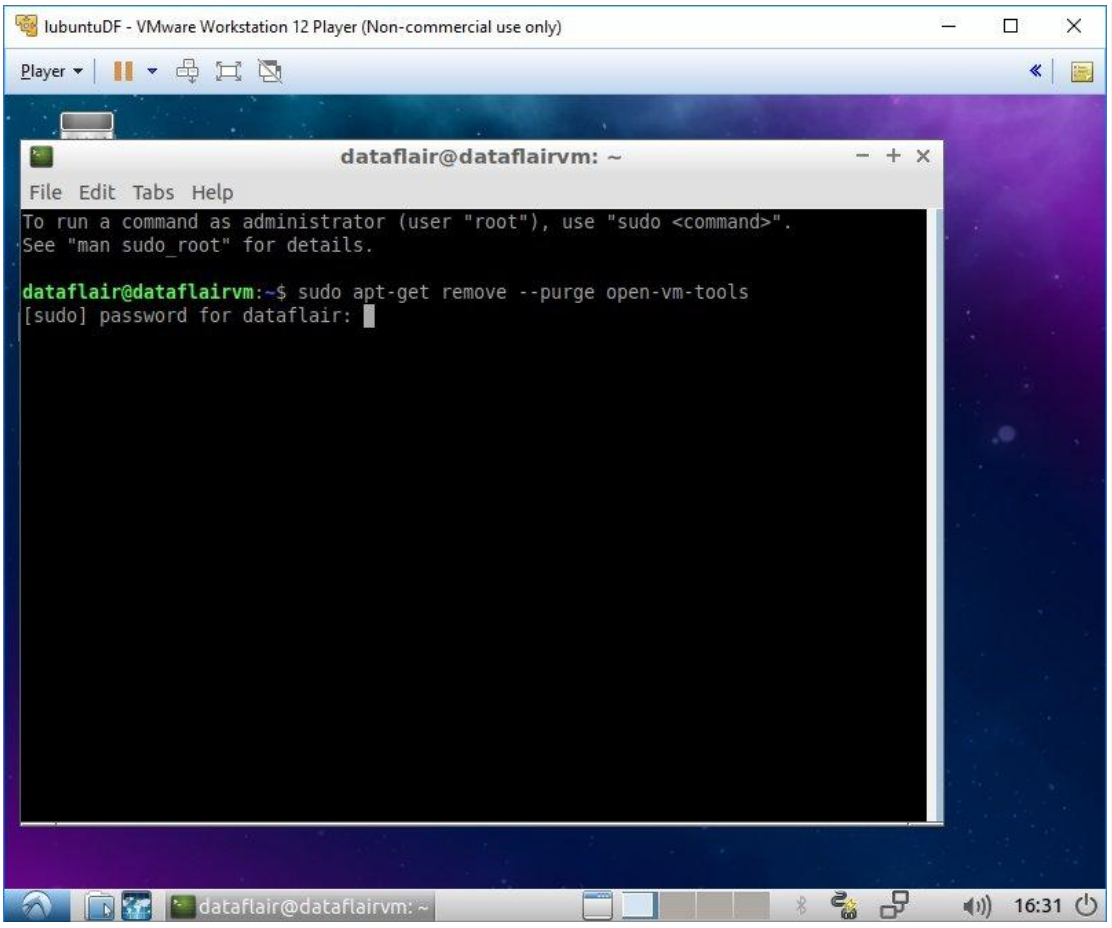


Run the following commands in the terminal and then press Enter:

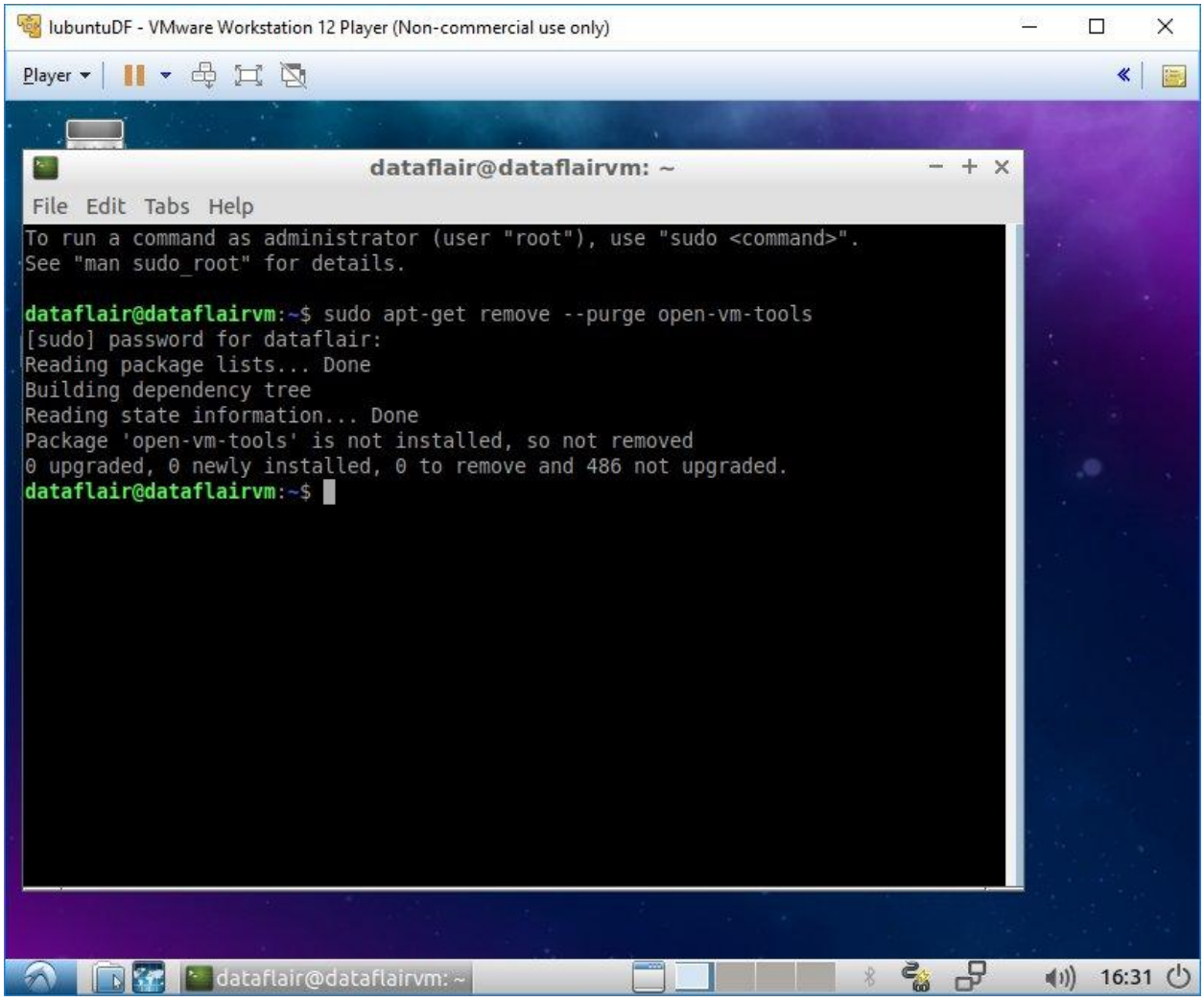
```
sudo apt-get remove --purge open-vm-tools
```



Enter the password for your user and then press Enter:

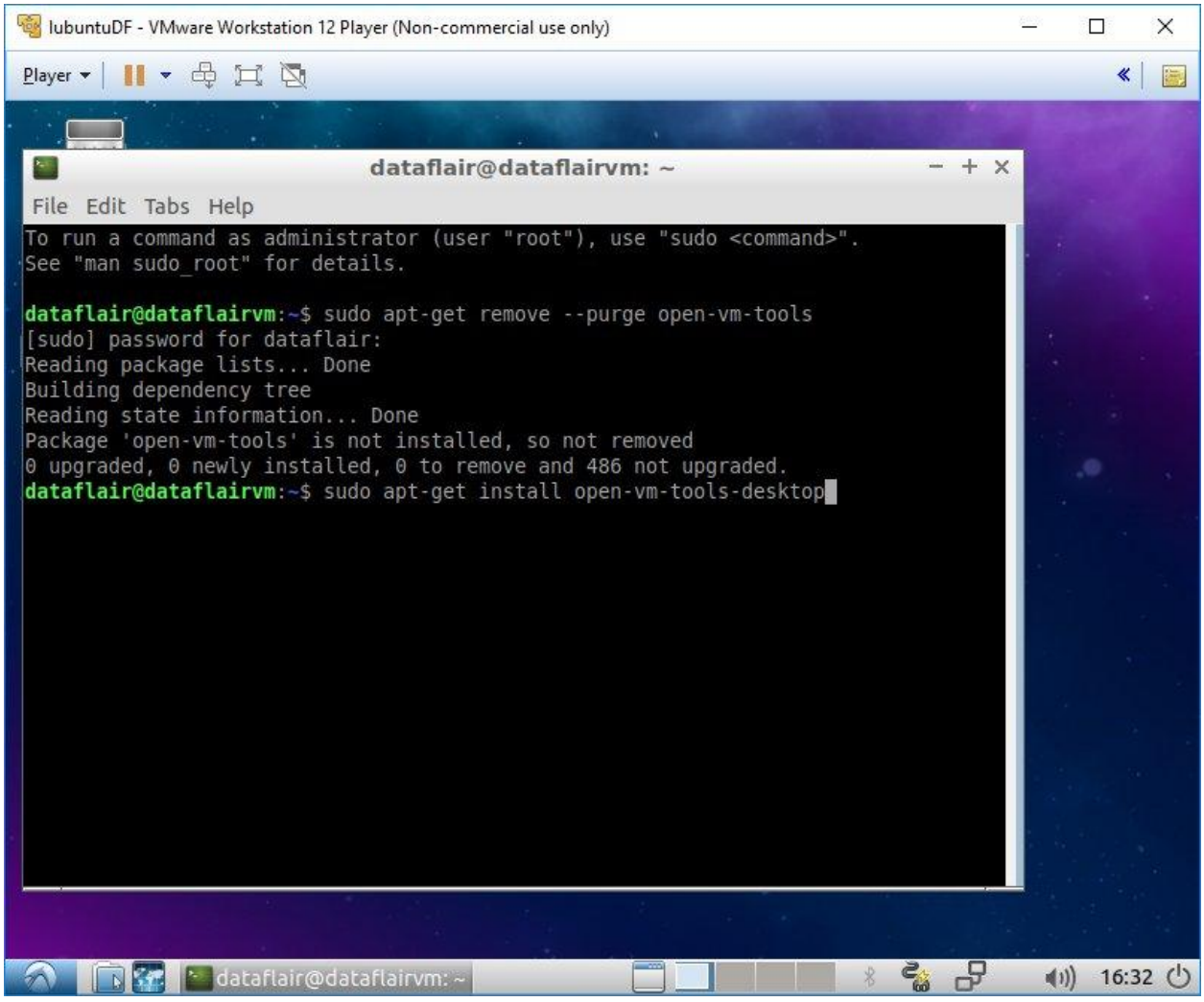


Entering password of user

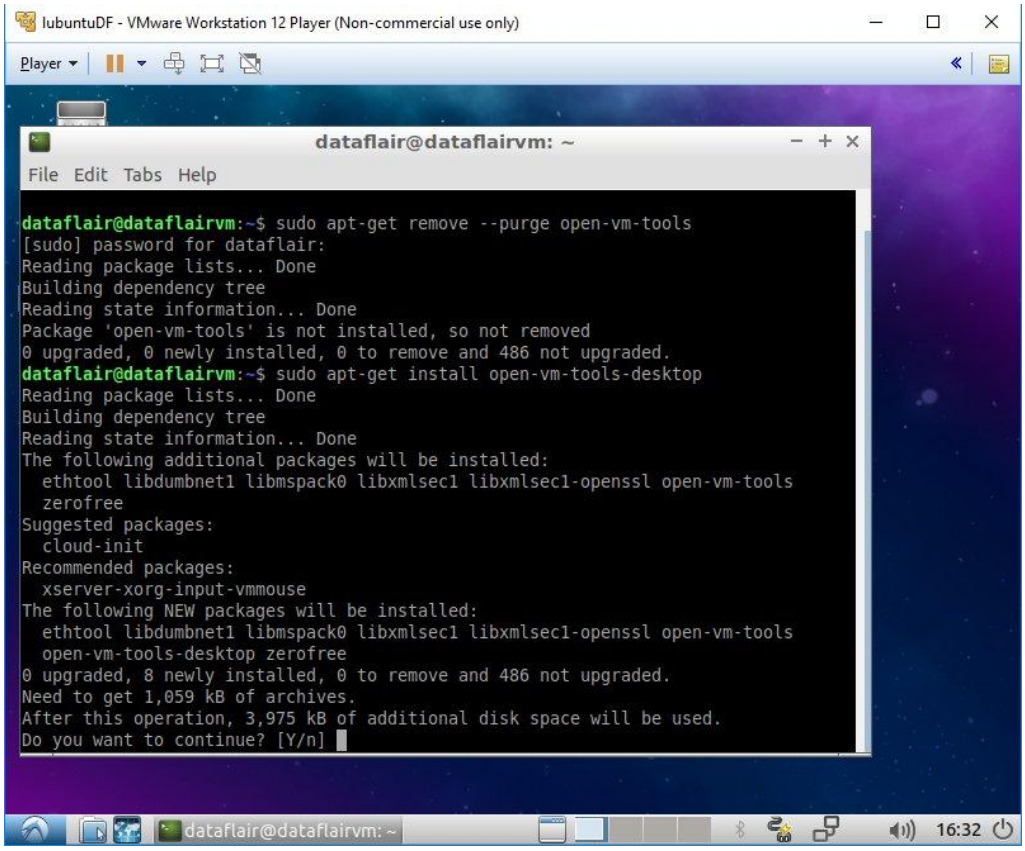


Execution after entering password

```
sudo apt-get install open-vm-tools-desktop
```

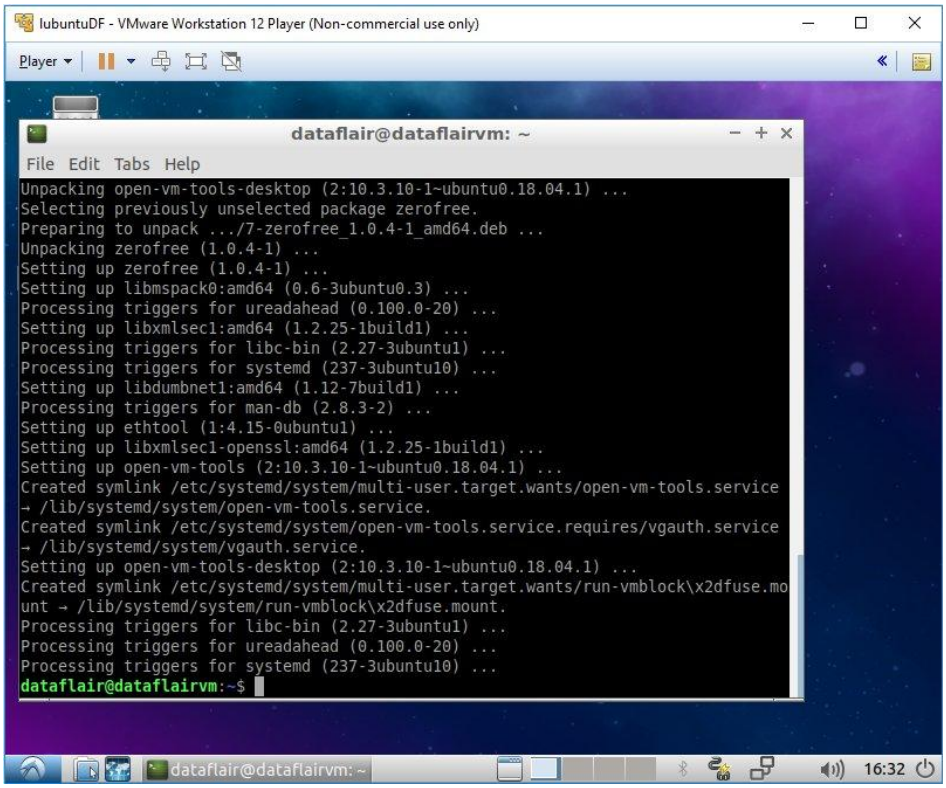


Run get install command

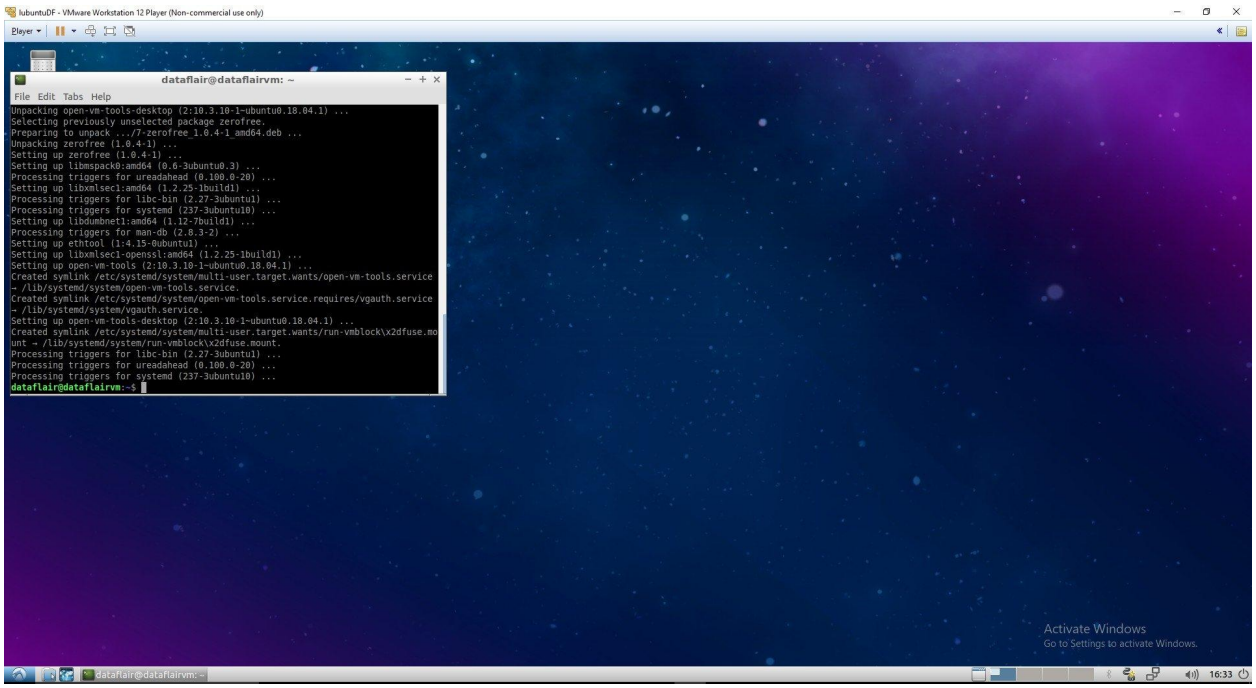


Get the install command running

Type "Y" and then press Enter:



Your OS now shows up on the full screen:



Congratulations!! You have successfully installed Lubuntu OS on the VMware Player for your machine. I hope you are clear on all the steps.

Installing Xampp in Linux

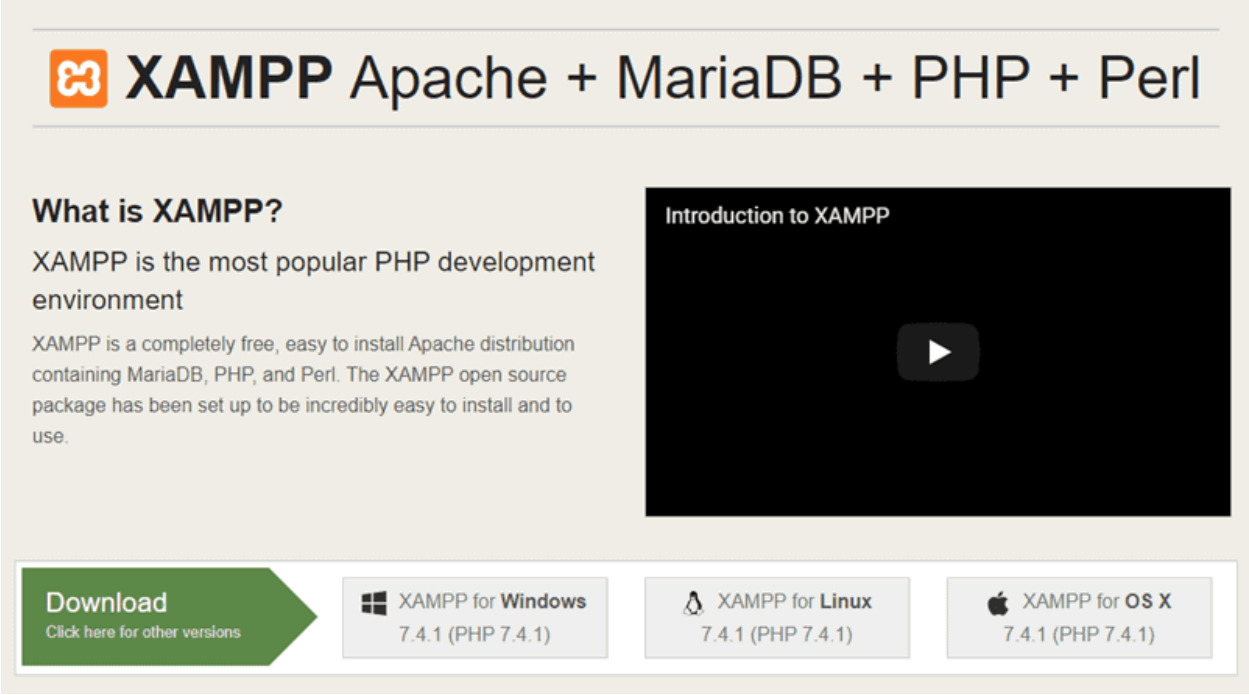
Here we will learn about the process of installing Xampp in Lubuntu. The steps are easy to demonstrate, and necessary screenshots are also provided to guide you throughout the process of installation.

