Tek-it-Izy - User notice

Apache web server installation guide

SAE2.03 - D2
DEMOUGE Colin
LEFRANCOIS Thibaut

Tek-it-Izy

Summary

Apache web server installation guide

[0] Preamble p.2

[1] OS installation and Raspberry configuration p.3

[2] Internet connexion p.3

[3] Downloading softwares p.4
- -Apache
- -PHP

[4] SSH connexion p.5

[5] SFTP connexion p.6

[6] Default website p.9
- -Site configuration
- -Upload personal pages

[7] Public website p.14

[8] Intranet website p.18

[9] Appendices p.22

[0] Preamble

You will find along with the notice, a .zip archive containing 3 different folders. Each folder is associated with his own site. There will be the folder Default, www and Intranet.

- Each of these folders will have a very precise tree structure which will correspond to the final configuration of the Apache Server.

- Each of these folders will be composed of html css files for the display of the site in question, but will also have other folders such as log which will not be affected : this is only to correspond to the Apache configuration.

You will find in the manual the necessary commands to transfer files like these, but it will of course be re-specified in each chapter which ones will be added, where on the Raspberry and with which command all this accompanied by explanations on what you implement

But to start the manipulations we will ask you to check if there is a srv folder at the root of your raspberry and, if it's not the case, to create it and grant it the rights with the following command :

```
sudo mkdir -m 755 /srv
```

This folder will later contain the www and intranet folders. It is the folder where the Apache Web server will go to write logs, retrieve error pages if needed, and especially to display the html pages corresponding to all the sites.

We recommend decompressing the archive in question on your computer desktop, as the commands used to transfer the files will be defined assuming that the archive is decompressed on your desktop.

[1] OS Installation

In order to install the RPi OS Lite operating system on your Raspberry, I recommend that you follow the first steps in our user implementation guide that we provided earlier.
However, be careful here to select "Raspberry Pi OS Lite 64-Bits" instead of "Raspberry Pi OS".

[2] Internet connexion

IP address, DNS and router configuration

- Run this command to enter the configuration file :

```
sudo nano /etc/dhcpcd.conf
```

- Once you are inside the file, add these lines at the end of it :

```
static ip_address=10.192.51.244/16
static routers=10.192.0.255
static domain_name_servers=10.2.40.230
```

- Make sure to save the file before quitting, by pressing Ctrl+O, then Y, Enter and finally Ctrl+X.

Proxy configuration

- Run this command to enter the environment configuration file :

```
sudo nano /etc/environment
```

- Once you are inside the file, add these lines at the end of it, then save and quit :

```
http_proxy=http://username:password@wwwcache.univ-lr.fr:3128/
https_proxy=http://username:password@wwwcache.univ-lr.fr:3128/
```

Set the date and time correctly

- To set the date, simply run this command :

```
sudo date -s "YYYY-MM-DD HH:MM:SS"
```

To make sure that the configurations have been applied, you can restart the Rapsberry by running this command to set the date, simply run this command :

```
sudo reboot now
```

[3] Software installations

Apache

To install Apache 2, execute these 3 commands one after the other in the terminal :

```
sudo apt update
sudo apt upgrade
sudo apt install apache2
```

Check that the package has been correctly installed :

```
dpkg -l apache2
```

Make sure that the Apache service is started :

```
systemctl status apache2
```

PHP

To install the PHP module, run this command :

```
sudo apt-get install php libapache2-mod-php
```

Restart Apache to take into account the new configuration then restart the Raspberry by running these two commands in a row :

```
sudo systemctl restart apache2
sudo reboot now
```

[4] SSH connexion

If you want to connect to your Raspberry from another computer, for example to change the settings of your Apache web server, you can use SSH, which will allow you to access your Raspberry from any other machine connected on the same network.

