SOAN GVOKA
JOEY JUAN
AHOUSSI NATHAN

Report of the Project for the Installation of a Development Station

Project of S1- S103

On the first day of the project, January 5th, we had to choose an operating system to install on our Raspberry Pi. From there, our first objective and idea were **to install Arch Linu, but we encountered problems with partition formatting despite the SD card being in the correct format, that is, FAT32. After multiple attempts, we concluded that there may be compatibility issues with the Raspberry, which took up almost the entire first session. Therefore, with this information in mind, we settled on the Raspberry Pi OS operating system.**

After installing it on the first computer, we first flashed the provided SD card and then extracted the OS directly into our PC using the terminal. We then launched Raspberry PI Imager to install the OS on the SD card.

To install the OS on the SD card, we had to find the OS file in the computer's folders, using “rpi-imager” **we just had to choose the SD card as the storage for the system and launch the writing process, but it required a password for Mass Storage Device authentication. To bypass this, we used the command "sudo rpi-imager" to give ourselves the necessary rights.**

After the writing and installation on the SD card was complete, we inserted it into the Raspberry Pi. Unfortunately, this was the end of the first session. During the next session, **we were finally able to see the Raspberry Pi interface, we wanted to connect to Google using the already installed Google Chrome, but we had not configured the time and date for the computer.
To fix this, we used the terminal and typed in the command "date -s "MM/DD/YYYY HH:MM:SS" then we began setting up the Apache server to host our sites from the Raspberry Pi and be able to access them from another computer using only the IP address and site address.**

To do this, we first had to retrieve our respective COIN project sites, so we used a computer with Windows so as not to have to install FileZila on the Raspberry. After retrieving our sites on FileZila, we had to send them to the Raspberry, to do this we sent the site files to Google Drive and then retrieved them on the Raspberry.

After retrieving the sites, we had to move them to the sites-available file and use the commands "sudo a2ensite" and "sudo a2enconf" to create symbolic links in the enabled directories. Thanks to this, Apache then took into account the relevant configuration files after reloading the server using the command "sudo systemctl reload apache2".

After that, we had to configure the listening ports in the ports.conf file that we found in /etc/apache2/ports.conf, which we had modified earlier but restored after activating the SSL module by running the command "sudo a2enmod ssl" and reloading the Apache server with "sudo systemctl reload apache2".

We then looked at the 000-default.conf file located in /etc/apache2/sites-available/000-default.conf, but ultimately did not modify anything except the ServerAdmin section where we put our respective student emails for each of our sites.

Then, we created virtual hosts for each of our 3 sites in, for example, the file /etc/apache2/sites-available/example.com.conf, where we changed the ServerName, ServerAlias corresponding to our sites and the DocumentRoot to allow our site files to have a directory.

Unfortunately, at this point we were still on localhost so the sites were only visible on the local machine where the Rasberrypi was. To fix this, we had to modify the host in the file host located in /etc/hosts to add the IP address of the server followed by the port that we had assigned to each of the sites. This should have looked something like
"127.0.0.1 [the corresponding address]**"**

After that, we restarted the Apache server and it worked, all that was left was to decorate the index.html page by linking our sites. Unfortunately, in the next class we were no longer able to restart the Apache server.

After several attempts, we finally managed to fix it.

We had a problem during the installation of the last site because we gave ourselves all rights in the files and not just in the files we wanted, so we had to start over from the beginning because it was now impossible for us to use the "sudo" command. But fortunately, starting over only took us 10 minutes to get back to where we were, that is, putting the third site, making it look a little better and finally adding a hub page in the port 80 location that allows to redirect to all the available sites.

Then the bigger problem arrived. TLS didn’t want to work. It just refused to. After troubleshooting a lot of error messages that where given by “apachectl configtest” and “systemctl status apache2” and trying a lot of different tutorials to configure the ssl certificate. Today, 19/01/2023, we completely erased apache and reinstalled it and reconfigured everything and redid the ssl certification and it finaly worked and with it the password also worked (because we only configured it on port 443) those where the commands today to finally have SSL:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/toto.key  -out /etc/ssl/certs/toto.crt
```

(to create the key and the self-signed certificate)

```
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:10.42.0.2
Email Address []:
```

(Only filled the important information)

```
sudo a2enmod ssl (to enable ssl on the apache server)
sudo a2enmod socache _shmcb (just in case it wasn’t automatically done with ssl)
sudo systemctl restart apache2 (to restart apache)
```

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

```
(And we modified these to lines to become :)
SSLCertificateFile  /etc/ssl/certs/toto.crt
SSLCertificateKeyFile /etc/ssl/private/toto.key
(So they point to the correct location of the key and the certificate)
```

And to enable the login to access the https server we did this :

```
sudo apt install apache2-utils
sudo htpasswd -c /etc/apache2/.htpasswd toto
(the password we chose was): tutu
sudo nano /etc/apache2/sites-available/default-ssl.conf
(This has been added to the file) :
<Directory "/var/www/html">
    AuthType Basic
    AuthName "Restricted Content"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
  </Directory>
sudo apache2ctl configtest (To make sure the syntax was ok)
sudo systemctl restart apache2
```

So in the end we redid everything at the last second because of a random bug we couldn’t troubleshoot.

All usernames used were: toto
All passwords used were: tutu

We also found a solution to sync the date wich was to add a command to the .bashrc file

```
sudo date -s "$(wget -qSO- --max-redirect=0 google.com 2>&1 | grep Date: | cut -d' ' -f5-8)Z"
```