Steps to Installing Xampp in Linux

The following steps are given as follows:

Step 1: Download the Installation Package
Before you can install the XAMPP stack, you need to download the package from the official Apache Friends webpage. Link - https://www.apachefriends.org/download.html

Click the XAMPP for Linux link and save the file.



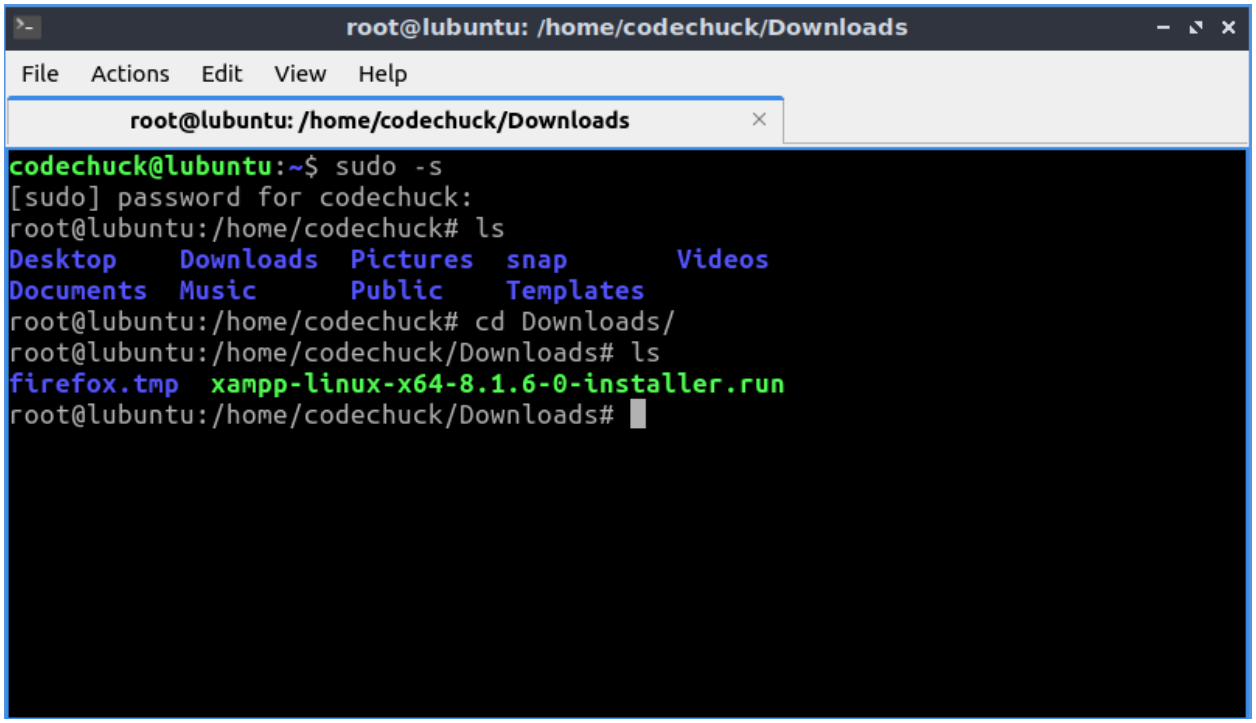
Step 2: Make the installation package executable. To run the installation process, the file permissions require modification.

To execute a program, you need to make the file executable. Open the terminal (Ctrl+Alt+T) and follow these instructions to do so:
Firstly, you will need to get root access to run the file, otherwise Linux won't allow us to run or execute the file. First, to get you or allow you to run a command as root, the command:

```
Sudo -s
```

In Linux, the 'cd' (**Change Directory**) command is one of the most important and most widely used commands for newbies as well as system administrators. After getting root access (to run commands as root), the 'ls' command is used to list files or directories in Linux and other Unix-based operating systems. (Just like you navigate in your File Explorer or Finder with a GUI, the ls command allows you to list all files or directories in the current directory by default, and further interact with them via the command line).

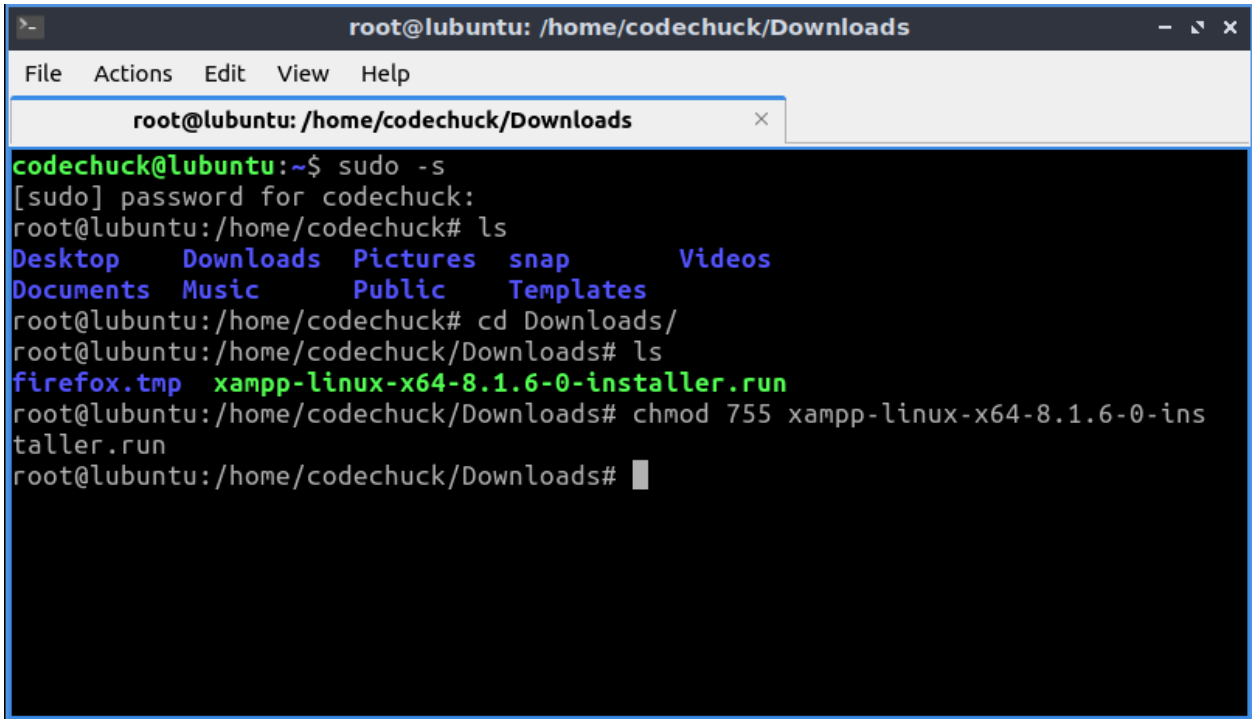
```
Cd Downloads/
```



Step 3: Launching setup wizard

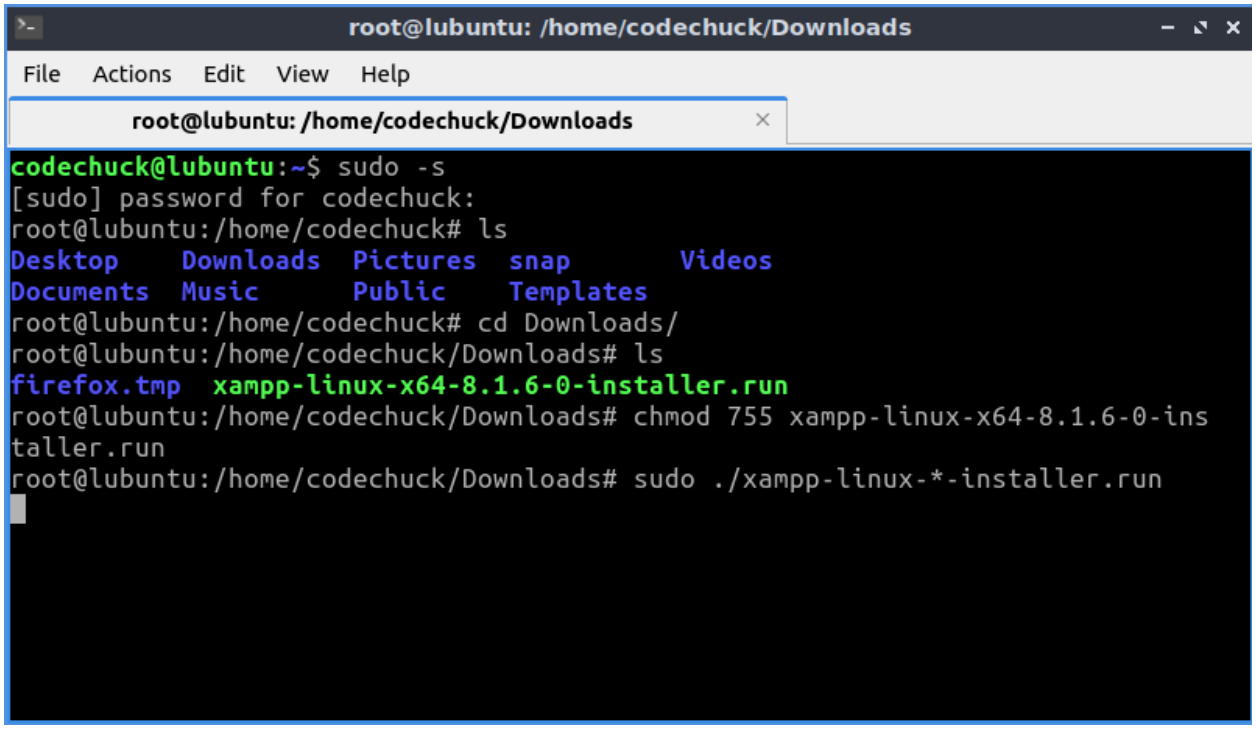
Now we need to change the permission to the installer.

`Chmod 755 xampp-linux-*-installer.run`

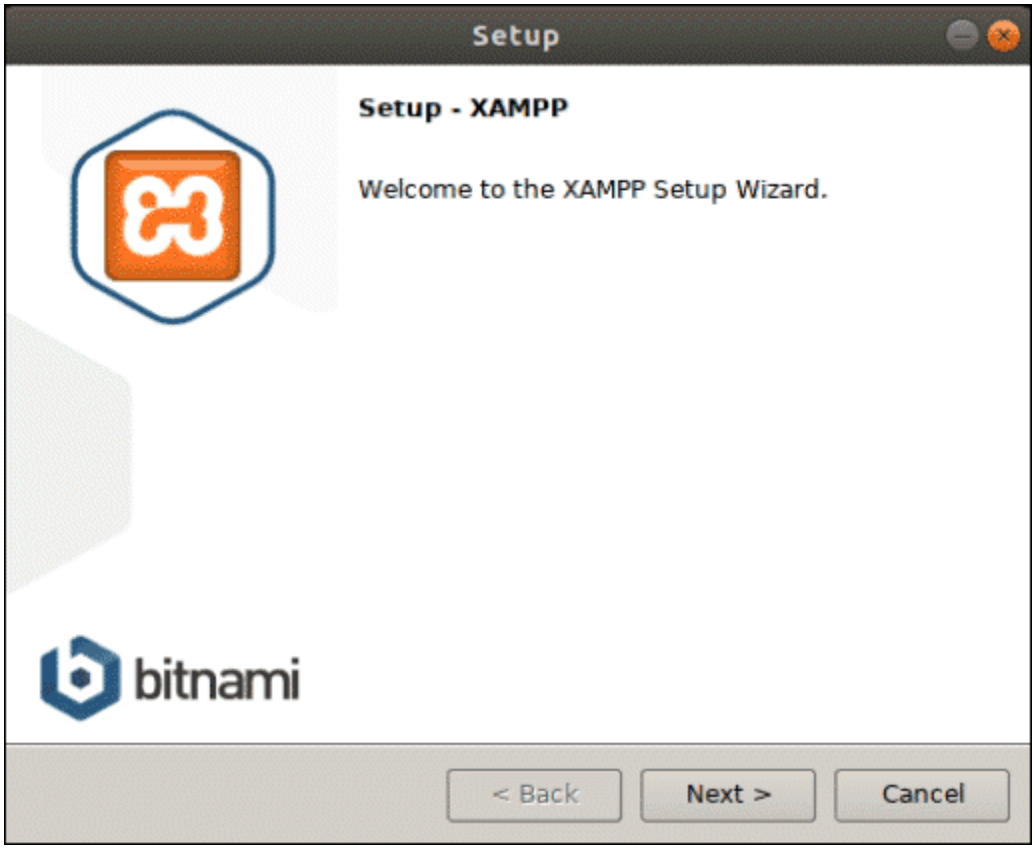


Run the installer

`sudo ./xampp-linux-*-installer.run`

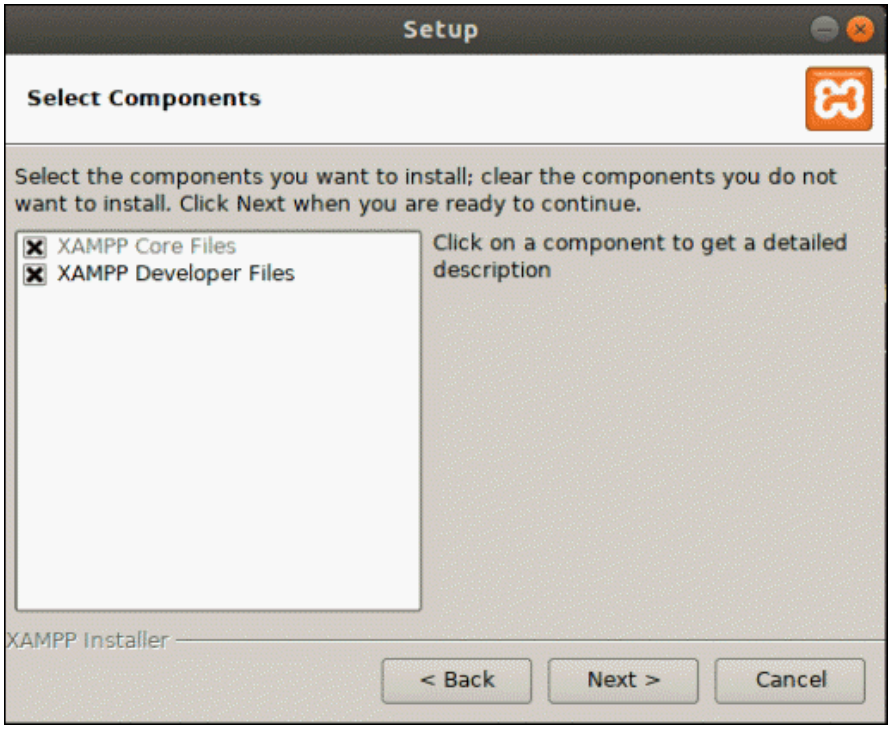


The XAMPP Setup Wizard opens in a new window on top of the terminal that will appear as in the following image:

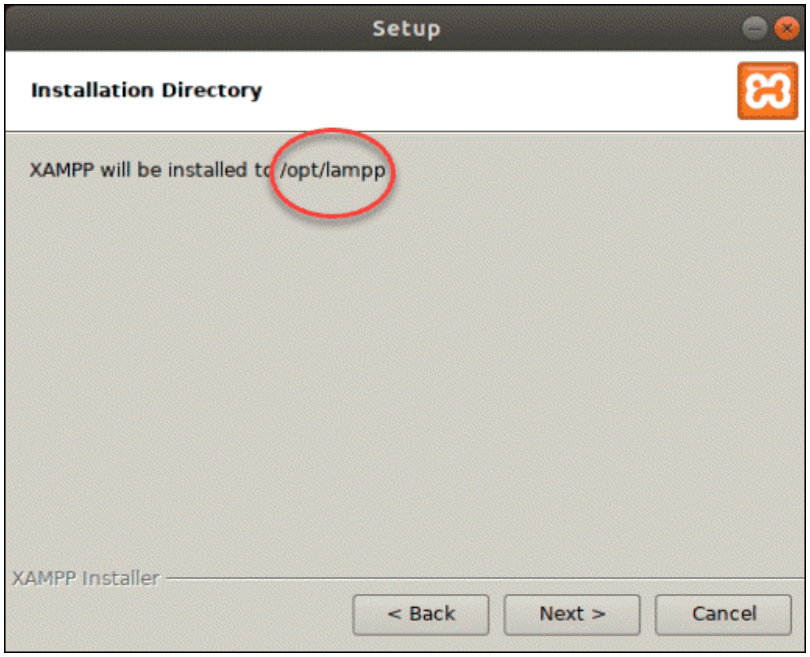


Step 4: Install XAMPP in linux

4.1. Click Next and, in the following Select Components dialogue, choose the components you want to install. We recommend keeping the default settings and continuing with Next.

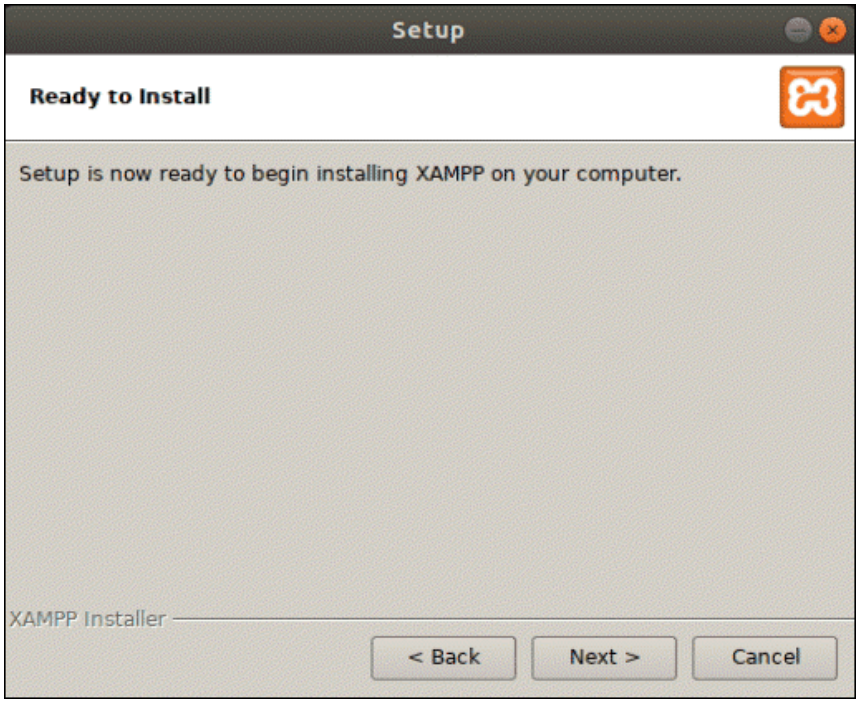


4.2. After selecting the components, the setup wizard will show you the location where it will install the software. To proceed, click Next.



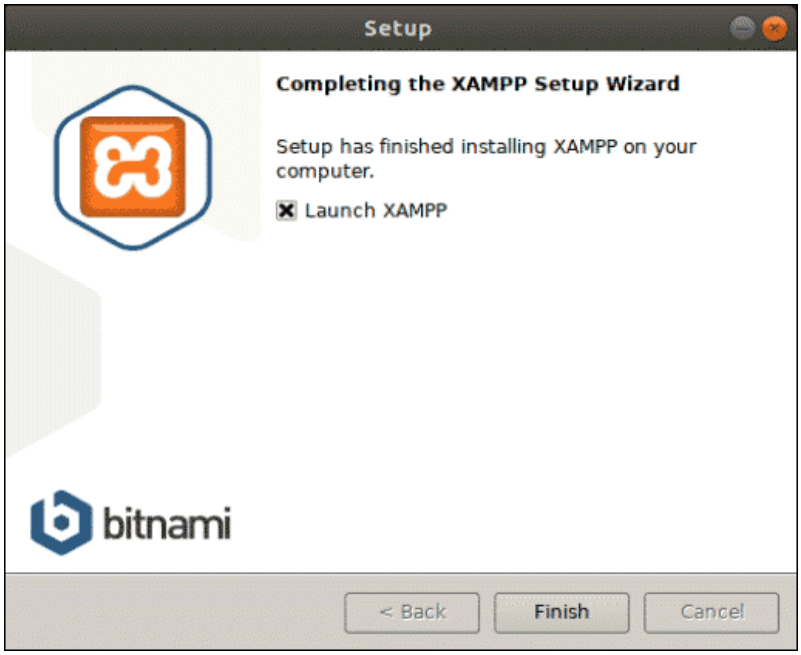
4.3. The following dialogue box offers to install sponsored applications on top of the XAMPP installation. These include packages such as WordPress, Joomla, Drupal, and others. You can deny installing additional software by unchecking the Learn more about Bitnami for XAMPP box.

4.4. Next, the wizard will notify you that it is ready to install XAMPP on your system. Click Next to start.



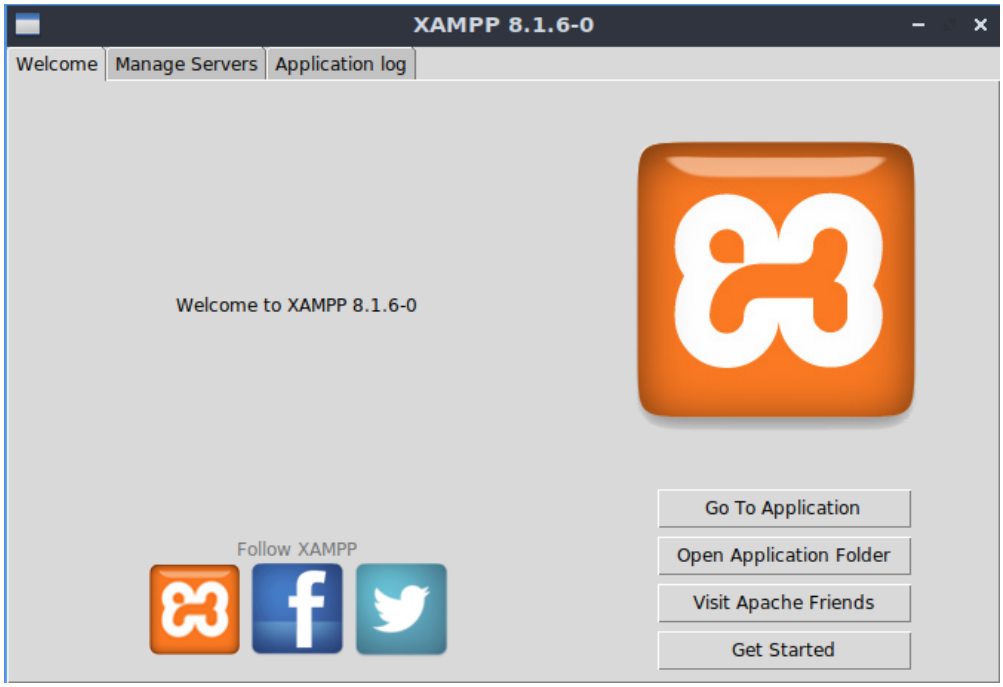
4.5. This will launch the installation process and a dialogue box showing the progress will appear on your screen.

