

Report S103

INSTALLATION OF A STATION FOR THE DEVELOPMENT

TP1I

DENOUAL GLEN – VENGEON NOE

Table of contents

Introduction	2
1- Objectives	2
2 - Available material	2
3 - Assignment of tasks	2
Install an Operating System (OS) on the Raspberry	3
1 - Download the image file	3
2 - Flash an image	3
Raspberry configuration	4
What is an Apache Server?	4
Installation process	4
Edit the HTML	5
Virtual Host usage	5
What is Virtual Host	5
Configuration process	5
Conclusion	6

Introduction

1- Objectives

The project S103 consists in a first step to:

1. install an Operating System on the Raspberry.
2. Then we need to install an Apache Server on the Raspberry, this step we allow us to configure a website on the Raspberry.
3. The next step and set up multiple websites on the Raspberry. More specifically it will take one site per student in the group and each website should be accessible on the same server through virtual hosting.
4. And we were also free to configure a password and a hostname to access the site.
5. And if we wanted to get a very high grade, we should put TLS on the website.

2 - Available material

For this work, the available material is:

- 1 Raspberry Pi 400 (A kit: keyboard+mouse+Raspberry card into the keyboard),
- 1 SD card (16Go),
- 1 SD card reader/adaptor (only to connect the SD card to the pc),
- 1 Ethernet-to-USB adapter (to connect the raspberry to the pc),
- 1 UTP Patch (short Ethernet cable).

3 - Assignment of tasks

Important: For the division of tasks, it is only the work of researching guides on the internet (the major part of the work). Otherwise during the manipulation, we were all behind the other one to learn, to help if it was necessary but specially to understand everything that was done.

- Noé: installation of the OS
- Glen: install Apache Server and set up a website
- Florian: configure Virtual Host and set up multiple websites

Install an Operating System (OS) on the Raspberry

1 - Download the image file

In this part we will install an Operating System (shorted in « OS ») on the Raspberry. Indeed, every Raspberry need an OS to work on it. In fact, a raspberry is a compact computer and the power varies according to its price. And like every computer, it is necessary to have an OS to interact with the computer. Thanks to this we will be able to host our website and even more.

First, we decide to install the Raspberry Pi (formerly Raspbian) operating system as recommended, since no one in the group has experience with the Arch Linux ARM operating system. To do this, simply search the Internet for "how to install the Raspberry Pi operating system" and follow the steps.

Step 1: we went to the official Raspberry website : <https://www.raspberrypi.com/software/> for download the file of the OS (extension : img.xz)

Step 2: We search on the official raspberry website to manually install an operating system image so we clicked « See all download options »

Manually install an operating system image

Browse a range of operating systems provided by Raspberry Pi, and download them to install manually.

[See all download options](#)



Step 3: We download the « Raspberry Pi OS with desktop for system : 64-bit » on the IUT's computer. We choose the 64-bit version because the Raspberry pi 400 has a 64-bit system and so it's allows to the Raspberry more performance : https://www.frandroid.com/produits-android/ordinateurs/1229779_raspberry-pi-un-gain-de-performance-gratuit-de-48-grace-au-64-bits (*french article*)

2 - Flash an image

Step 4 : Since now after the download of the OS img.xz was completed we follow these steps <https://www.youtube.com/watch?v=ntaXWS8Lk34> this is an official tutorial video showing how to use the Raspberry Pi Imager.

And so to detail, we connect the SD card to the computer through the adapter provided.

Then, we launched the Raspberry imager application (already present on the IUT PCs),

We select the correct Operating System (OS) file in our download directory

Moreover, select our 16BG SD card, for the « STORAGE ».

At last we click on « WRITE », wait the loading and click « CONTINUE »

That's all! Raspberry Pi Imager is a good way to install Raspberry Pi OS without going through the Windows terminal and it's very easy to use and it's quick.

Raspberry configuration

Although in the previous step, we could have pre-configured the Raspberry as indicated in the tutorial, we decided to configure it only now. Always keeping in mind that the goal is to set up a website. And so, for this, we need one essential thing: an Internet access for our Raspberry to download an Apache Server.

We remove SD card from the adapter to put into the Raspberry directly, start the computer and change the display source of the screen. We were able to set a password to our user on the Raspberry, we save it in a shared document.

Indeed, we were surprised to see that we had no Internet connection when we tried to search for an Apache server download on Chromium (*default search engine for Raspberry Pi OS*). The error message was « *Your clock is behind* » that means our Raspberry was not set to the right time. We could observe thanks to the graphic user interface provided by Raspberry Pi OS in the upper right corner that effectively the date and time were not correct.

So, we search how to set it on time. And so we find a simple command that is `sudo date -s 'YYYY-MM-DD HH:MM:SS'` we just put the right date like : `sudo date -s '2023-01-10 09:55:32'` and it works. Now we got an Internet access on our Raspberry

Install Apache Server and set up a website

1- What is an Apache Server?