To do this, first start the SSH service by running this command :

```
sudo systemctl start ssh
```

Then, enable SSH globally so that it will always be running on your raspberry

```
sudo systemctl enable ssh
```

You can now verify that SSH is correctly running with this command :

```
systemctl status ssh
```

Now that SSH is properly set up, you can connect to your Raspberry from another machine by using this command on the other computer and following the instructions in the terminal :

```
ssh username@10.192.51.244
```

[5] SFTP connexion

SFTP is a secure file transfer protocol that can be used to transfer files between a Raspberry Pi Lite OS and a physical machine. It will be essential in the further configuration to implement pre-configured files and make these manipulations simpler, more efficient and more secure.

To manipulate using this protocol, you will need to connect from a terminal on your physical machine to the Raspberry Pi using the following command (only if the previously verified SSH status showed that it was active) :

```
sftp username@10.192.51.244
```

Now we are going to see some commands that can be useful to you through this protocol, whether it is for your future use or for the rest of the configuration.

Postscript : Commands that will only be used in the terminal that you used on your physical machine to connect to the Raspberry.

- To send files from your machine to your Raspberry :

Here is an example to transfer a file named "test.png" from the account named "tekitizy" which is located in the "Images" folder of your physical machine to the "Images" folder of the Raspberry admin account :

```
put /home/tekitizy/Images/test.png /home/admin/Images
```

- And in the opposite direction, so to retrieve a file named test.png from the Raspberry to the images folder of our physical machine you can use :

```
get /home/admin/Images/test.png /home/tekitizy/Images
```

To summarize, the get and put commands will be useful to send or retrieve files between your raspberry and your physical machine.

- The syntax of a get command in SFTP from your physical machine is as follows :

```
get [remote_file_path] [local_file_path]
```

- The syntax of a put command in SFTP from your physical machine is as follows :

```
put [local_file_path] [remote_file_path]
```

Where **[remote_file_path]** is the path to the remote file (on the Raspberry Pi), and **[local_file_path]** is the path to the local file (on your physical machine) where you want to download the remote file.

If you want to transfer a directory between your physical machine and your Raspberry Pi via SFTP, you will need to use the "-r" option, which means that all files and directories located in the source directory will be transferred to the destination directory.

Here is the syntax and an example:

```
put -r [local_file_path] [remote_file_path]
get -r [remote_file_path] [local_file_path]
```

[6] Default website

[6.1] Site Implementation

First of all, for the configuration of the default web site, we will ask you to implement all the contents of the folder named "default" in the html folder which is located in the folder "/var/html" which is itself created automatically when Apache is installed. To do this use the following command :

```
put -r /home/<username>/default/* /var/www/html/
```

Don't forget that this command must be used in a terminal connected in sftp as explained in chapter 5, and don't forget to modify username by the name of your account.

You can now find the files that Apache retrieves to display your default site at "/var/www/html" with this tree structure :

```
html/
├── counter.txt
├── index.php
└── style.css
```

[6.2] Site configuration

WARNING : In the following command blocks, the sentences starting with "#" are not to be included in your code, they simply provide some explanations about what the associated command is used for and each line break represents a separate command

We recommend that you first configure the default website as it will serve to verify the implementation of this server and summarize the configuration with different file trees. It is also straightforward to configure, which will give you a quick answer to its proper functioning.

To start this configuration, you will need to edit, with the admin account, to the default configuration file using this command :

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

Once in the text editor, you will have to add some directives that we will explain right after.

For this purpose, we provide you with the entire contents of the file below so that you can copy/add the missing parts of the code :

```
<VirtualHost *:80>
     ServerName raspb244.univ-lr.fr
     DirectoryIndex perso.php index.html perso.html index.php
     ServerAdmin webmaster@localhost
     DocumentRoot /var/www/html

     ErrorLog ${APACHE_LOG_DIR}/error.log
     CustomLog ${APACHE_LOG_DIR}/access.log combined
 </VirtualHost>
```

You can now exit the file using the keyboard shortcuts "CTRL+X" to exit and "Y" to save your changes before exiting.