4.6. The final dialogue box should display that the installation has finished installing. You can complete the process and launch XAMPP by clicking Finish.



Step 5: Launch XAMPP

By clicking Finish in the previous step, XAMPP launches its control panel, which will appear as in the image below.

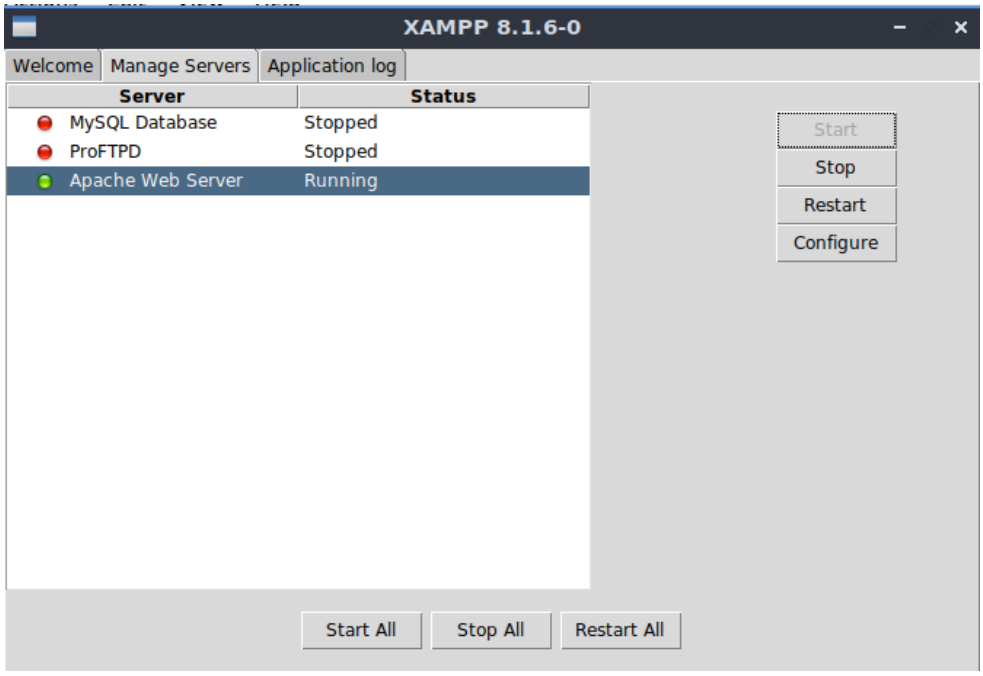


In case you forget to 'Launch Xampp' after the installer, you can simply follow the commands to open Xampp.

`sudo -s` (to run a command as root)

```
/opt/lampp/manager-linux-x64.run
```

Open the Manage Servers tab to see all the available services and their status. You can change their status by selecting Start or Stop.

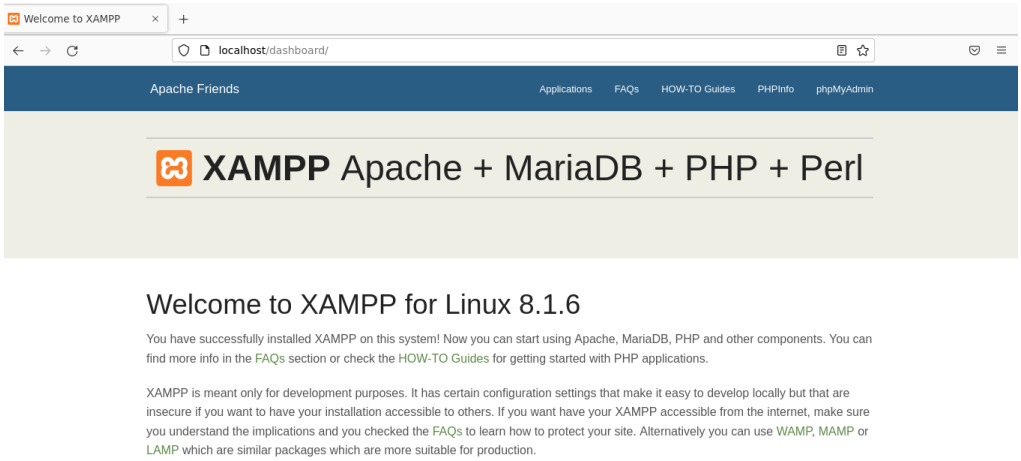


Step 6: Verify XAMPP is Running
Make sure you have successfully installed the XAMPP stack and that everything is running smoothly.

1. Verify that localhost is working by entering the following URL in a browser:

http://localhost/dashboard (or use the IP address of your machine)

If the XAMPP dashboard page displays as in the image below, you have successfully installed the stack.



Using OpenSSL to Generate self-Signed Certificate

Here we will learn about the process of using OpenSSL in Lubuntu to generate self-signed certificates. The steps are easy to demonstrate, and necessary screenshots are also provided to guide you throughout the process of installation.

Steps to Installing OpenSSL on Linux

For this, we need root authority (root will verify the intermediate authority). For which we will create certificates. In this case, we need root, intermediate, and server to get the work done. For the required task, we can use OpenSSL (almost everyone uses this). The following steps are given as follows:

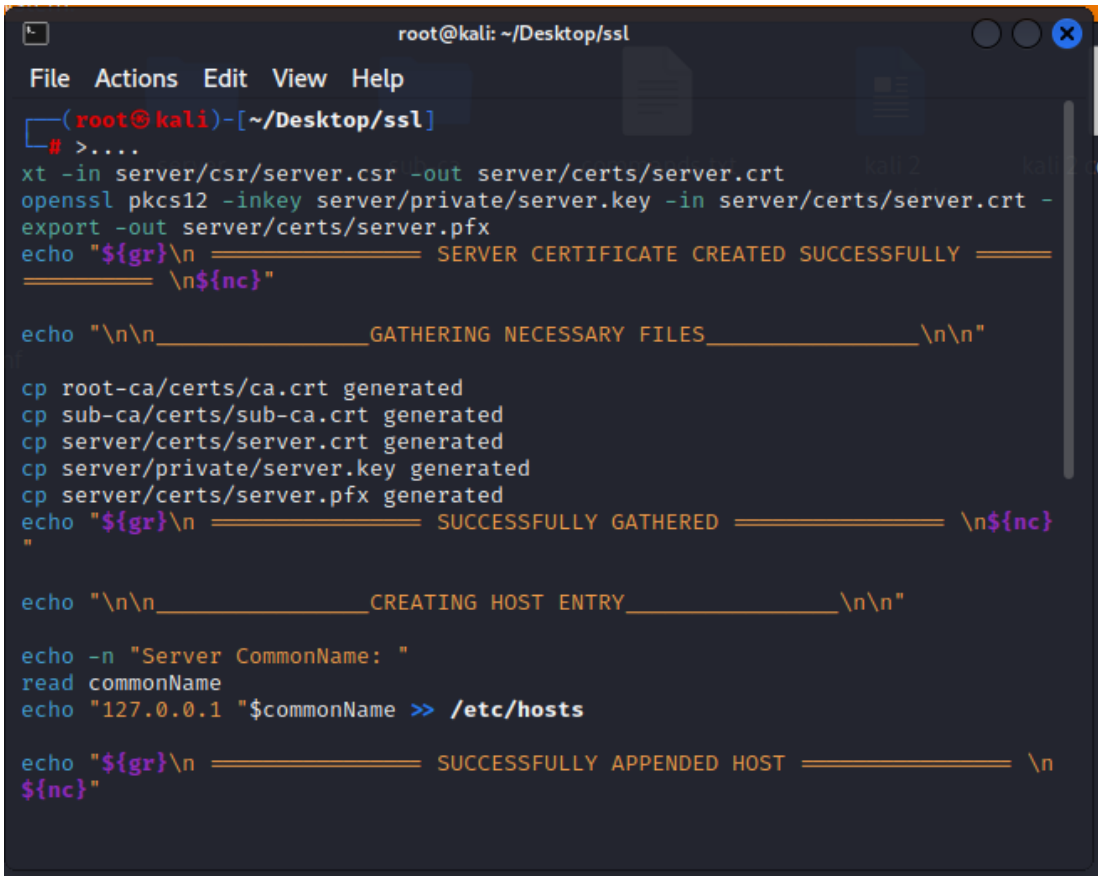
Step 1: Installation commands for OpenSSL are given below.

```
echo "\n\n_________________GENERATING ALL
DIRECTORIES_________________\n\n"
gr='\033[1;32m'
nc='\033[0m' # No Color

mkdir -p
{root-ca,sub-ca,server}/{private,certs,index,serial,pem,crl,csr}
mkdir generated
touch root-ca/index/index
touch sub-ca/index/index
openssl rand -hex 16 > root-ca/serial/serial
openssl rand -hex 16 > sub-ca/serial/serial
cp root-ca.conf root-ca
cp sub-ca.conf sub-ca
echo "${gr}\n ================ FOLDERS CREATED SUCCESSFULLY
================ \n${nc}"

echo "\n\n_________________GENERATING ALL THE
KEYS_________________\n\n"

openssl genrsa -aes256 -out root-ca/private/ca.key 4096
openssl genrsa -aes256 -out sub-ca/private/sub-ca.key 4096
openssl genrsa -out server/private/server.key 2048
echo "${gr}\n ================ KEYS CREATED SUCCESSFULLY
================ \n${nc}"

echo "\n\n_________________GENERATING ROOT
CERTIFICATE_________________\n\n"
```

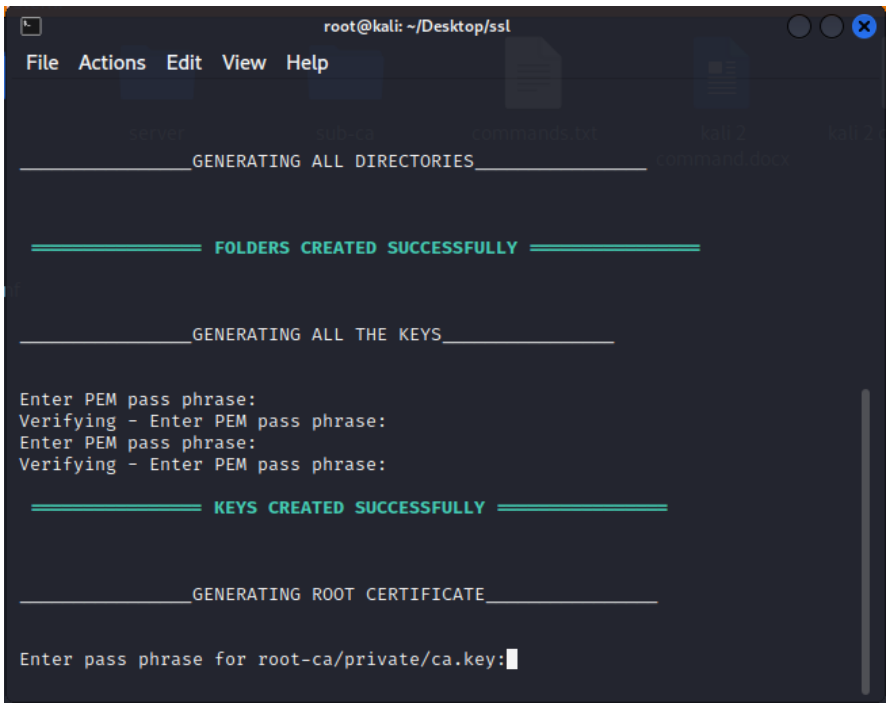
```
openssl req -config root-ca/root-ca.conf -key root-ca/private/ca.key
-new -x509 -days 7305 -sha256 -extensions v3_ca -out
root-ca/certs/ca.crt
echo "${gr}\n ================ ROOT CERTIFICATE CREATED SUCCESSFULLY
================ \n${nc}"

echo "\n\n_________________GENERATING SUB-ROOT
REQUEST_________________\n\n"

openssl req -config sub-ca/sub-ca.conf -new -key
sub-ca/private/sub-ca.key -sha256 -out sub-ca/csr/sub-ca.csr
echo "${gr}\n ================ SUB-ROOT REQUEST CREATED SUCCESSFULLY
================ \n${nc}"

echo "\n\n_________________GENERATING SUB-ROOT
CERTIFICATE_________________\n\n"

openssl ca -config root-ca/root-ca.conf -extensions
v3_intermediate_ca -days 3652 -notext -in sub-ca/csr/sub-ca.csr -out
sub-ca/certs/sub-ca.crt
echo "${gr}\n ================ SUB-ROOT CERTIFICATE CREATED
SUCCESSFULLY ================ \n${nc}"

echo "\n\n_________________GENERATING SERVER
REQUEST_________________\n\n"

openssl req -key server/private/server.key -new -sha256 -out
server/csr/server.csr
echo "${gr}\n ================ SERVER REQUEST CREATED SUCCESSFULLY
================ \n${nc}"

echo "\n\n_________________GENERATING SERVER
CERTIFICATE_________________\n\n"

openssl ca -config sub-ca/sub-ca.conf -extensions server_cert -days
365 -notext -in server/csr/server.csr -out server/certs/server.crt
openssl pkcs12 -inkey server/private/server.key -in
server/certs/server.crt -export -out server/certs/server.pfx
echo "${gr}\n ================ SERVER CERTIFICATE CREATED
SUCCESSFULLY ================ \n${nc}"

echo "\n\n_________________GATHERING NECESSARY
FILES_________________\n\n"
```

```
cp root-ca/certs/ca.crt generated
cp sub-ca/certs/sub-ca.crt generated
cp server/certs/server.crt generated
cp server/private/server.key generated
cp server/certs/server.pfx generated
echo "${gr}\n ================ SUCCESSFULLY GATHERED ================
\n${nc}"

echo "\n\n_________________CREATING HOST ENTRY_________________\n\n"

echo -n "Server CommonName: "
read commonName
echo "127.0.0.1 "$commonName >> /etc/hosts

echo "${gr}\n ================ SUCCESSFULLY APPENDED HOST
================ \n${nc}"
```

Step 2: After the installation of OpenSSL, we need to open a terminal inside the designated folder where all the necessary certificates will be stored. ed.

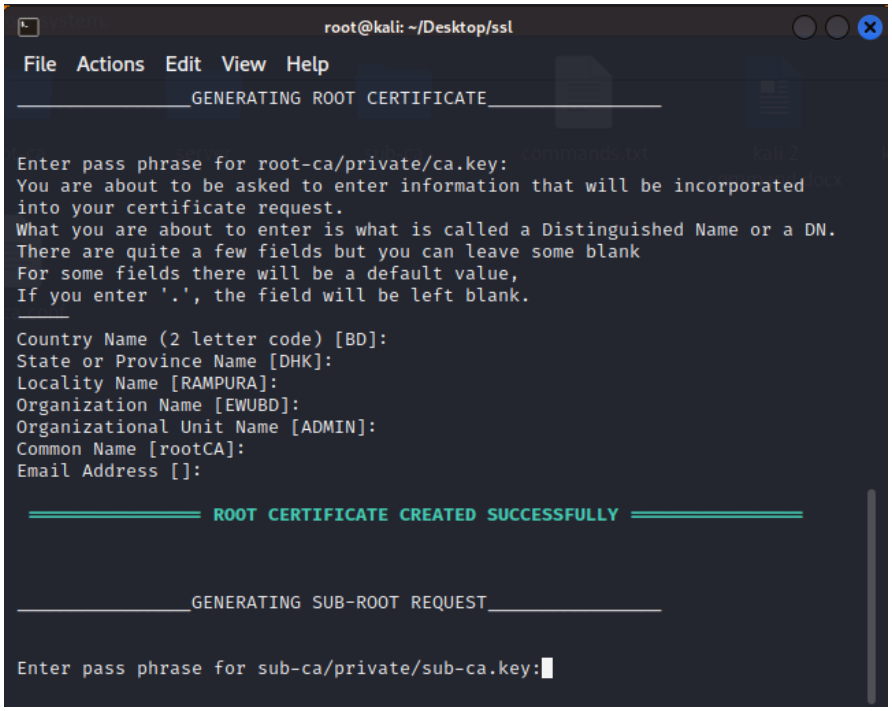


Step 3: There will be a message pop up inside the terminal showing us creating the “Folders created successfully” message where the certificates will be stored (their information along with their serial numbers, keys etc.).

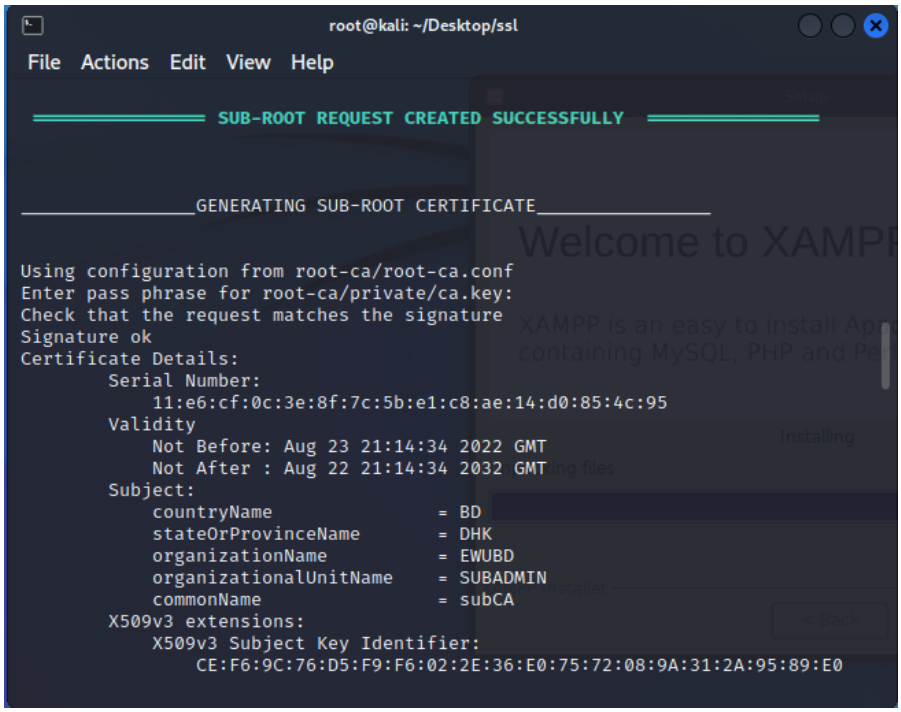


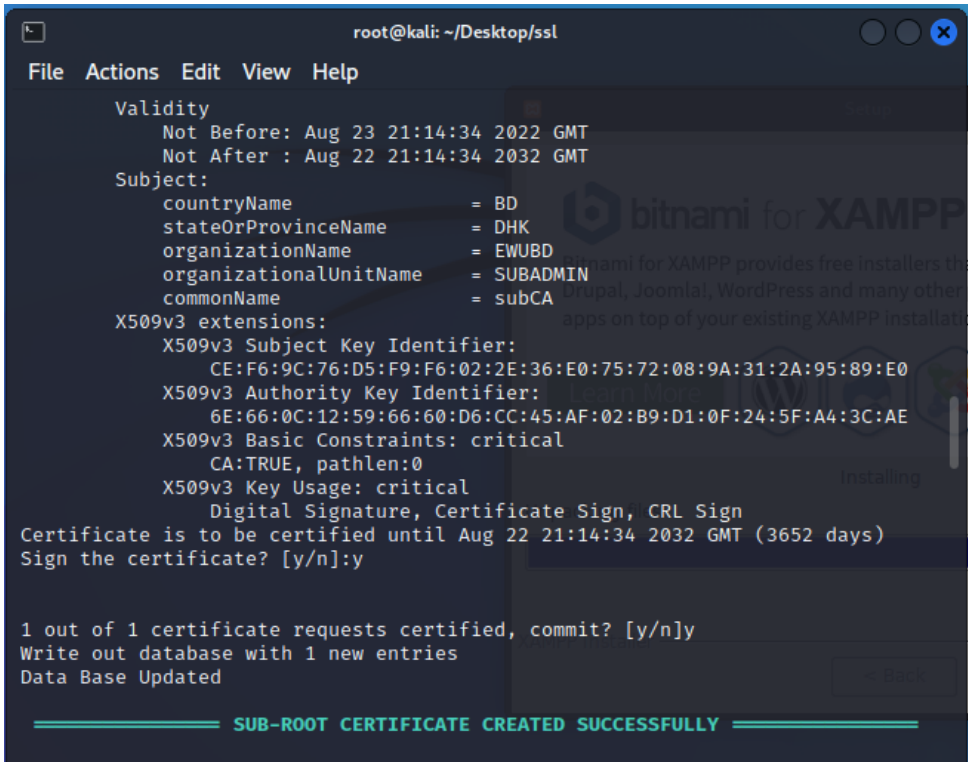
N.B: While entering the "PEM pass phrase”, keep in mind that the password should be the root password (for the sake of simplicity, we are considering it as the "root" password and it will be given after the installation processes in this certificate generation.

Step 4: Generating the root certificate (there will be predefined information provided, but we can also edit all the necessary information when needed. (while using the pass phrase as "root").