Apache is a free and open-source web server software that powers about 46% of the world's websites. One of its main advantages is its ability to be customized, which makes it particularly upgradeable. Its role is to listen to requests made by browsers (which ask for web pages), to look for the requested page and to send it back.

2 - Installation process

As usual, we have researched how to install an Apache server on a raspberry. For this part we did not follow only one tutorial but we followed several like:

<https://bytexd.com/how-to-install-apache-web-server-on-linux/>

<https://www.ubuntupit.com/how-to-host-a-complete-website-on-a-raspberry-pi-guide/>

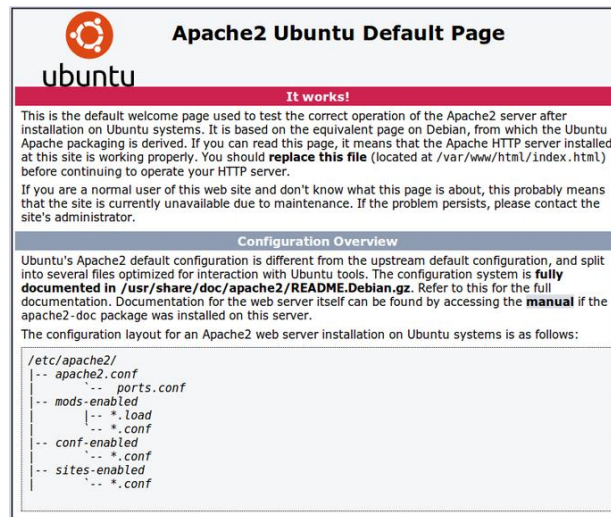
Concretely we have made these commands:

- 1- `sudo apt update`: Allows you to install the latest version of a package
- 2- `sudo apt install apache2`: Here, the package is apache2
- 3- `apache2 -v`: displays the Apache server version and its creation date
This is to verify that the download was successful and that the correct version was correctly implemented
- 4- `sudo systemctl status apache2`: Set the activity status of the Apache server to: active.

5- **hostname -I**: Allows to get the ip of the Raspberry
we will need it later so we just copy-paste it in a notepad
During the procedure our IP is the following: 10.42.0.4

6- **ifconfig -a**: Displays the information necessary for network configuration

And after that we got this text on our website at 10.42.0.4, as the text says, it means that the installation of apache2 was successful. We now have our website up and running!



3 - Edit the HTML

And we are even able to edit the HTML code. To do that, we just need to access to the .html file and use a text editor like nano for example.

By default, it is located in the following path: /var/www/html, we can do this by using the command « cd » (change directory): **cd /var/www/html**

Now if we list all the files of this directory with the ls command there is a index.html. This is the file we were looking for. We can eventually see its content to know if it is the right file by using the command: **cat index.html**.

For edit it we just need to use the nano command: **nano index.html**

Virtual Host usage

1 - What is Virtual Host

In computing, virtual hosting is a method that servers such as web servers use to host more than one domain name on the same computer, sometimes on the same IP address, while maintaining separate management of each of those names.

2 - Configuration process

Here are the commands for the virtual hosting:

1- `sudo etc/apache2/sites-available/siteun.conf` (`sudo etc/apache2/sites-available/site_name.conf`).
This is for create in the target directory, a .conf file named "siteun" (which corresponds to our site name) and allows you to configure a web site on an Apache server.

3- `nano /var/www/siteun` allows you to edit the text of the file requesting

2- In this file .conf we enter this code and modify it as required

Ex: we replace ServerName by "siteun" ...

```
<VirtualHost :80>
    ServerName site_name.com
    ServerAdmin
webmaster@site_name.com
    DocumentRoot
/var/www/site_name
    ErrorLog
${APACHE_LOG_DIR}/error.log
    CustomLog
${APACHE_LOG_DIR}/access.log
    combined
</VirtualHost>
```

Is to configure an Apache Virtual Host for a specific site.

3- `sudo a2ensite siteun.conf` allows you to activate a Virtual Host configuration file for Apache.

4- `sudo service apache2 restart` allows to restart Apache 2.

5- `sudo nano /etc/hosts` allows to open the host file with nano.

6- `10.42.0.0 siteun.com` allows you to configure the port of the web sites.
`10.42.0.1 sitedeux.com`
...

And we repeat the operation for each site.

Conclusion

To conclude this project, we will talk about what we have learned, the difficulties we have encountered and the choices we have made. In addition, we will talk about the optional part.

During this project we could learn during this project we were able to learn how to use a Raspberry board, to reinforce our use of Linux commands and we were introduced to Apache. Concerning the issues we encountered, we had only few minor issues. Finally, we only lost time due to our lack of experience and tutorials that were not perfectly adapted to our context.