Here is the list of changes made to the file and what it corresponds to :

- ServerName directive changed by raspb244.univ-lr.fr
  - to specify the domain name used to access the server

- Added DirectoryIndex directive with a file list
  - to specify the file names to be used as the home page of the site

[6.3] Site customization

So you have your default site configured and accessible from the Internet at the desired address: raspb244.univ-lr.fr

During the implementation with the sftp command in the "/var/www/html" folder, you have integrated the different files that Apache accesses to display the default site. If you want to modify it to add your own code you can then follow one of these 2 steps :

- To only modify the existing page, use a text editor to access this page and do what you want with this command :

```
sudo nano /var/www/html/index.php
```

- If you want to implement your own default page, we advise you to import the necessary files with an SFTP connection but first of all delete the contents of the "/var/www/html" folder with this command :

```
sudo rm -rf counter.txt index.php styles.css
```

You can now add your files or simply create your own (remembering to name the main page "index") with this command :

```
sudo nano /var/www/html/index.<html/php>
```

[6.4] Upload personal pages

Your default site address will also allow you to view and publish personal pages for each user using an address that varies depending on the user such as : *raspb244.univ-lr.fr/~<username>*

But to do this, you will first need to activate the Apache module that allows this using this command :

```
sudo a2enmod userdir
```

Once done, you will need to verify that your userdir.conf file is similar to the one below. To view it, run :

```
sudo nano /etc/apache2/mods-available/userdir.conf
```

Compare what you have with the file displayed below. If necessary, modify your file in the text editor that opened and exit/save using the keyboard shortcuts mentioned earlier. (You shouldn't normally have to add or modify anything, but still check it!)

```
<IfModule mod_userdir.c>

     UserDir public_html
     UserDir disabled root

     <Directory /home/*/public_html>
          AllowOverride FileInfo AuthConfig Limit Indexes
          Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
          Require method GET POST OPTIONS
     </Directory>

</IfModule>
```

Once you have followed these steps, you just need to add a folder named "*public_html*" at the root of your user and place the HTML pages that the user wants to publish on the default site.

Here are the steps to navigate to your user's root directory and create a "*public_html*" folder :

```
cd #To go to the root directory of the currently logged in user
sudo mkdir public_html #To create the directory
sudo chmod 755 public_html #Assign certain access rights
cd public_html #To go where you need to add your html pages.
```

By default, if you have only one page in the "*public_html*" directory, the address "**raspb244.univ-lr.fr/~<username>**" will work. Otherwise, you just need to specify the name of your page in the address. Here is an example : raspb244.univ-lr.fr/~<username>/<index.html>

Once all these manipulations are done, you just need to restart the Apache service on your machine so that the modifications made are updated, and this can be done using this command:

```
sudo service apache2 restart
```

[7] Public website

[7.1] Site Implementation

First of all, for the configuration of the public website, we are going to ask you to implement the whole folder named "www" in the srv folder which is located at the root of your Raspberry and which you must have created during the preamble. To do this use this command :

```
put -r /home/<username>/Desktop/notice/www /srv
```

Don't forget that this command must be used in a terminal connected in sftp as explained in chapter 5, and don't forget to modify username by the name of your account.

Now you can find at the root of your raspberry, in the srv folder, a www folder with this tree structure :

```
www
├── log
│   ├── acces
│   │   └── access.log
│   └── erreur
│       └── error.log
└── public
    ├── arborescence
    │   └── you_can_see_this_file.txt
    ├── erreurs
    │   ├── 401.html
    │   ├── 403.html
    │   └── 404.html
    ├── index.html
    ├── more-mission.html
    ├── more-presentation.html
    ├── styles.css
    └── testfolder
```

[7.2] Site configuration

We strongly recommend that you set up your public site second. The public site is the window of your company that will be visible on the Internet by anyone, it aims to highlight : your missions, your team, your talents, your contacts and more