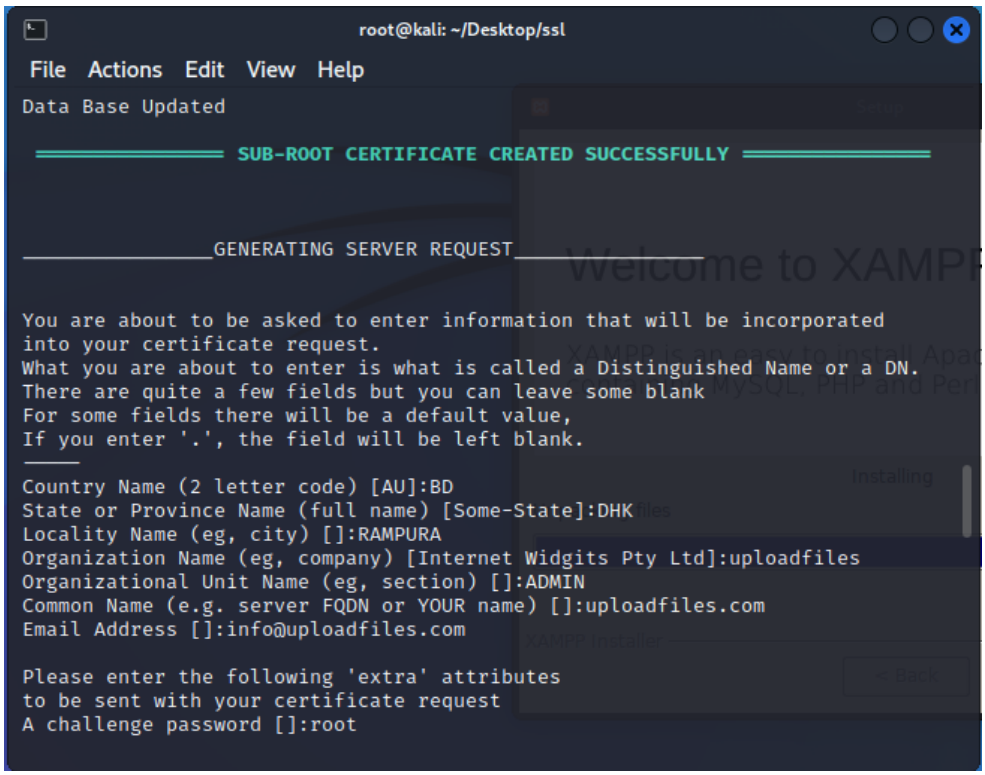


Step 5: After generating the sub-root certificate (there will be predefined information provided, but we can also edit all the necessary information when needed. (while using the pass phrase as 'root'). There will e preview which is given below -

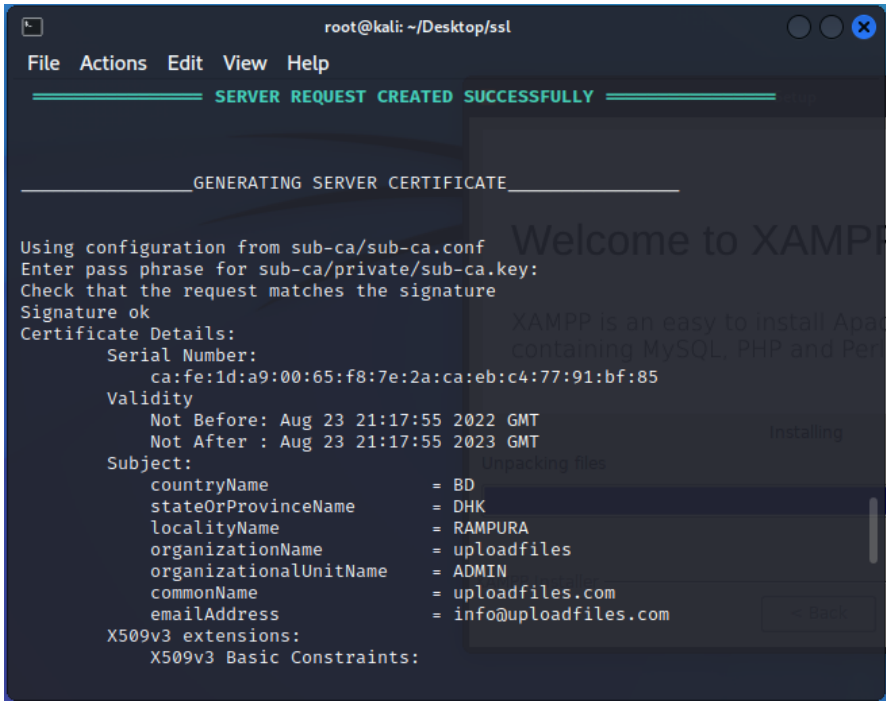




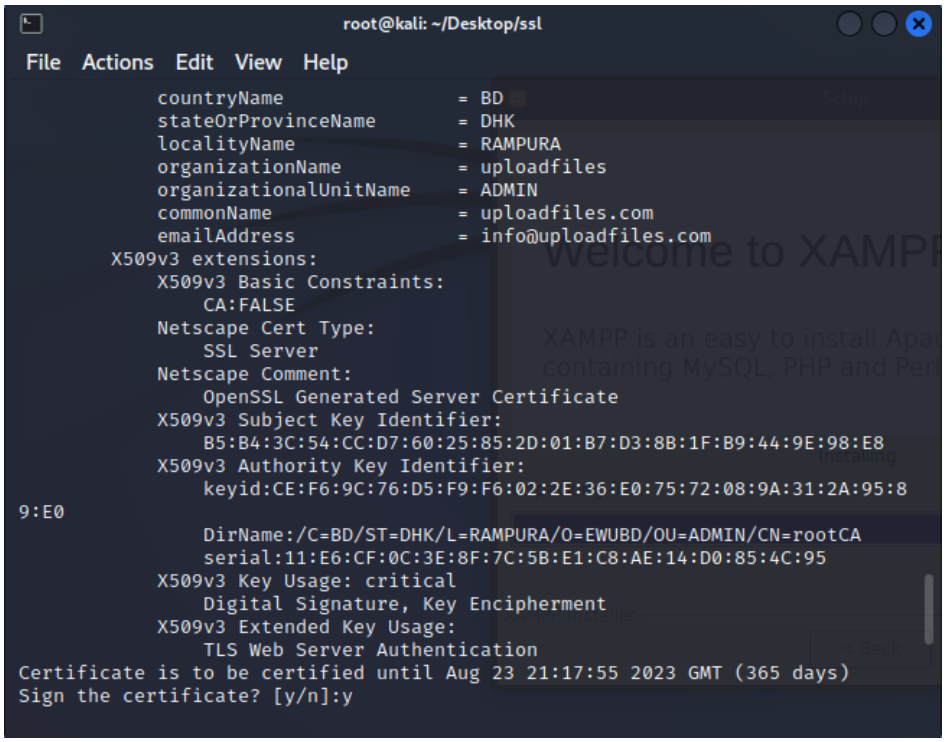
Step 6: Generating a server request. While generating the request, we must keep in mind that the password or pass phrase must be the same (in both cases).

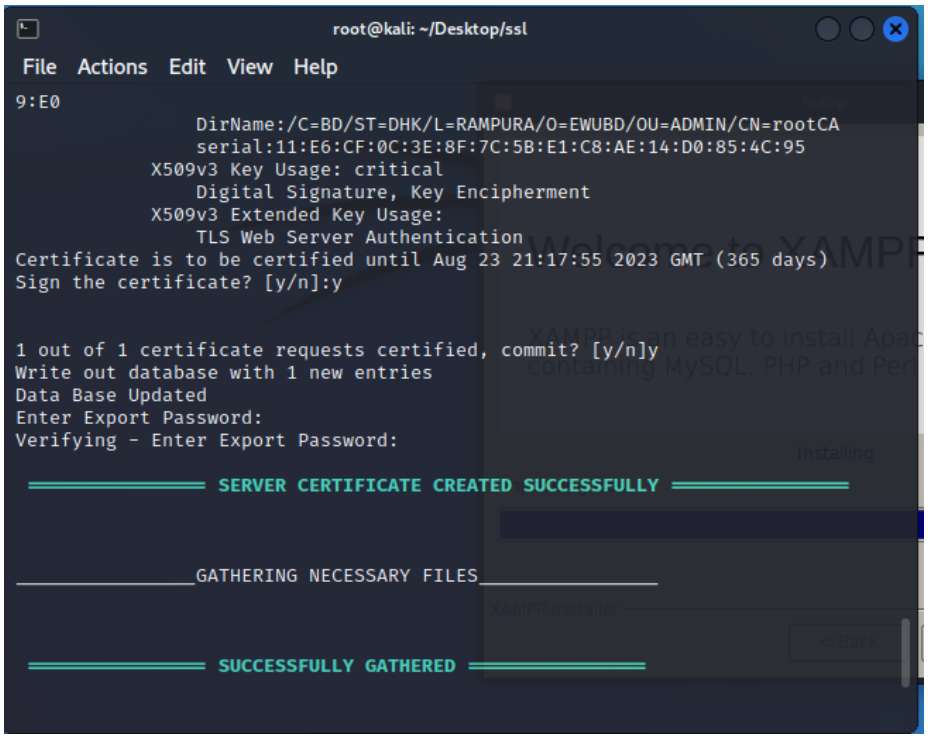


Step 7: Completing the server request Now the server certificate will start the process (there will be an overview of the details).

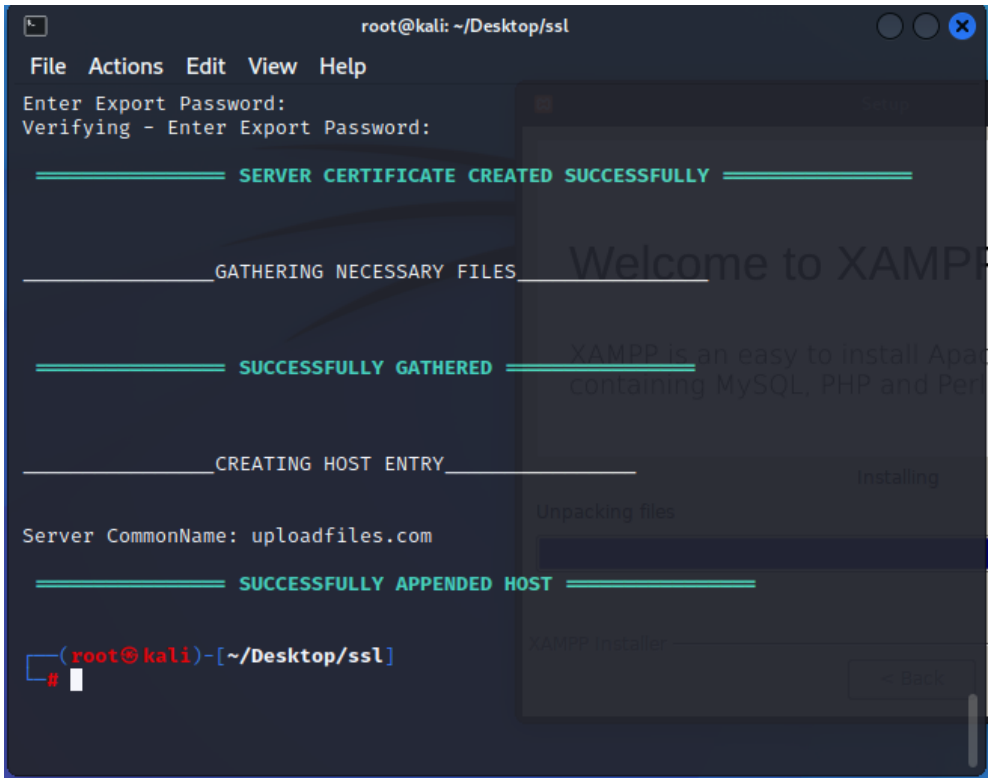


Step 8: Giving the server certificate the signing request



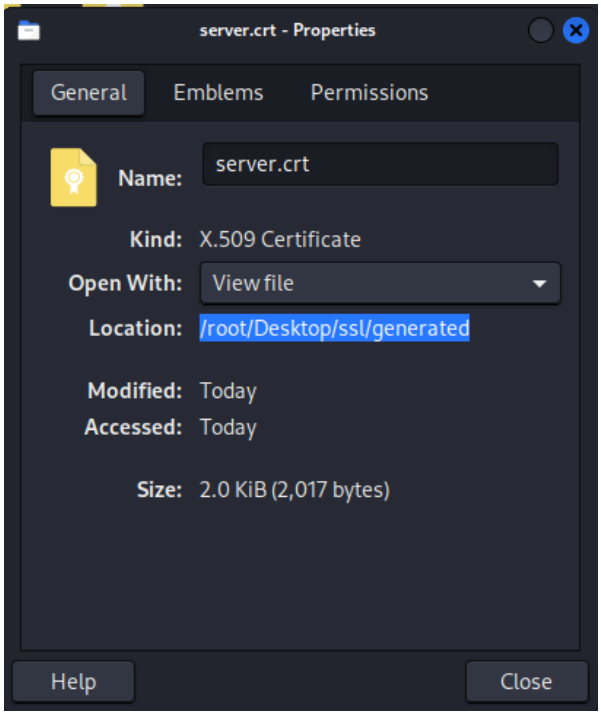


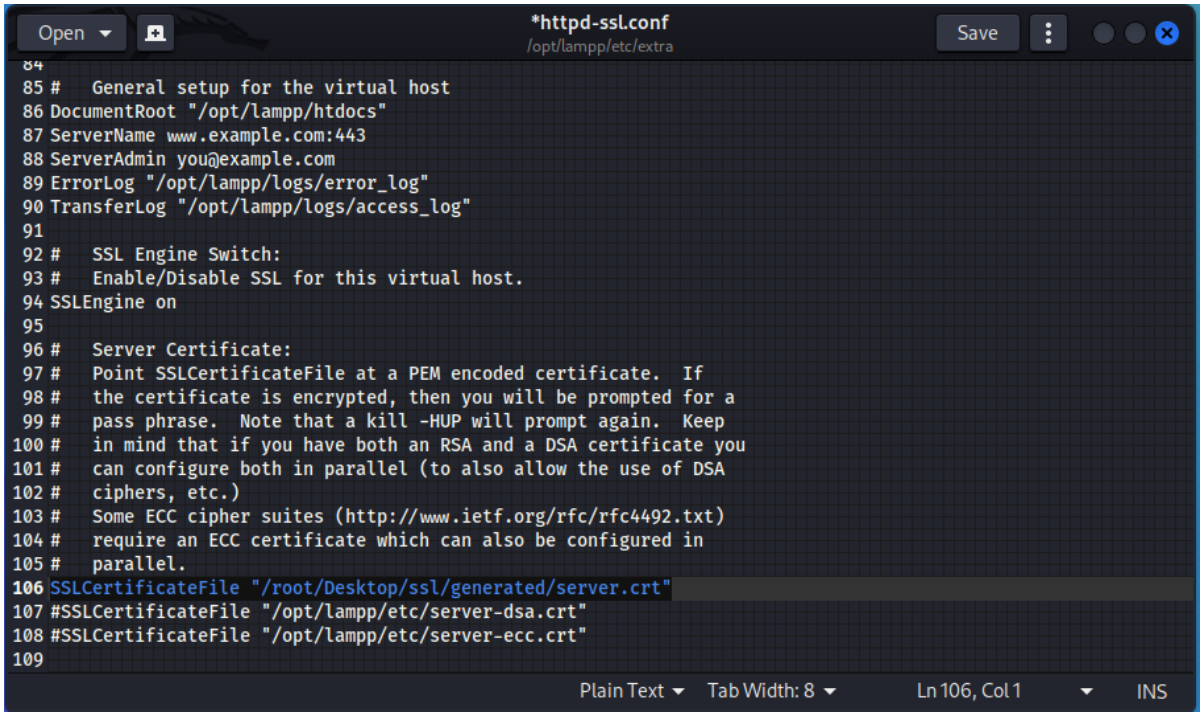
Step 9: All the necessary files will be generated in "generated" folder using the server common name as uploadfiles.com



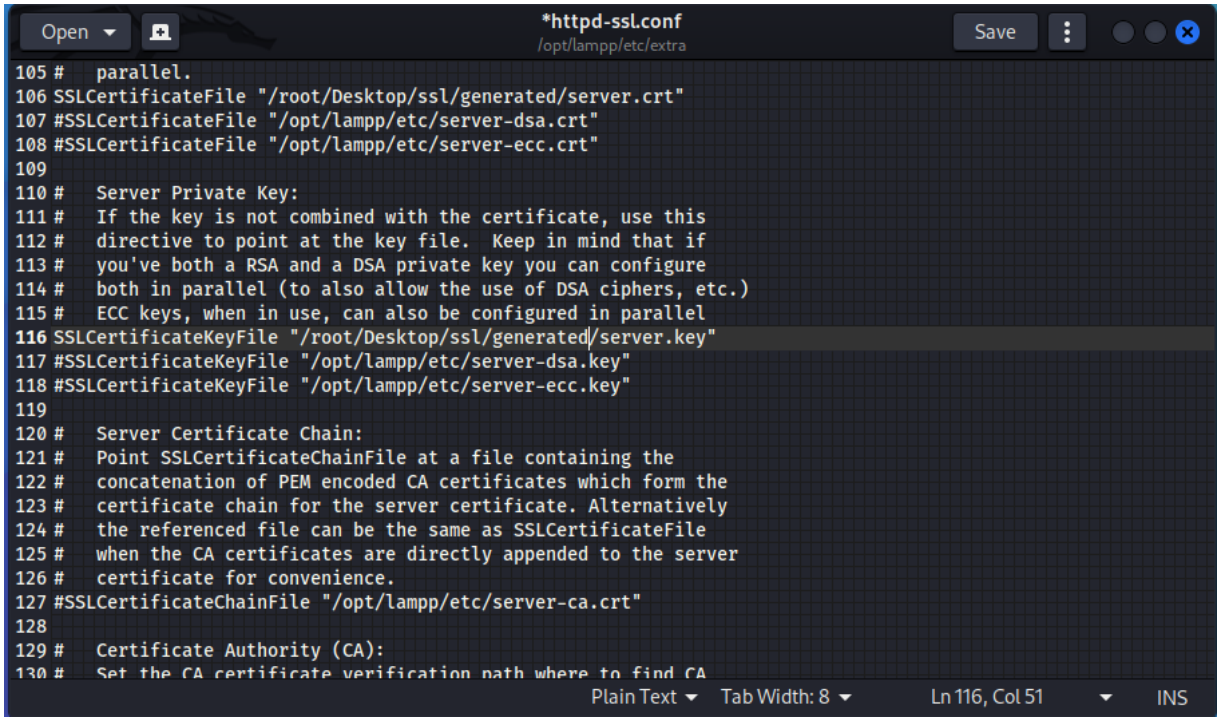
Step 10: We need the location of the SSL certificate file. For that we can put our newly generated server.crt location (because we are going to use it for the demonstration purpose)

N.B: using gedit to open the .conf file. If your Linux machine doesn't contain "gedit" then install using terminal by "sudo apt install gedit")

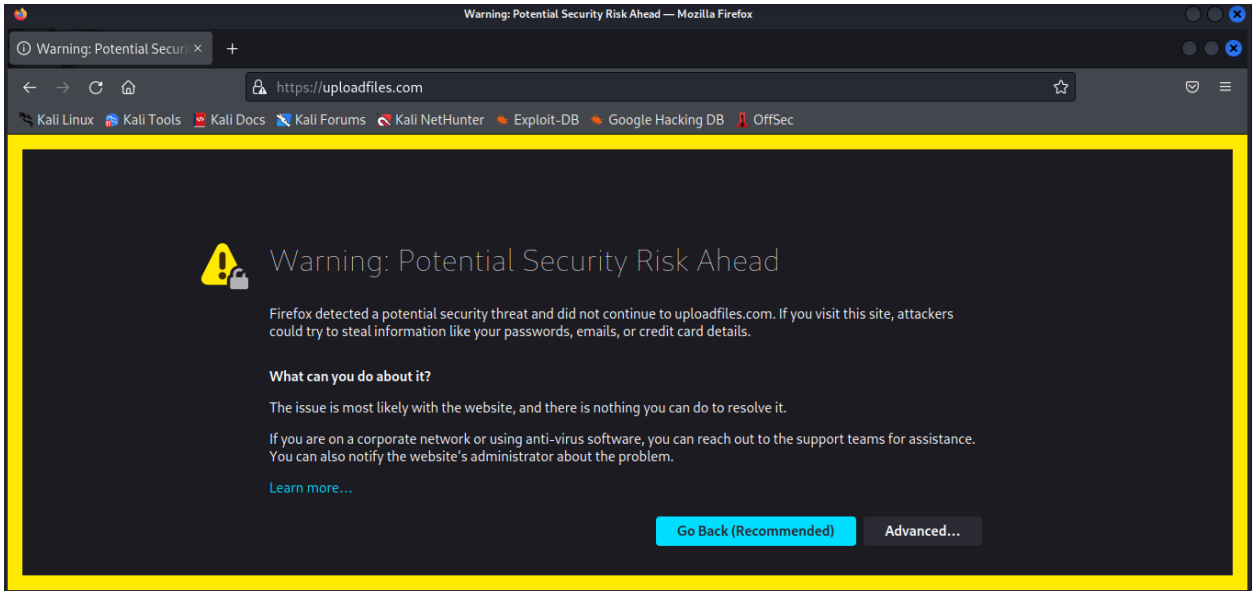




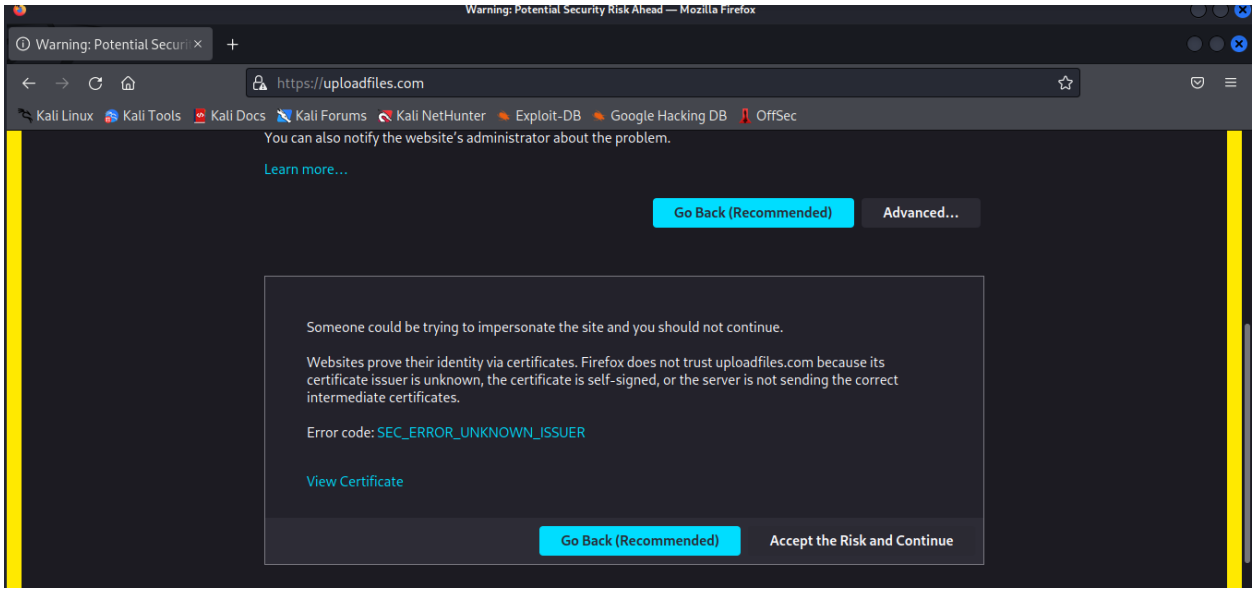
Step 11: We will also change the location of ssl certificate key file (line 116)



Step 12: After the previous step, when we search for the designated domain name in a browser, we can see that the particular website isn't secure to surf.

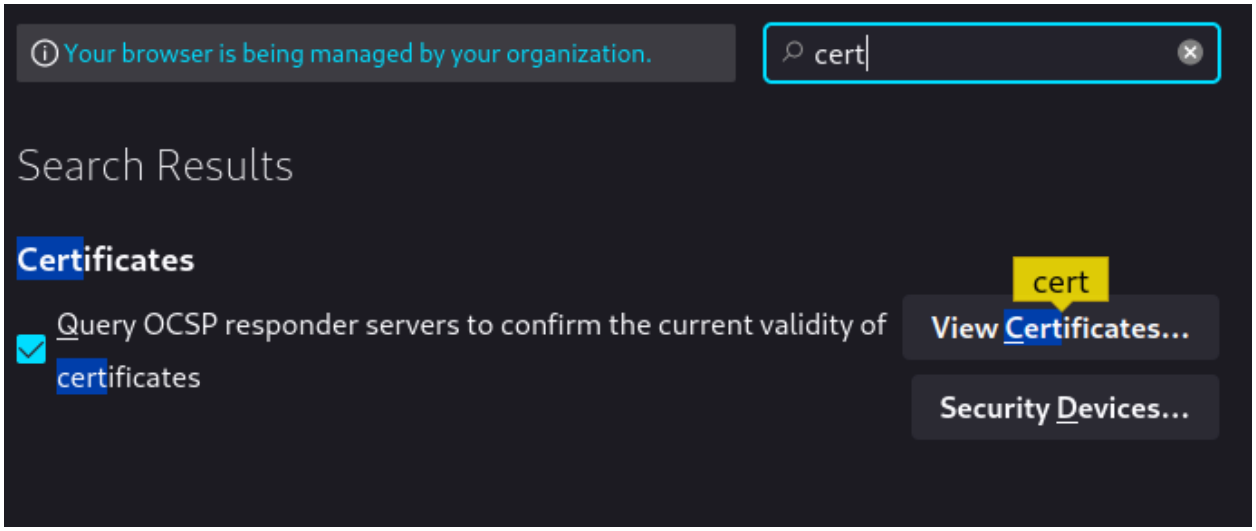


Click on the 'Advanced…' to see the details behind the error.

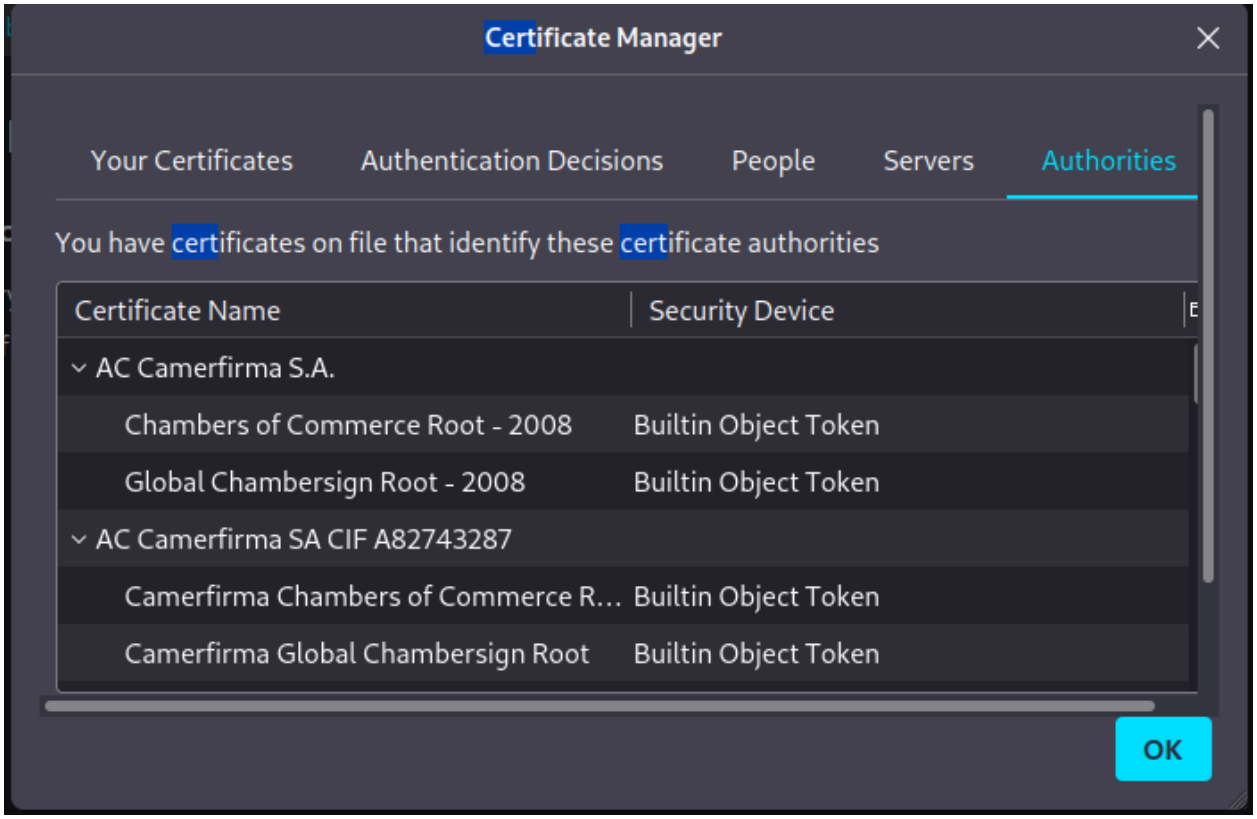


Step 13: The reason behind the error is that the browser is not trusted or unable to recognize the domain name "https://uploadfiles.com'. So, we need to upload our self-signed certificates into the browser.

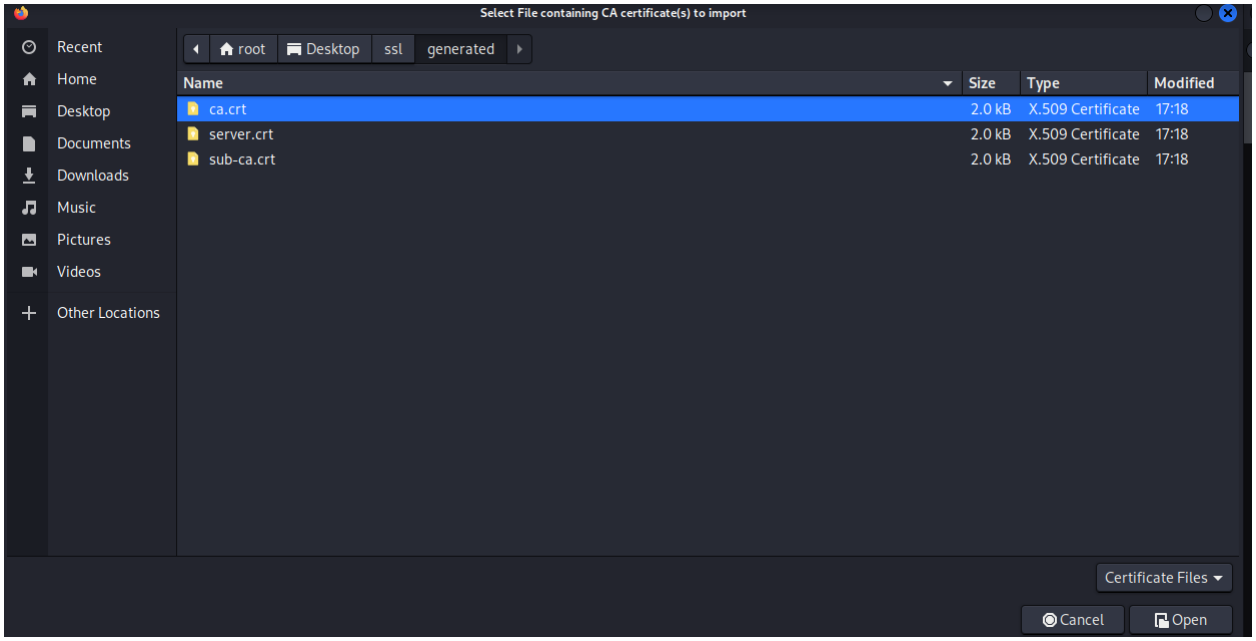
In settings, we can search for "cert" to find out the list of certificates where we can upload our self-signed certificates and make the browser trust our domain name.



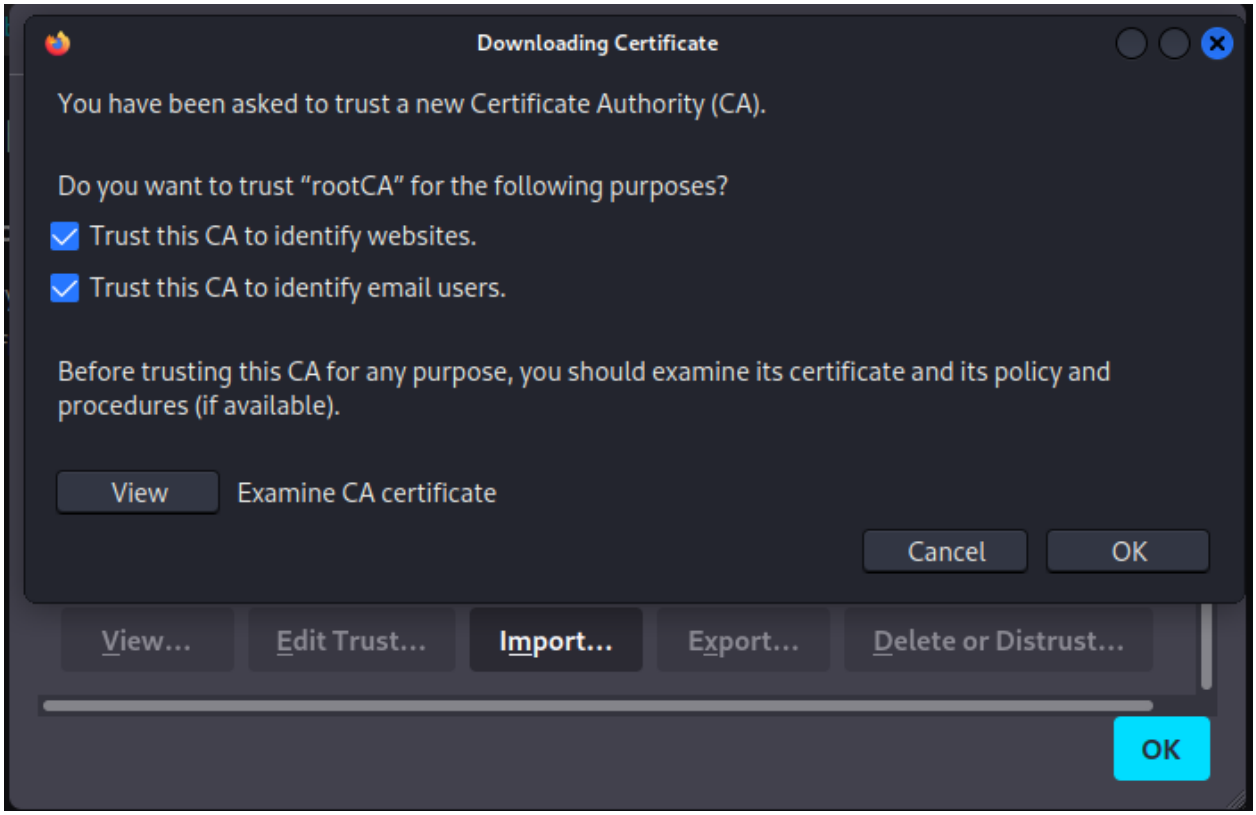
In the certificate manager we can upload our ca.crt and sub-ca.crt.

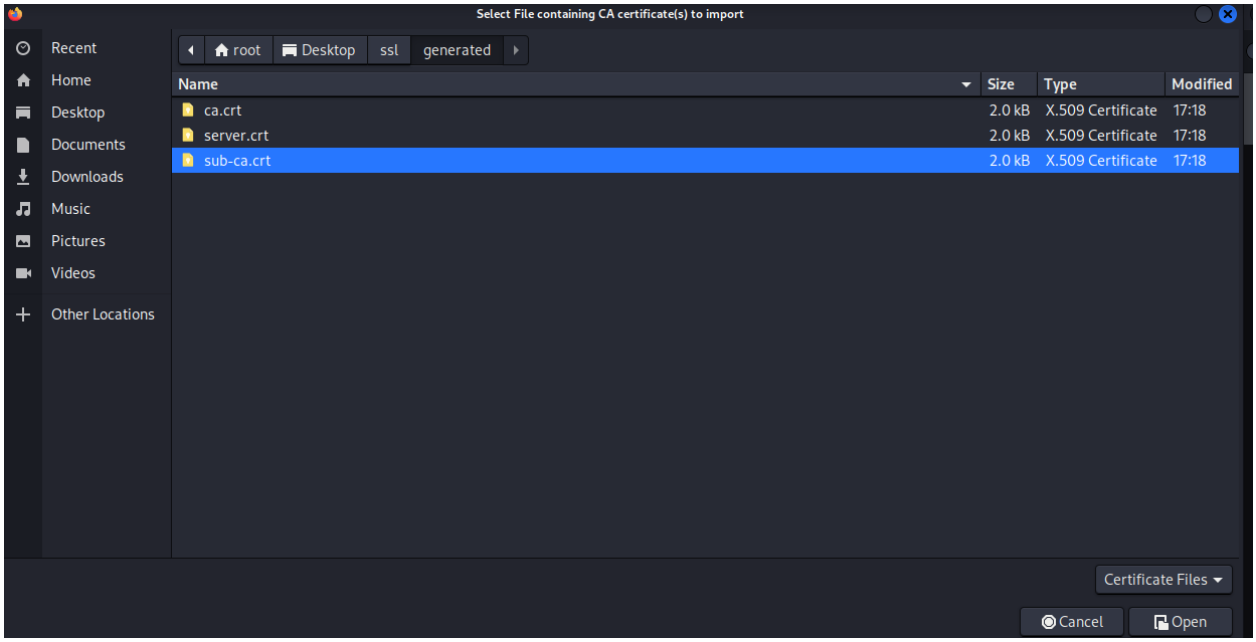


Each time we need to select our .crt file and hit the “open” button.

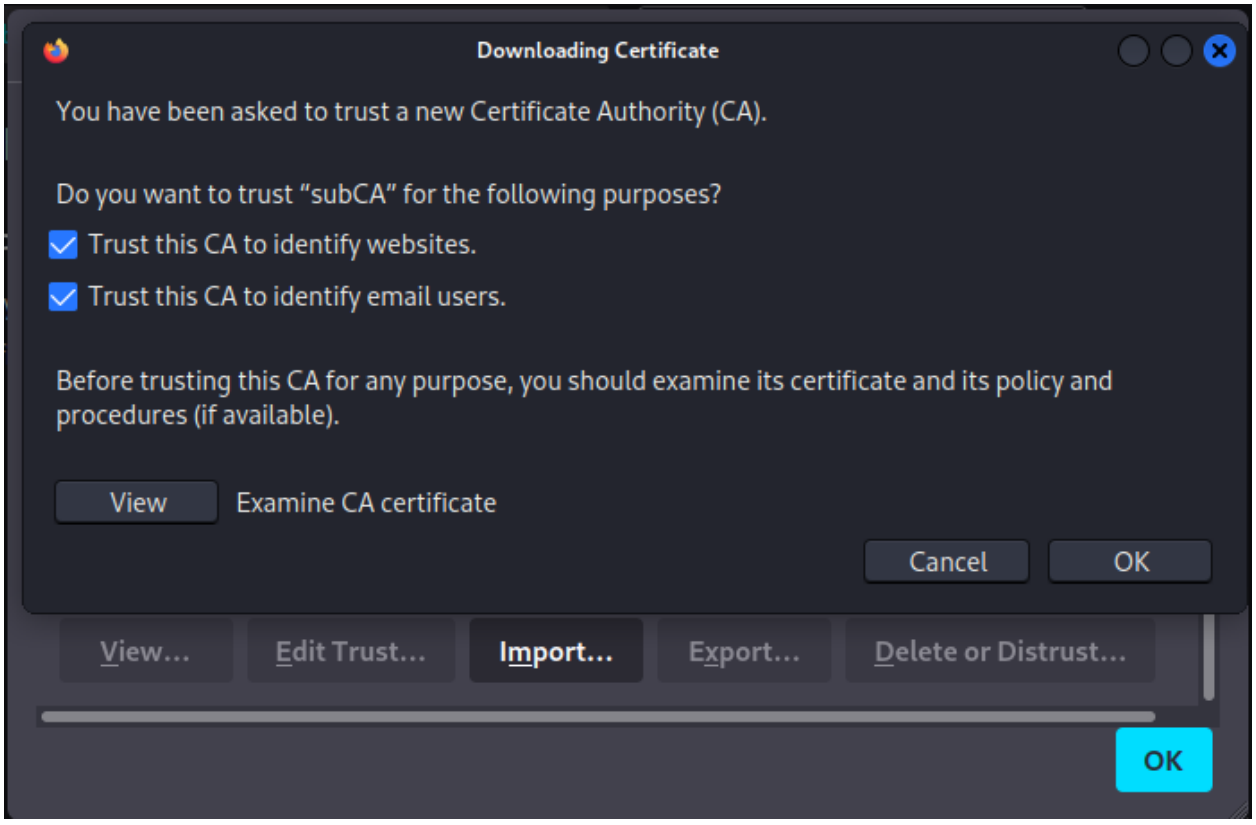


We need to make sure that we check all the necessary boxes to trust this ca.

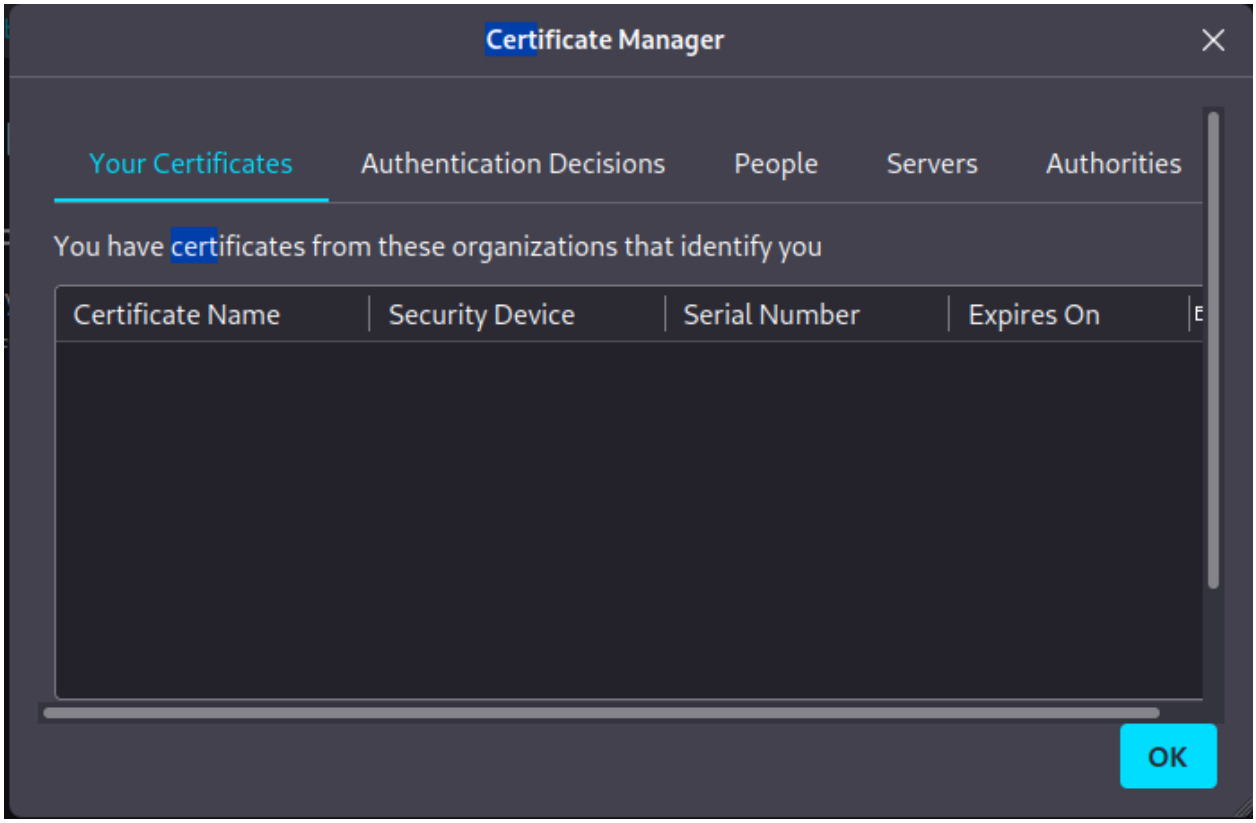


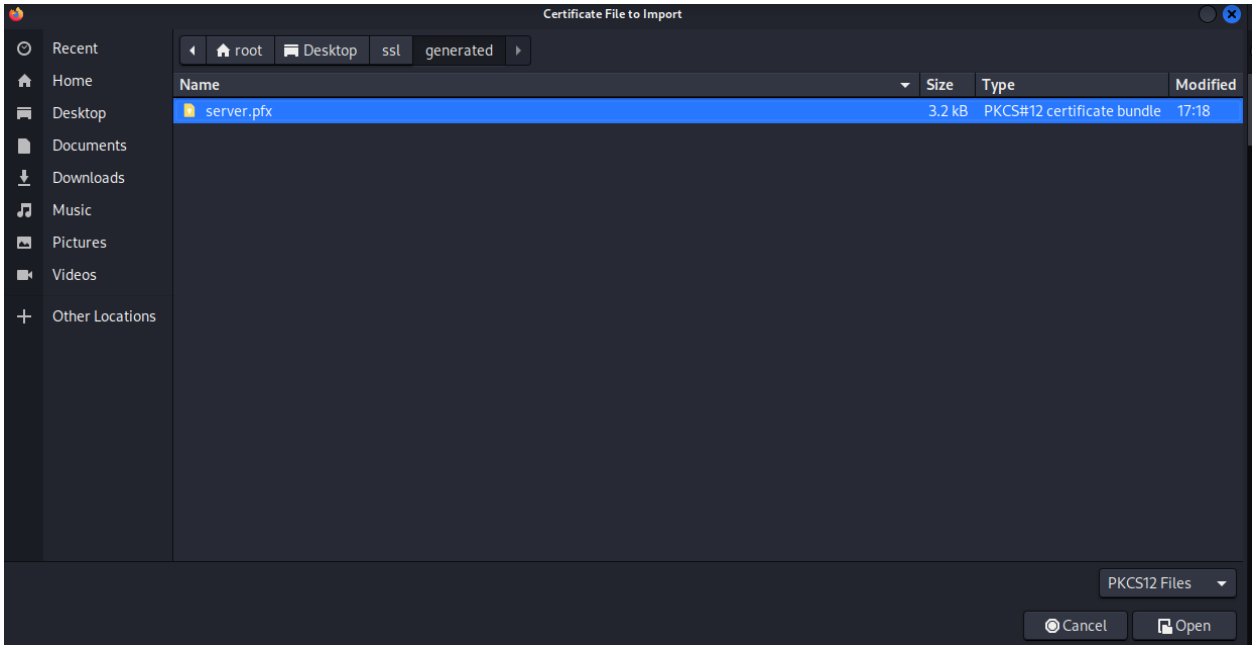


The process is same as previous.