To start this configuration, you will need to create, with the admin account, the public configuration file using this command :

```
sudo nano /etc/apache2/sites-available/site-public.conf
```

Once in the text editor, we advise you to paste the following text, which is a configuration that meets the needs you have described :

```
<VirtualHost *:80>

        ServerName www.tek-it-izy.org
        DocumentRoot /srv/www/public
        DirectoryIndex index.html accueil.html

        ErrorLog /srv/www/log/erreur/error.log
        CustomLog /srv/www/log/acces/access.log combined

        ErrorDocument 404 /erreurs/404.html
        ErrorDocument 403 /erreurs/403.html
        ErrorDocument 401 /erreurs/401.html

        <Directory /srv/www/public>
                Options Indexes FollowSymLinks
                AllowOverride
                Order allow,deny
                Allow from all
                Require all granted
        </Directory>
</VirtualHost>
```

You can now exit the file using the keyboard shortcuts "CTRL+X" to exit and "Y" to save your changes before exiting.

In this configuration file we find mainly the different parameters applied to the site, defined in the Directory tag, but we also find :

- The domain name of the server defined by the ServerName directive
- The root directory where HTML, CSS and other resources of a website are stored by the DocumentRoot directive
- The root directory where the location server access and error messages will be stored by the ErrorLog and CustomLog directive
- The path to a custom page to display when an HTTP error occurs by the ErrorDocument directive

You will notice that all these paths/directories defined in this configuration file exist ! They have been automatically added during the sftp transfer at the beginning of this chapter, and that we find visually on the tree structure

Once all these manipulations are done, don't forget to restart again the Apache service on your machine so that the modifications made are updated, and this can be done using this command:

```
sudo service apache2 restart
```

[7.3] Site customization

So you have your public site configured and accessible from the Internet at the desired address: www.tek-it-izy.org

But in case you want to modify the public site from a display/style point of view either HTML/CSS you will have to delete the HTML files in the www folder and implement your own or recreate them directly. All this can be done with these different commands : (next page)

- Delete all the files related to the page we have created according to your needs :

```
cd /srv/www/public #Go to the folder to delete
sudo rm -rf arborescence *.html styles.css testfolder
```

- If you want to implement your own, use the SFTP commands described above, otherwise create your own files with this command :

```
sudo nano index.html
```

Remember to create at least one page named index.html, which will be the first one displayed as described in the configuration file of the public site in the DirectoryIndex directive. Otherwise, you can modify it according to your needs.

[8] Intranet website

[8.1] Site Implementation

First of all, for the configuration of the intranet website, we are going to ask you to implement the whole folder named "intranet" in the srv folder just like the public website seen before :

```
put -r /home/<username>/Desktop/notice/intranet /srv
```

Now you can find at the root of your raspberry, in the srv folder, a www folder with this tree structure :

```
intranet
├── arborescence
│   └── youcantseethisfile.txt
├── erreurs
│   ├── 401.html
│   ├── 403.html
│   └── 404.html
├── index.html
├── log
│   ├── acces
│   │   └── access.log
│   └── erreur
│ .     └── error.log
└── testfolder
```

[8.2] Site configuration

For us, your intranet website needs to be configured last. Indeed this site is secured by a password and may contain some confidential informations about your start-up. It aims to give your employees a means of private communication, file transfer, project management and more.

To start this configuration, you will need to create, with the admin account, the intranet website configuration file using this command :

```
sudo nano /etc/apache2/sites-available/site-intranet.conf
```