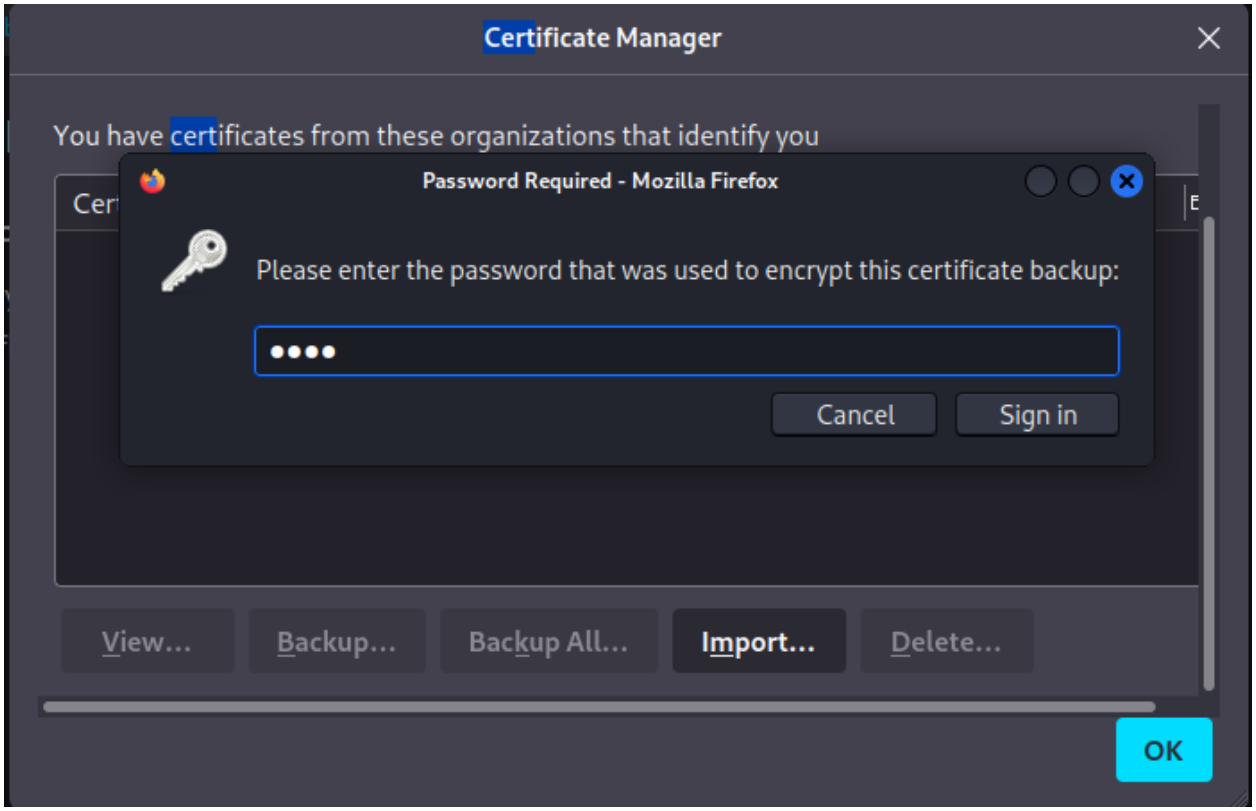


Step 14: After uploading the .crt files into the “Authorities” section, in the certificate manager, there is a section called “Your Certificates." Here we need to upload the generated server.pfx file to ensure our self-signed certificate.

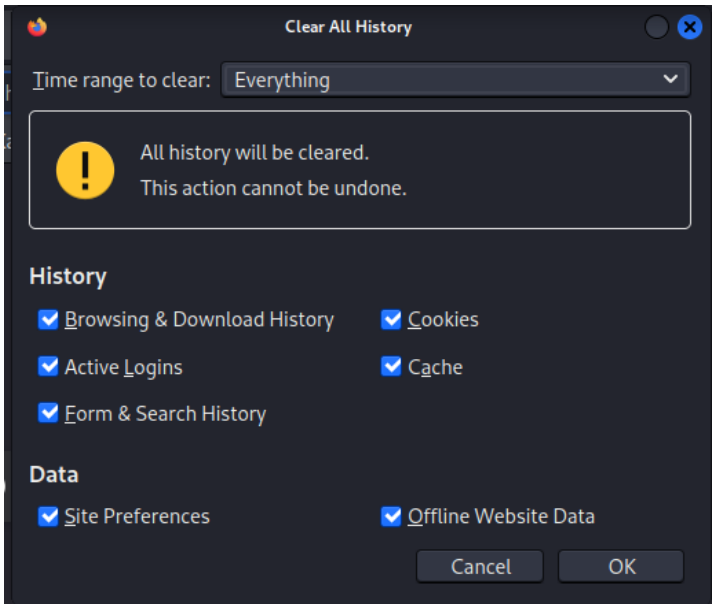




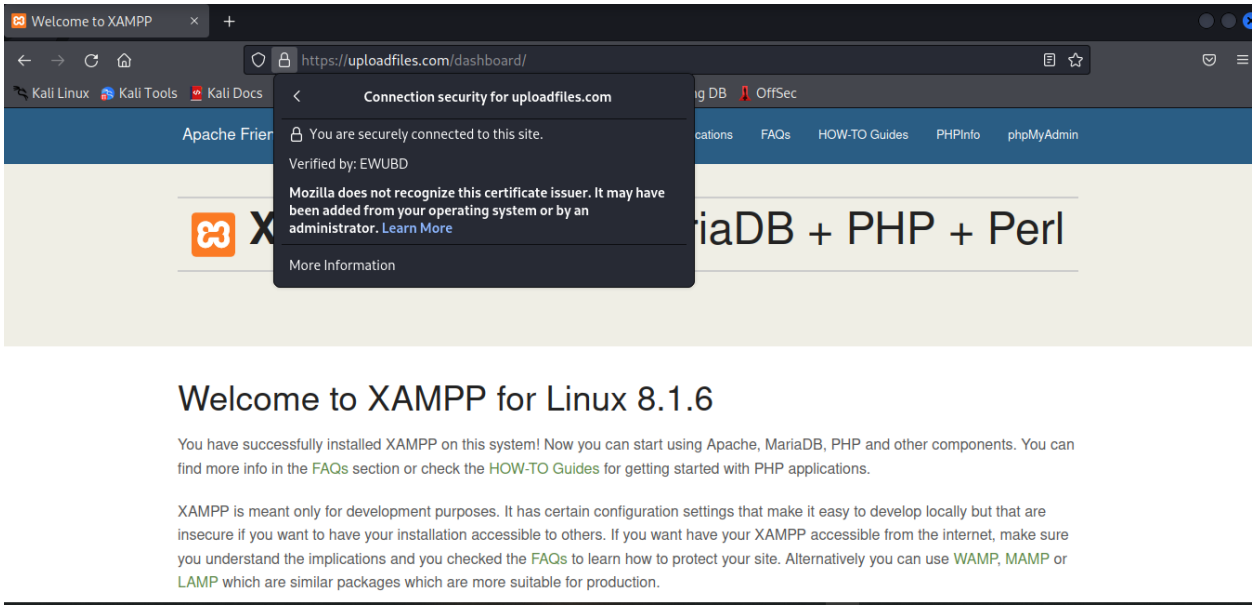
After uploading the file, we need to use the password that was used to encrypt the certificate, which was “root” (that same password when we gave it in the pass phrase checking while generating the certificates using OpenSSL).



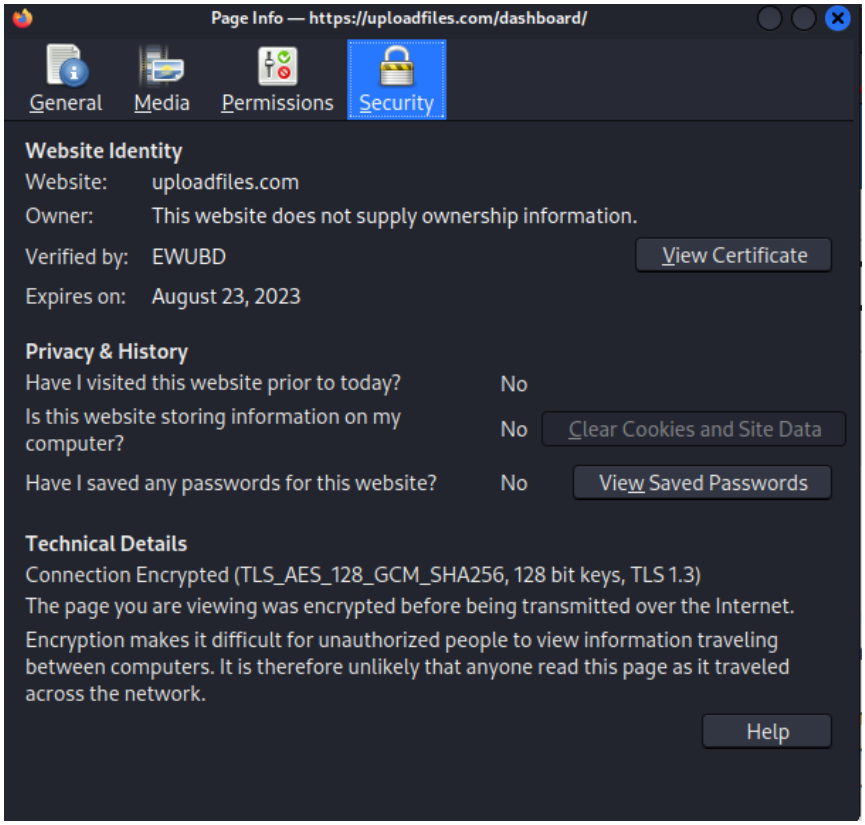
Step 14: How we need to delete all the cache files or cookies from the browser, then restart the browser.

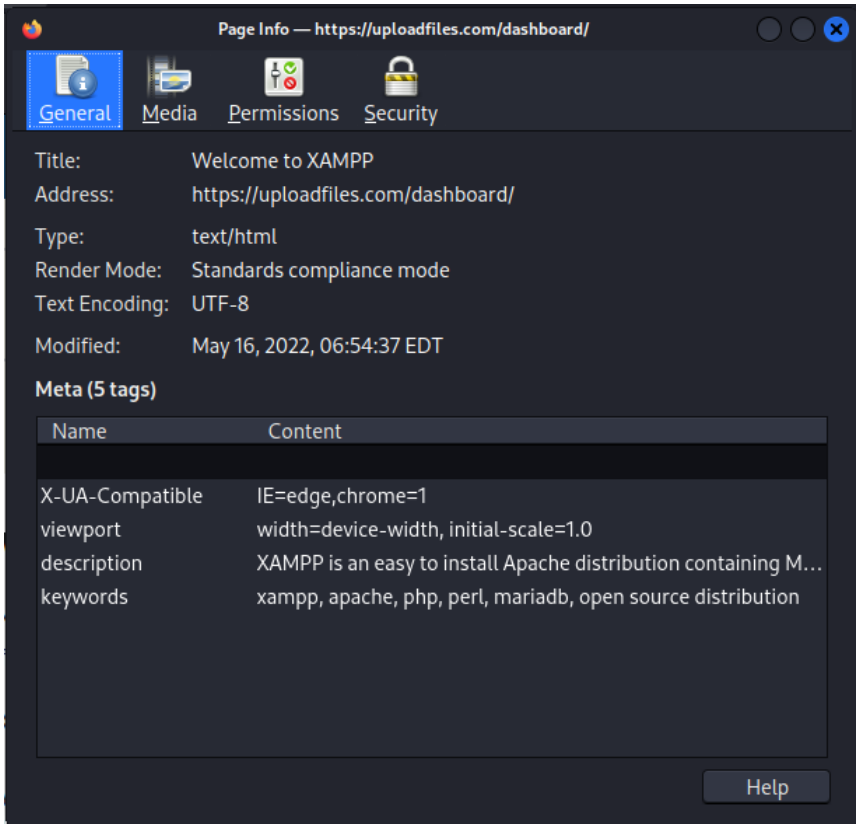


Step 15: Now we can get our padlock sign. Which means our certificates while generating with OpenSSL were successful.



Here are a few more details about the certificates.



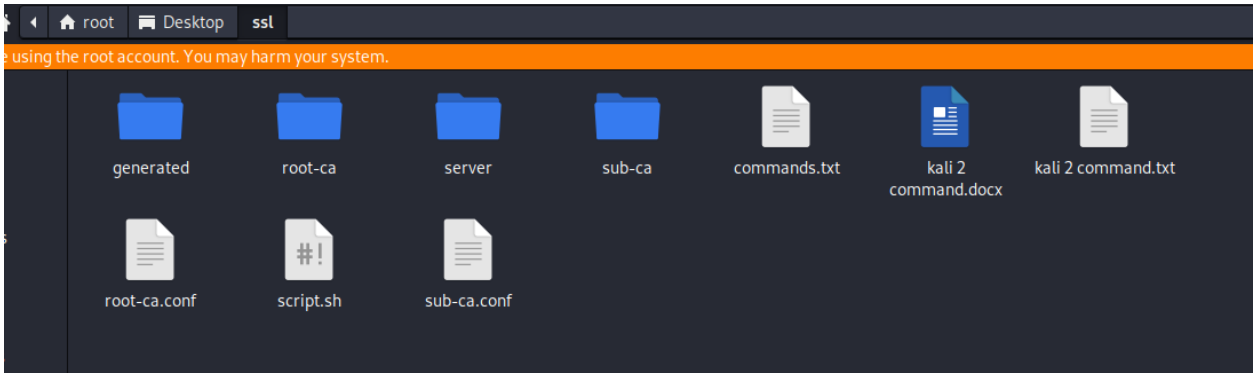

Page Info — https://uploadfiles.com/dashboard/
General
Media
Permissions
Security
Title: Welcome to XAMPP
Address: https://uploadfiles.com/dashboard/
Type: text/html
Render Mode: Standards compliance mode
Text Encoding: UTF-8
Modified: May 16, 2022, 06:54:37 EDT
Meta (5 tags)
Name
Content
X-UA-Compatible
IE=edge,chrome=1
viewport
width=device-width, initial-scale=1.0
description
XAMPP is an easy to install Apache distribution containing M...
keywords
xampp, apache, php, perl, mariadb, open source distribution
Help

Configuring The Client Machine To Get Access To The Secure Domain Name

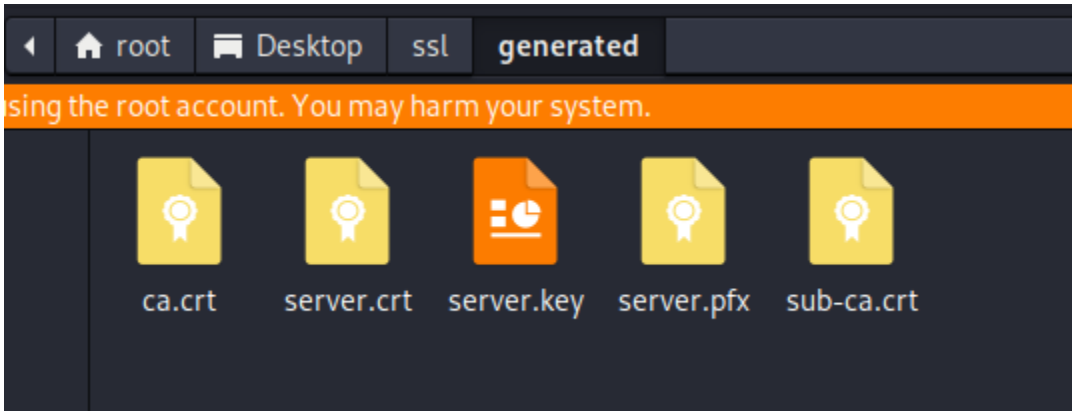
After the successful generation of certificates and assuring trust in browsers, now we need to get access to that configured domain name with a secured connection.
For this work we are creating another virtual OS in VMware (the process is the same).

From the server, we will share the generated certificates and other necessary files with the client machine so that we can do some little configuration.

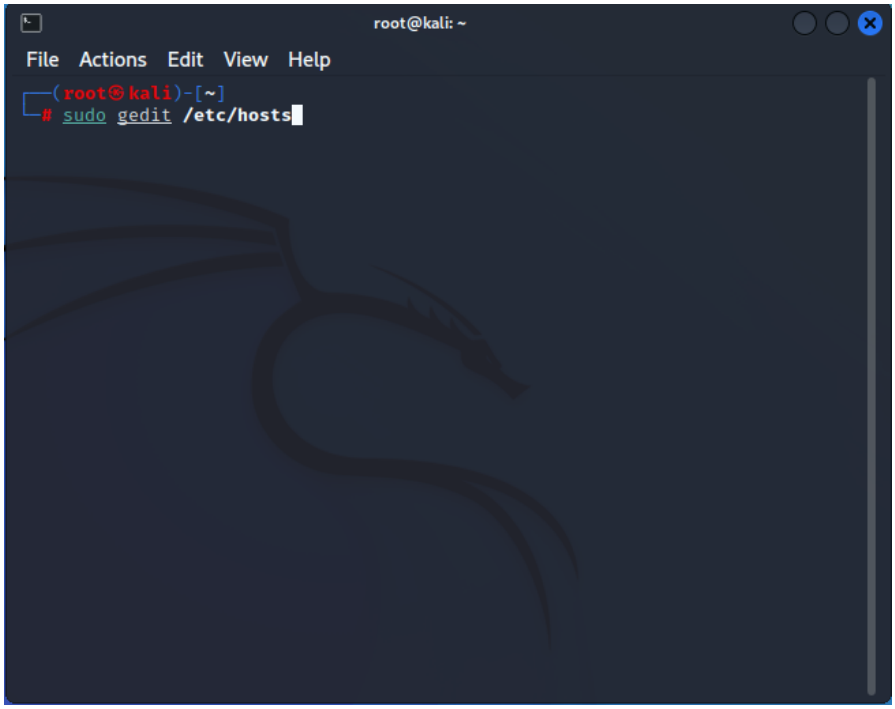
Step 1: Copy all the necessary SSL files to the client machine.



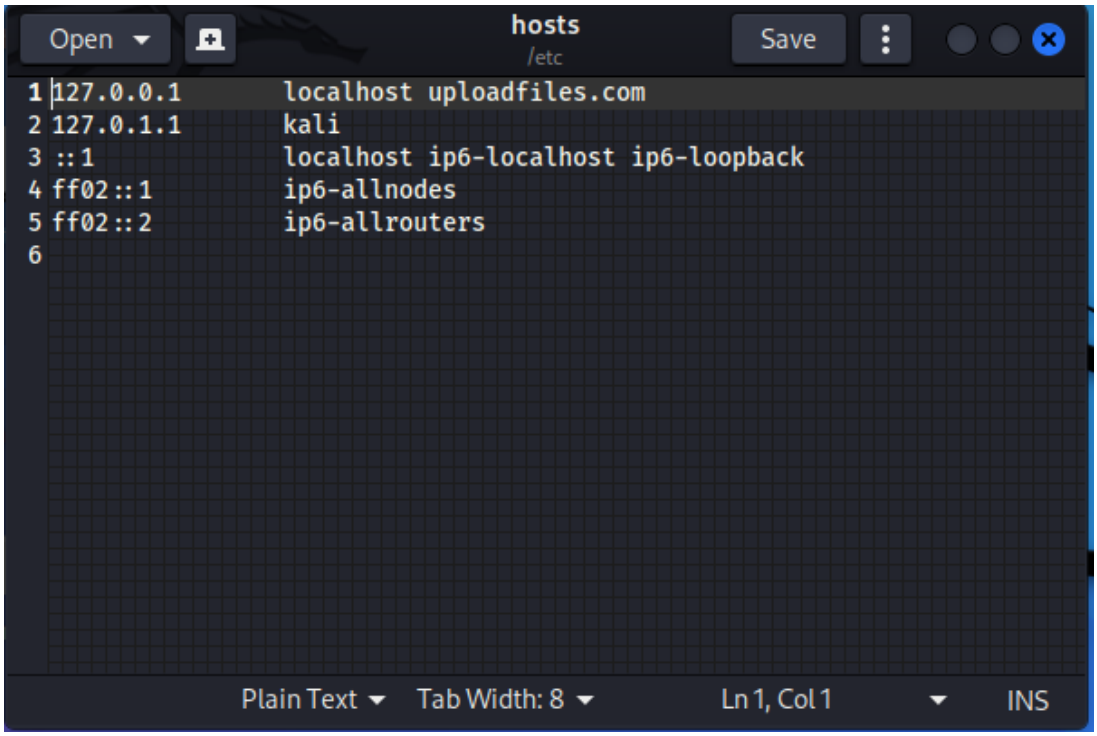
Step 2: From the "generated" folder we can find ca.crt, server.crt, server.key, server.pfx, and sub-ca.crt.



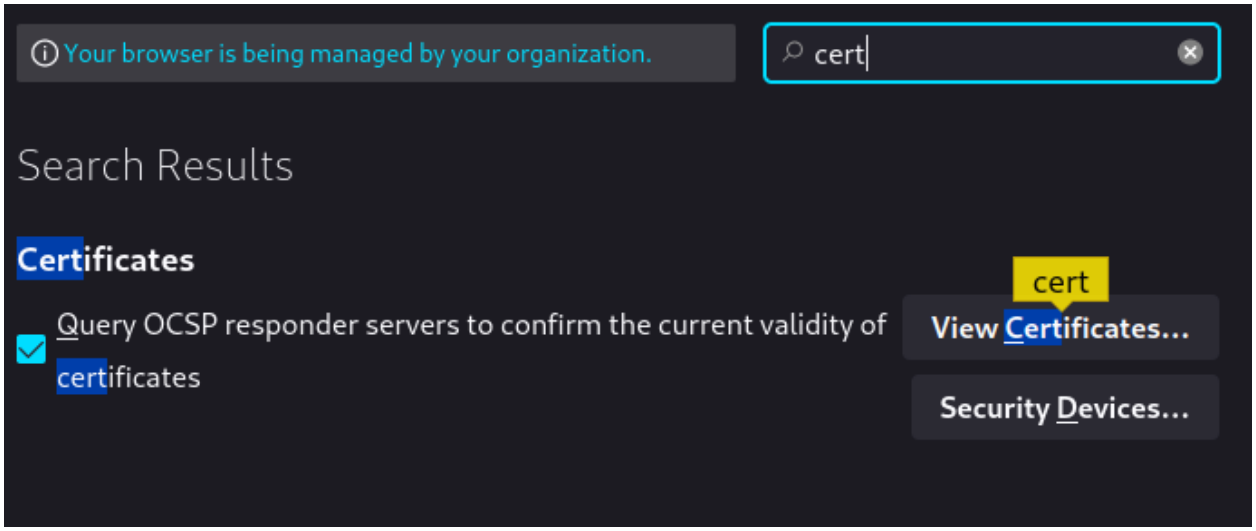
Step 3: Open terminal with the shortcut of ctrl+shift+t then type `sudo gedit /etc/hosts` (If gedit is not installed on the client machine, then simply install gedit, which can be used as a text editor. Use `sudo apt install gedit`).



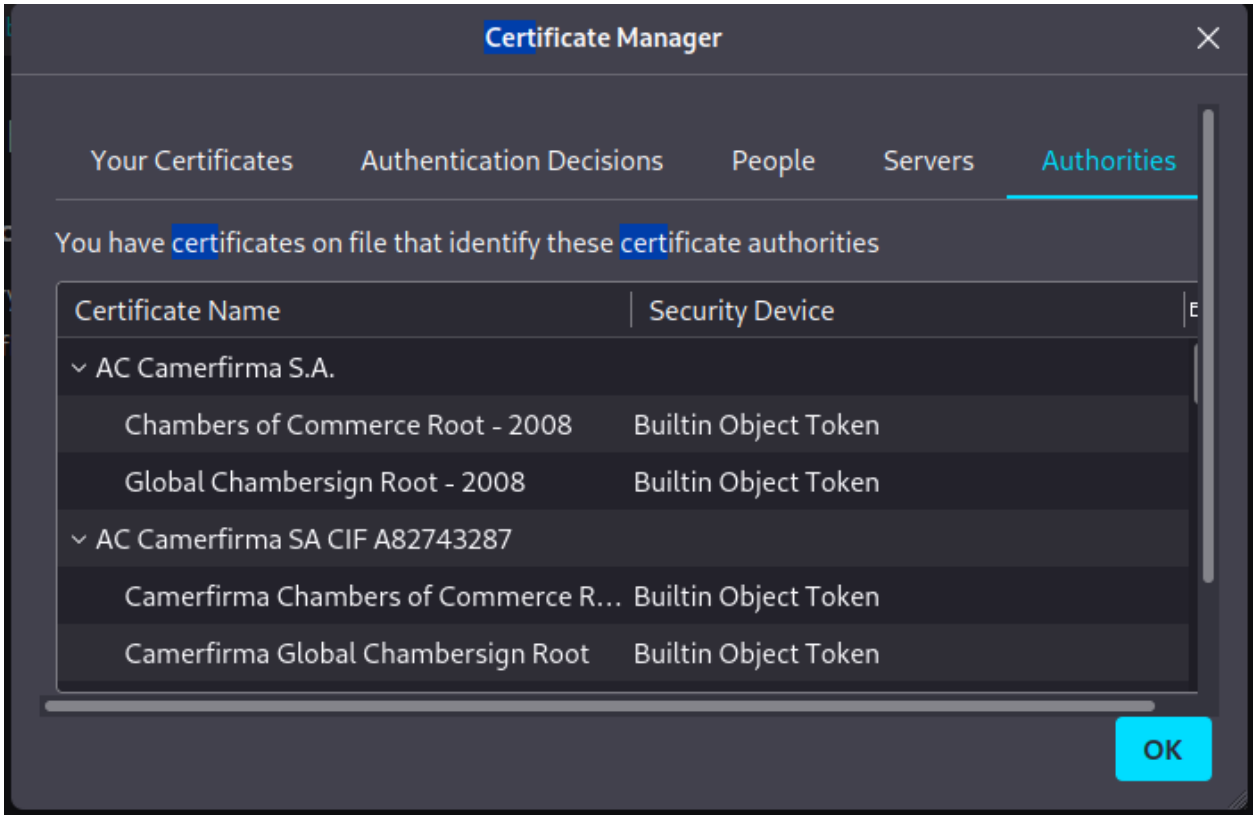
Then the hosts file will open.
We will add our domain name beside 127.0.0.1 localhost uploadfiles.com (like that)



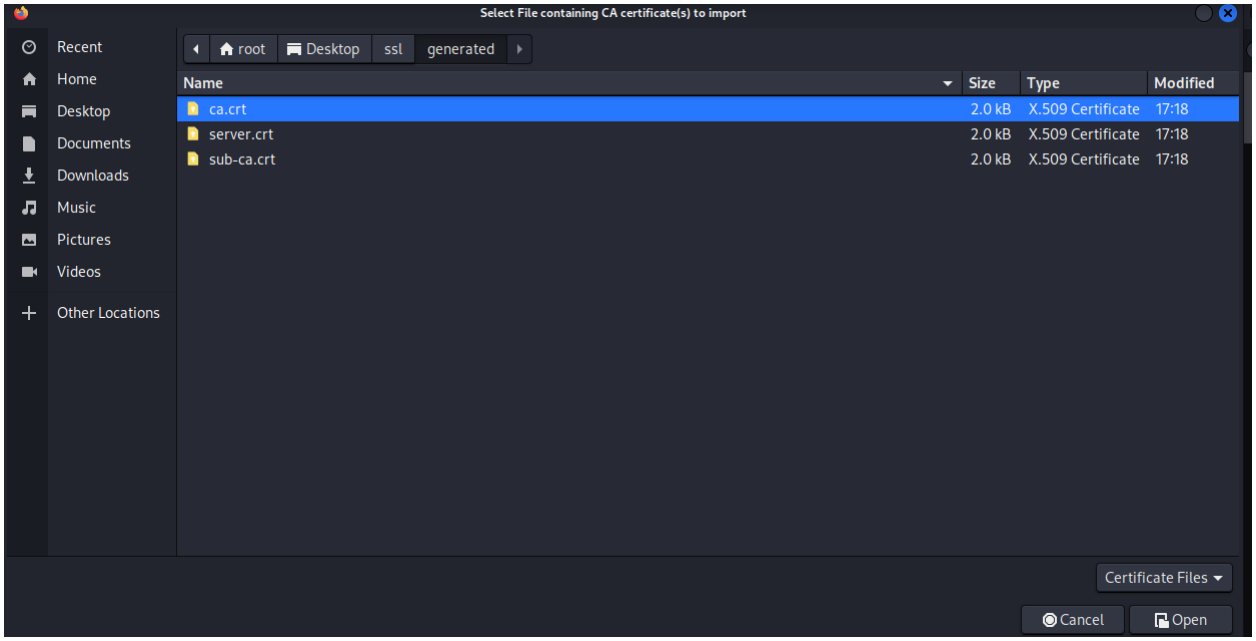
Step 4: Now we need to upload those .crt files into the browser (the steps alike previous)



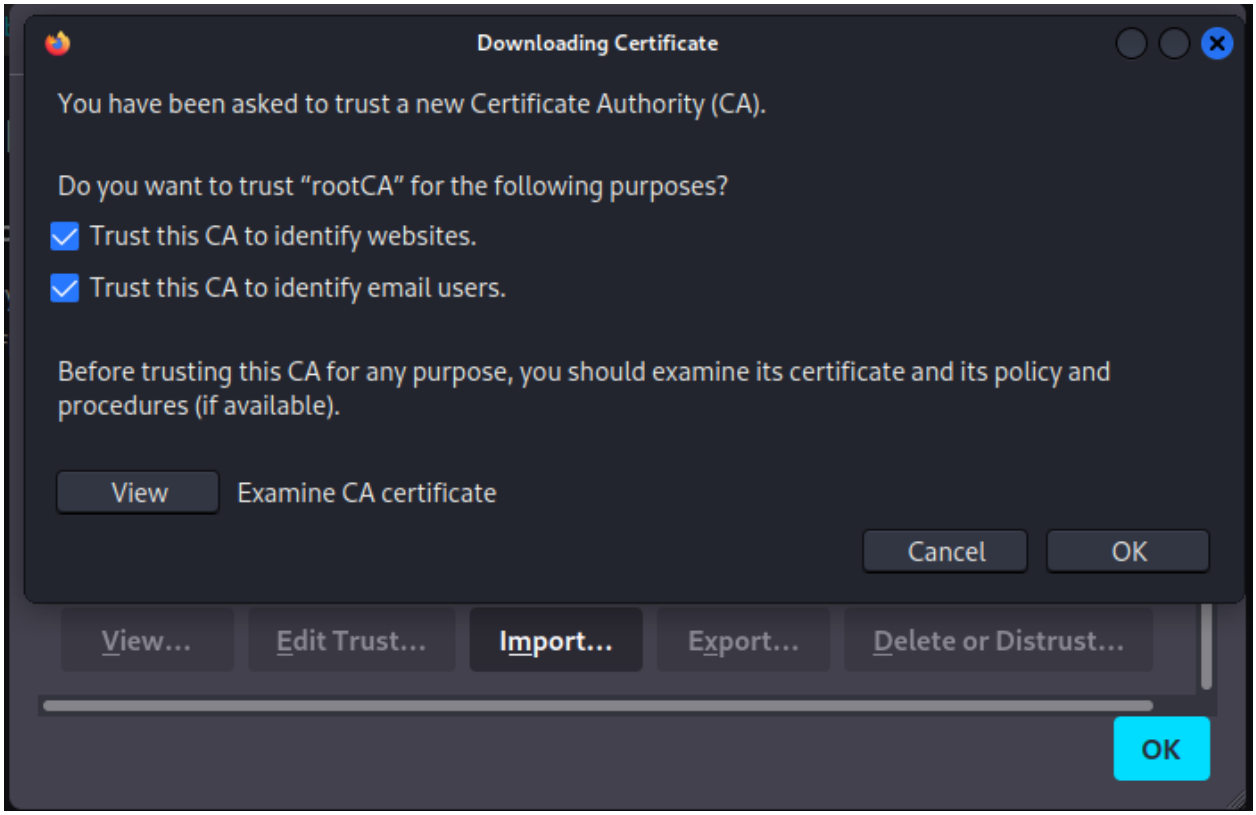
Scroll down the bar then there will be an "import button" to import the files.



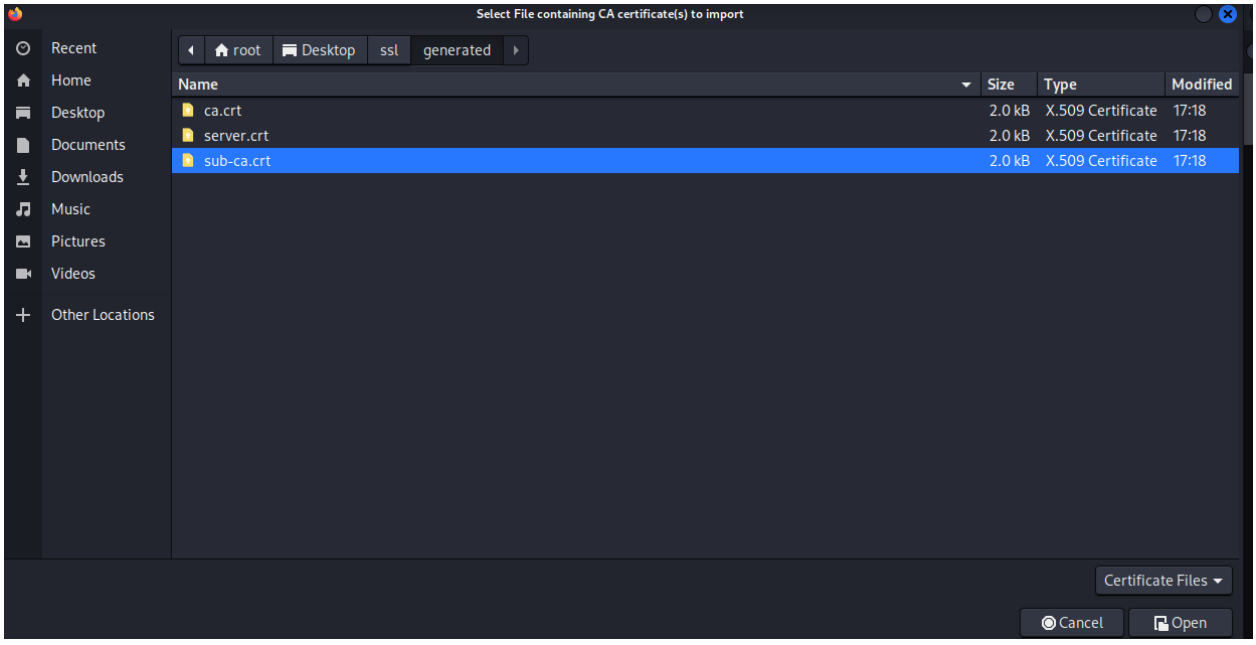
Choose the ca.crt file and click open.



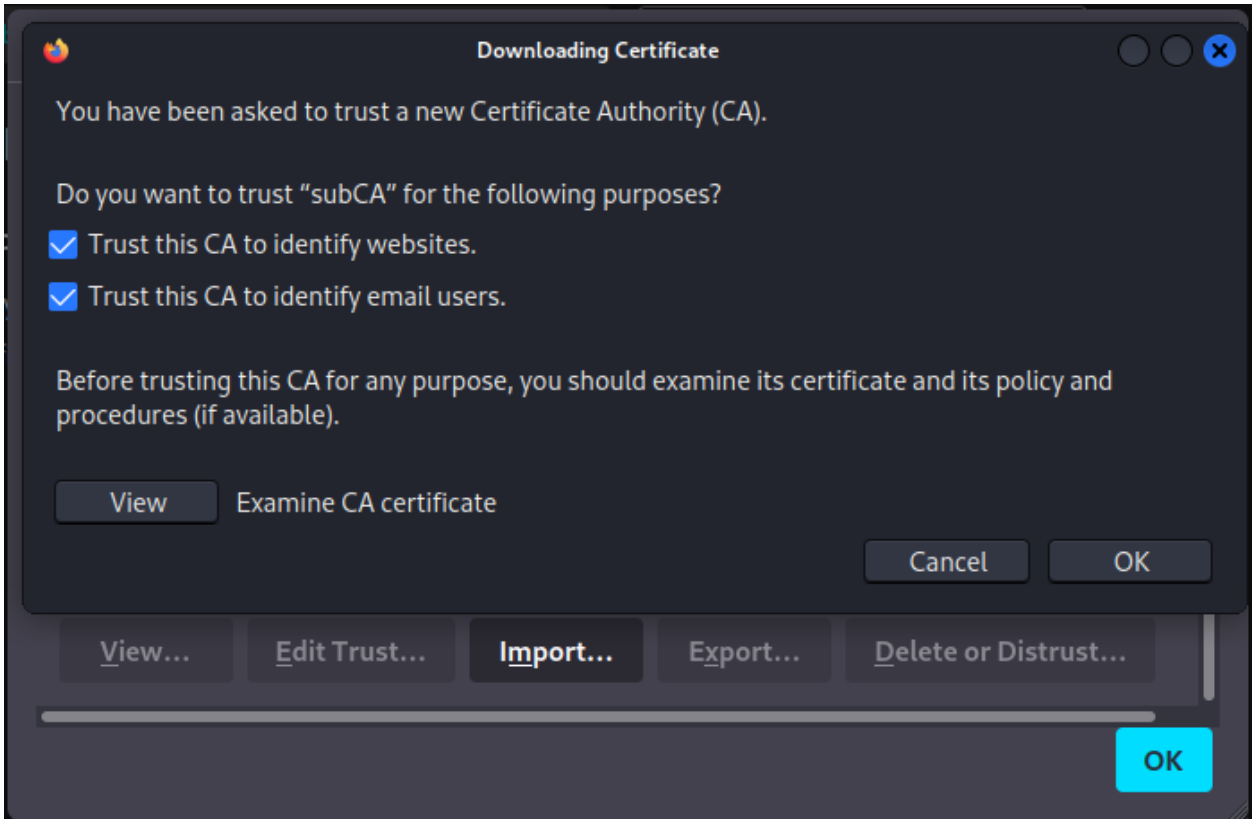
Check the boxes, then click OK.



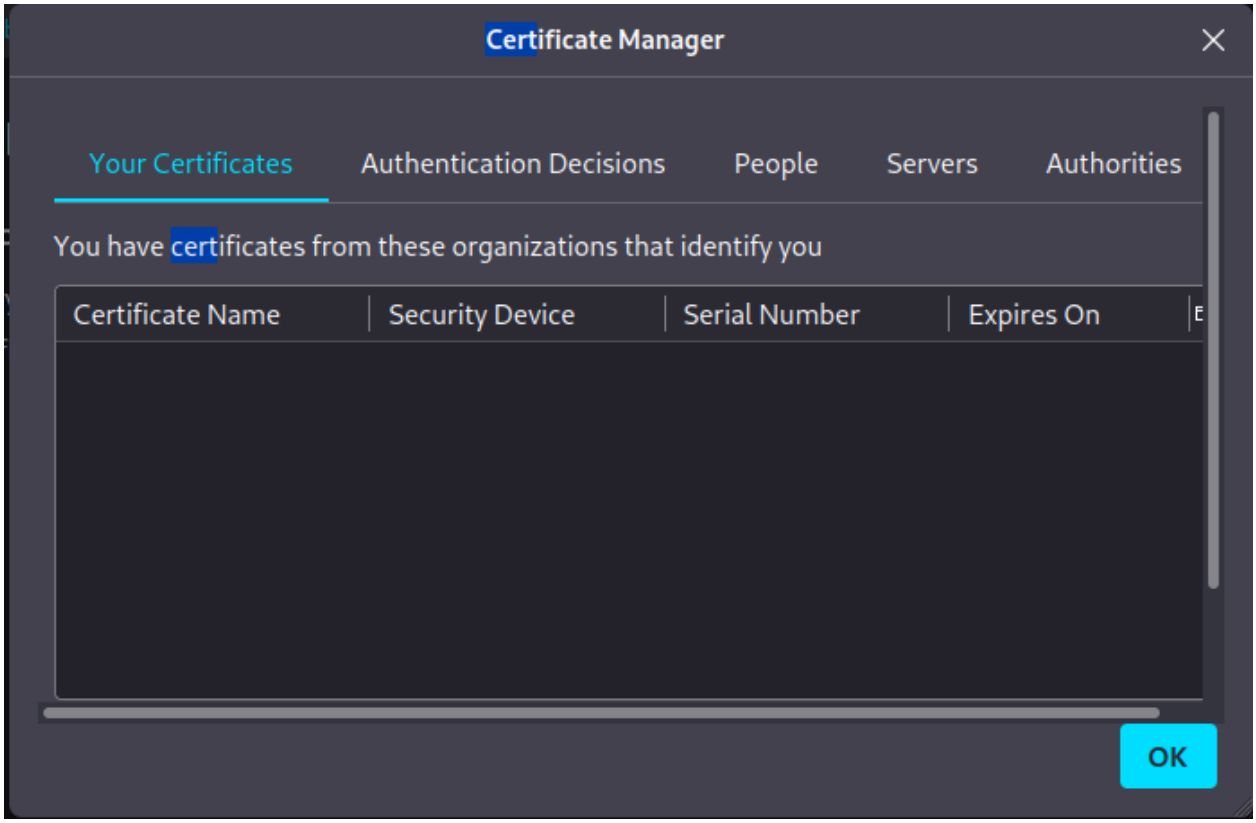
Choose the sub-ca.crt file and click open.