Once you're in the text editor, we advise you to copy & paste the following text, which is a configuration that meets the needs you have described :

```
<VirtualHost *:80>
     ServerName intranet.tek-it-izy.org
     Redirect / http://intranet.tek-it-izy.org:2080
</VirtualHost>

<VirtualHost *:2080>
    ServerName intranet.tek-it-izy.org
    DocumentRoot /srv/intranet
    DirectoryIndex index.html intranet.html index.php
    UserDir disabled
    ErrorLog /srv/intranet/log/erreur/error.log
    CustomLog /srv/intranet/log/acces/access.log combined

    ErrorDocument 404 /erreurs/404.html
    ErrorDocument 403 /erreurs/403.html
    ErrorDocument 401 /erreurs/401.html

    <Directory /srv/intranet>
        Options Indexes FollowSymLinks
        AllowOverride
        Order deny,allow
        Deny from all
        Require all granted
    </Directory>

    <Location />
        AuthType Basic
        AuthName "Zone restreinte"
        AuthBasicProvider file
        AuthUserFile /etc/apache2/passwords
        Require valid-user
    </Location>
</VirtualHost>
```

You can now exit the file using the keyboard shortcuts "CTRL+X" to exit and "Y" to save your changes before exiting.

You may have noticed that this configuration file has some similarities with the previous one, but there are still many differences that change everything.

First of all, the listening port is set to 2080 which is convenient to connect from another network than tek-it izy and more secure.
The <VirtualHost *:80> tag is only used to redirect users to port 2080 so you don't have to type **:2080** at the end of the ***intranet.tek-it-izy.org*** url.

We need to verify that the apache server is listening on both the default port and port 2080. To do this, we will edit the apache 2 ports configuration file by running this command :

```
sudo nano /etc/apache2/ports.conf
```

- Once in the file, add "**Listen 2080**" to the end of the file, then save and quit with "Ctrl + X" and "Y".

In the configuration file of the intranet website, you will find a **<Location />** tag, it defines the type and name of the authentication but, most importantly, the file containing the usernames and passwords of valid users thanks to the **AuthUserFile** directive.

This file, does not exist yet, and we will create it and add a user to it by running this command :

```
sudo htpasswd -c /etc/apache2/passwords intranet
```

- You will be asked to type twice the password you want for the user "intranet" to confirm the addition of it in the file.

You can now check the passwords file with the command below, you will see the username you added with his encrypted password.

```
sudo cat /etc/apache2/passwords
```

- If you want to add a new valid user to the file, you just have to run :

```
sudo htpasswd /etc/apache2/passwords username
```

- To delete a user, simply add "-D" between the command name and the file path.

Once all these manipulations are done, don't forget to restart again the Apache service on your machine so that the modifications made are updated, and this can be done using this command:

```
sudo systemctl restart apache2
```

[8.3] Site customization

So you have your intranet site configured and accessible from the Internet at the desired address: intranet.tek-it-izy.org

But in case you want to modify it, simply follow the same steps that for the public website. But to delete the intranet files instead of the public ones, use this command :

```
cd /srv/intranet #Go to the folder to delete
sudo rm -rf arborescence index.html testfolder
```

[9] Appendices

[9.1] Distribution of tasks in time

Week	Goal	Thibaut	Colin
S8	Finish the preparation work	Individual working (helping each other sometimes but not much)	
S9	First Milestone, Research and comparison of solutions	Research work and table preparation	Research
S10			Creation of the canva's pdf
S11	Start of the second Milestone : Web hosting with Apache 2. -OS Installation -Internet connection -SSH Connection	We were both searching and trying things on the raspberry	
S12	Configuration of the Default and public website	-User pages configuration -Error pages configuration	-Public website configuration
S13	-Configuration of the intranet website -HTML/CSS and PHP Pages	-Code for the intranet pages and the error pages	-Intranet configuration -Public and default site code
S14	-Final touches -Final delivery	-We prepared together the course of our presentation	
S11 to S14	Redaction of the User Notice	-Throughout our project, we kept all traces of what we did on our Raspberry in a Miro Board -This helped us to create our leaflet that we made on Canva as we went along in the project	

[9.2] Preparation work (TP SAE)

As requested, the links to our SAE2.03 project preparation work :

- Colin : https://bit.ly/tp-sae203-colin
- Thibaut : https://bit.ly/tp-sae203-thibaut