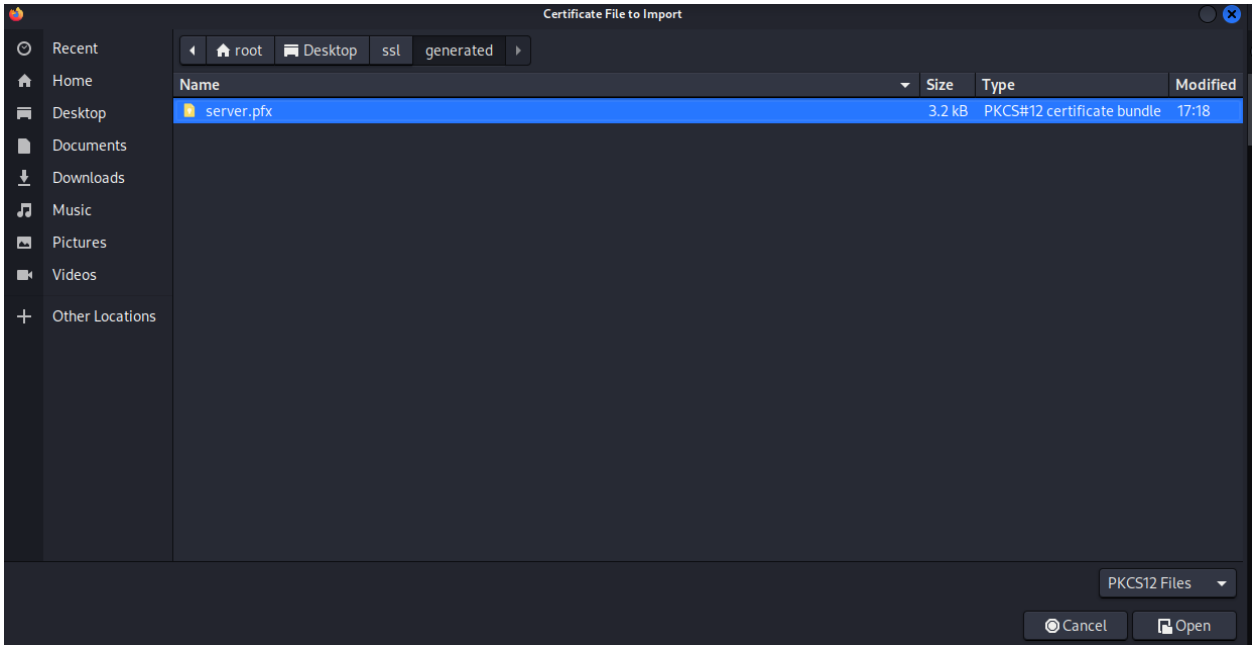
Check the boxes, then click OK.



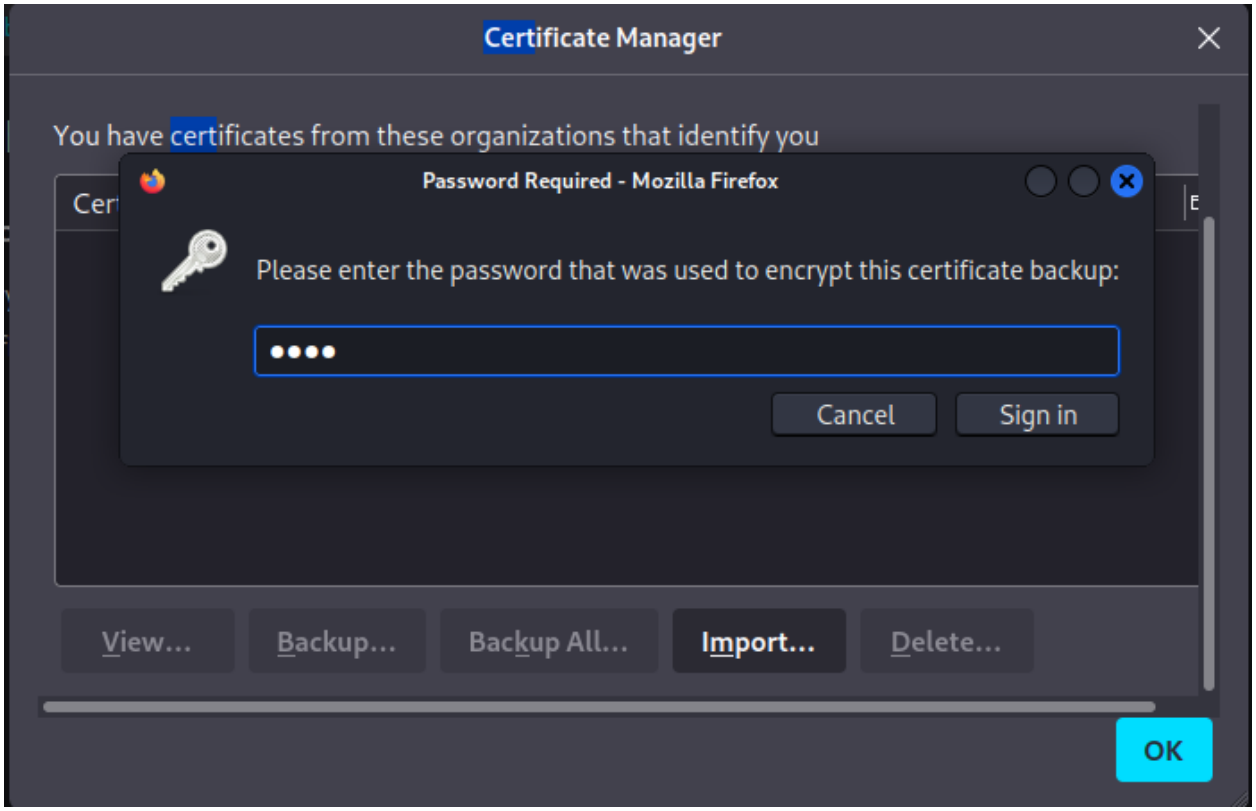
Scroll down the bar then there will be an “import button” to import the files.



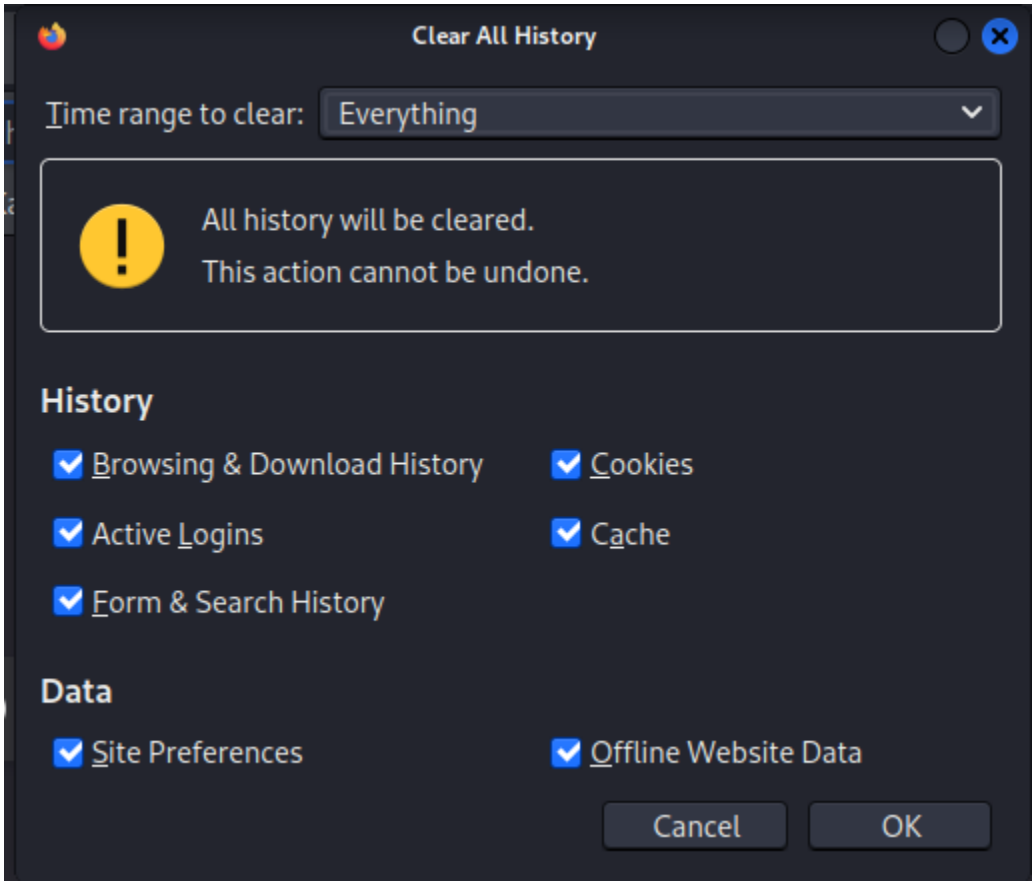
Selecting server.pfx, click open for further authentication.



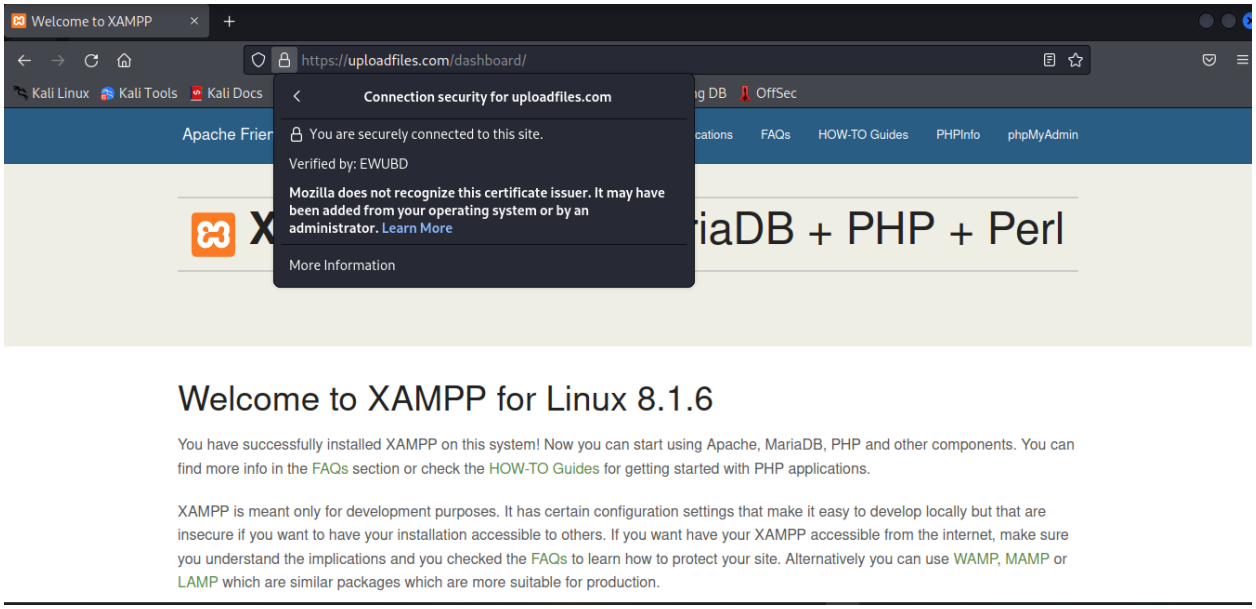
Now click "sign in."



Step 5: Now we need to clear the cache from the client machine browser. Then restart the browser.

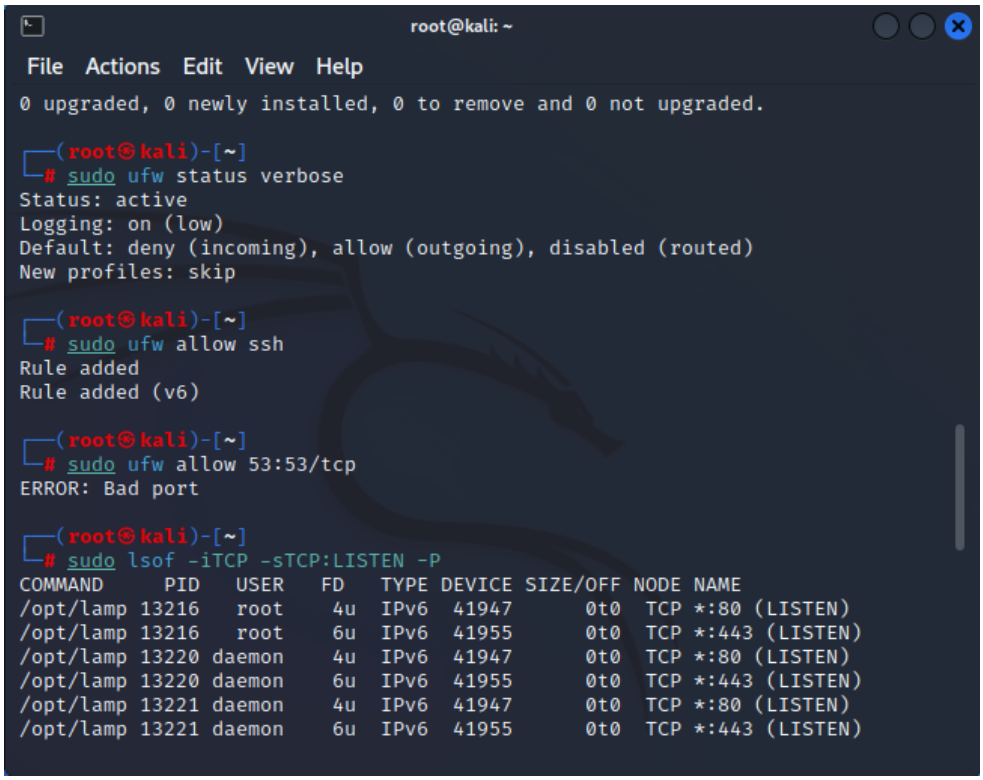


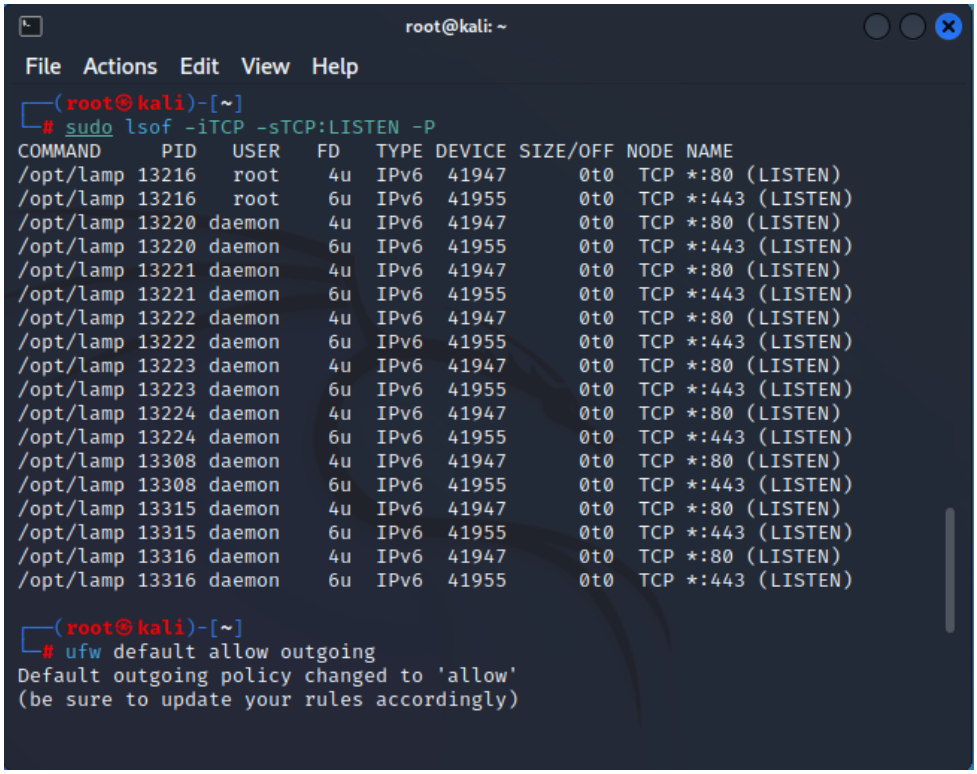
Step 6: We also brought the padlock sign into client machine browser.

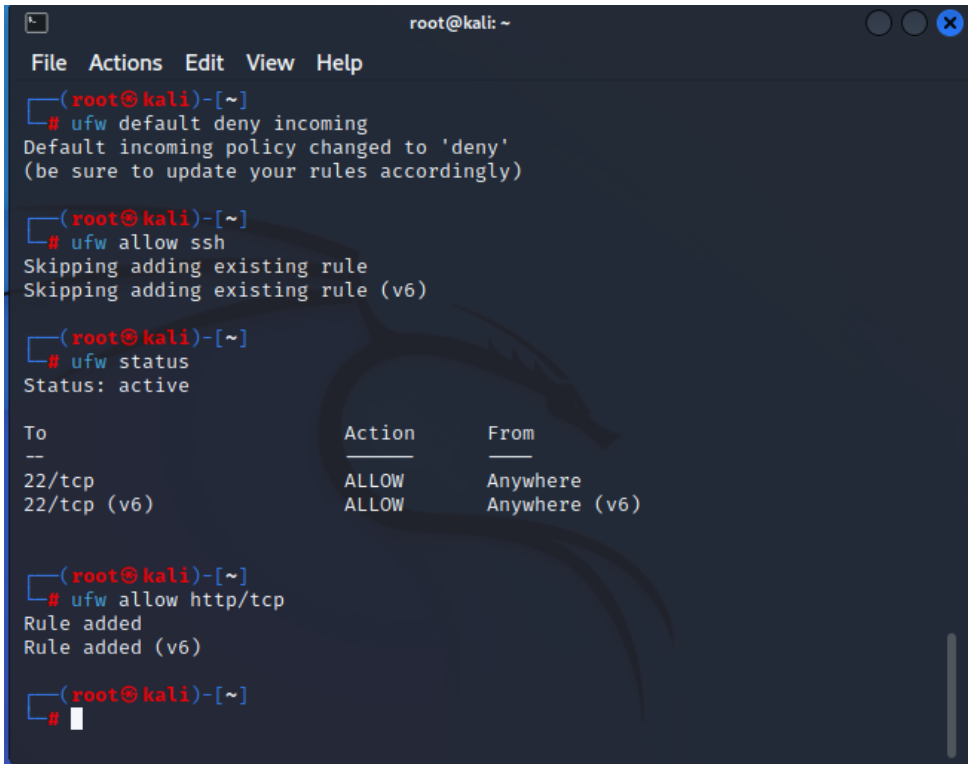


Firewall configuration to allow necessary ports (53, 80, 443) only

A firewall means protecting a network or system from unauthorized access with a firewall. The steps are shown below with appropriate commands.




root@kali: ~
File Actions Edit View Help
┌──(root㉿kali)-[~]
└─# sudo lsof -iTCP -sTCP:LISTEN -P
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
/opt/lamp 13216 root 4u IPv6 41947 0t0 TCP *:80 (LISTEN)
/opt/lamp 13216 root 6u IPv6 41955 0t0 TCP *:443 (LISTEN)
/opt/lamp 13220 daemon 4u IPv6 41947 0t0 TCP *:80 (LISTEN)
/opt/lamp 13220 daemon 6u IPv6 41955 0t0 TCP *:443 (LISTEN)
/opt/lamp 13221 daemon 4u IPv6 41947 0t0 TCP *:80 (LISTEN)
/opt/lamp 13221 daemon 6u IPv6 41955 0t0 TCP *:443 (LISTEN)
/opt/lamp 13222 daemon 4u IPv6 41947 0t0 TCP *:80 (LISTEN)
/opt/lamp 13222 daemon 6u IPv6 41955 0t0 TCP *:443 (LISTEN)
/opt/lamp 13223 daemon 4u IPv6 41947 0t0 TCP *:80 (LISTEN)
/opt/lamp 13223 daemon 6u IPv6 41955 0t0 TCP *:443 (LISTEN)
/opt/lamp 13224 daemon 4u IPv6 41947 0t0 TCP *:80 (LISTEN)
/opt/lamp 13224 daemon 6u IPv6 41955 0t0 TCP *:443 (LISTEN)
/opt/lamp 13308 daemon 4u IPv6 41947 0t0 TCP *:80 (LISTEN)
/opt/lamp 13308 daemon 6u IPv6 41955 0t0 TCP *:443 (LISTEN)
/opt/lamp 13315 daemon 4u IPv6 41947 0t0 TCP *:80 (LISTEN)
/opt/lamp 13315 daemon 6u IPv6 41955 0t0 TCP *:443 (LISTEN)
/opt/lamp 13316 daemon 4u IPv6 41947 0t0 TCP *:80 (LISTEN)
/opt/lamp 13316 daemon 6u IPv6 41955 0t0 TCP *:443 (LISTEN)
┌──(root㉿kali)-[~]
└─# ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)


root@kali: ~
File Actions Edit View Help
┌──(root㉿kali)-[~]
└─# ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
┌──(root㉿kali)-[~]
└─# ufw allow ssh
Skipping adding existing rule
Skipping adding existing rule (v6)
┌──(root㉿kali)-[~]
└─# ufw status
Status: active
To Action From
-- ------ ----
22/tcp ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)
┌──(root㉿kali)-[~]
└─# ufw allow http/tcp
Rule added
Rule added (v6)
┌──(root㉿kali)-[~]
└─#

```
root@ufw-tutorial:~# ufw allow from [redacted] to any port 22 proto tcp
Rule added
root@ufw-tutorial:~# ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
22/tcp                     ALLOW       [redacted]
22/tcp (v6)                ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
```

```
root@ufw-tutorial:~# ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     ALLOW IN    Anywhere
[ 2] 80/tcp                     ALLOW IN    Anywhere
[ 3] 22/tcp                     ALLOW IN    [redacted]
[ 4] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 5] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
```

```
root@ufw-tutorial:~# ufw delete 1
Deleting:
 allow 22/tcp
Proceed with operation (y|n)? y
Rule deleted
```

```
root@ufw-tutorial:~# ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 80/tcp                     ALLOW IN    Anywhere
[ 2] 22/tcp                     ALLOW IN    [illegible]
[ 3] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 4] 80/tcp (v6)                ALLOW IN    Anywhere (v6)

root@ufw-tutorial:~# ufw delete 3
Deleting:
 allow 22/tcp
Proceed with operation (y|n)? y
Rule deleted (v6)
```

```
root@ufw-tutorial:~# ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 80/tcp                     ALLOW IN    Anywhere
[ 2] 22/tcp                     ALLOW IN    [illegible]
[ 3] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
```

So that's how we can configure the firewall according to our needs.

Conclusion

This report will provide anyone with a proper guideline to do the task of Securing a Networked System with Public Key Infrastructure (Implementing Transport Layer Security on HTTP for https:// connection). This report contains very derived steps (solutions) to solve the problem with appropriate commands.