DEBIAN 12 SERVEUR INSTALLATION GUIDE

INSTALLATION GUIDE FOR A DEBIAN 12 SERVEUR WITH APACHE, POSTRESQL AND PHP

AUTHOR

GIBELLO GRÉGOIRE
BUT-1A − E1

Table des matières

1 - Debian Installation

1.1 - Introduction

The aim of this guide is to install and run a Debian 12 **without graphical interface** equipped with an Apache server, PostgreSQL and PHP that can be queried from the host machine.

We will detail all the steps with screenshots, as well as all the shell commands that need to be entered in order to complete this installation.
This installation was carried out as part of the SAE 2.03 project carried out in the 2nd semester of the first year of the BUT computer science department.

1.2 - System Requirements

1. **Operating System**:
    - A Linux distribution that supports QEMU/KVM. The following versions are compatible:
        - Ubuntu 20.04 LTS or later
        - Fedora 33 or later
        - Debian 11 (Bullseye) or later
        - CentOS 8 or later
        - Arch Linux (latest version)
2. **Processor**:
    - A 64-bit x86 processor

Note :
For this guide, we will install the OS on a Qemu/KVM virtual machine running Debian GNU/Linux 12 ("bookworm").

1.3 - Prepare for installation (ISO image)

An ISO image is required for installation on the virtual machine as it contains a complete copy of the Debian 12 installation media.

ISO image :
- This image allows the virtual machine to boot and install the operating system directly as if you were installing it on your own PC from a CD or USB key.

- you have to check its integrity to make sure it will work properly

The ISO image has been pre-installed in the /usr/local/images-ISO/ directory, but the files needed to check its integrity are available here:

https://cdimage.debian.org/cdimage/release/current/amd64/iso-cd/

A problem ?

If you have a problem at this stage, please refer to "Troubleshooting", problem N°1.

1.4 - Debian system installation

Debian 12 is installed on a virtual machine. Under Linux, you can launch the VM installation with this command :

S2.03-lance-installation

The parameters used to launch QEMU/KVM behind this command are as follows :

- `qemu-system-x86_64` : Specifies the QEMU binary for 64-bit x86 architectures.

- `-m 4G` : Allocates 4 GB of RAM to the virtual machine.

- `-enable-kvm` : Enables KVM (Kernel-based Virtual Machine) support for hardware acceleration.

- `-device VGA,xres=1024,yres=768` : Configures a VGA display device with a resolution of 1024x768.

The rest of the settings configure the hardware, networking with port forwarding, and virtual machine display settings.

Follow each step of the installation process. When not specified, choose the default option. Here are the main steps:

IMPORTANT !
The most crucial choice is to install Debian without a graphical interface. Don't miss that !

1. **Language** : English

2. **Location** : other/Europe/France

3. **Locales** : United States, en_US.UTF-8

4. **Keyboard** : French

5. **Hostname** : use server-"YOUR_UGA_LOGIN"

6. **Root Password** : A simple password is recommended, such as "root". In this context, it does   not pose a security issue.

7. **User Account - Full Name** : Your full name
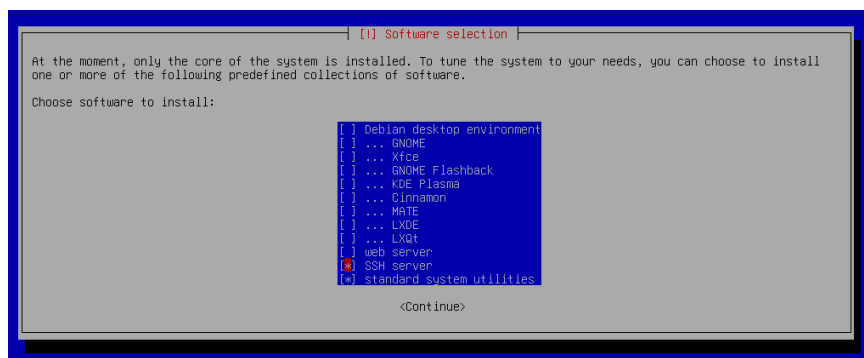
8. **User Name** : Your UGA login name
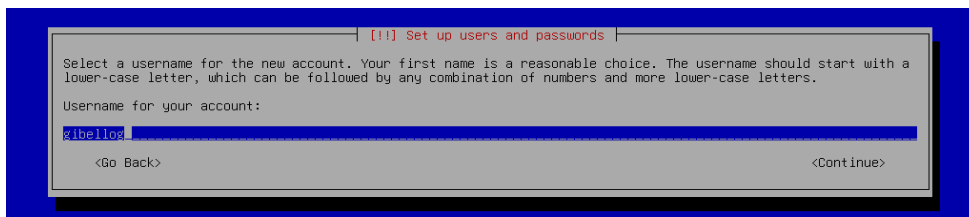
9. **User Password** : A simple password, such as "etu".

10. **Partition disks :** Guided - use entire disk

11. **Partition disks :** All files in one partition

12. **Partition disks :** Yes

13. **Software Selection** : Ensure "Debian desktop" is not checked and "ssh server" is checked

14. **Install GRUB :** Yes

15. **Device for boot loader** : /dev/sda

When the installation is complete, use the following command as root:

```
# poweroff
```

This is because you have to restart the virtual machine for finish the installation and continue this guide.

Note :
This is the command you need to use to cleanly shut down your virtual machine after each use. Don't shut it down in any other way!

There's just one last thing:

The file is stored locally on your Linux workstation, so to access it more easily in the future, check that the VM is switched off and then run this command :

```
S2.03-déplace-image-disque-sur-erebus4
```

That's it!

The installation is now complete, and to access your virtual machine you can type :

```
S2.03-lance-machine-virtuelle
```
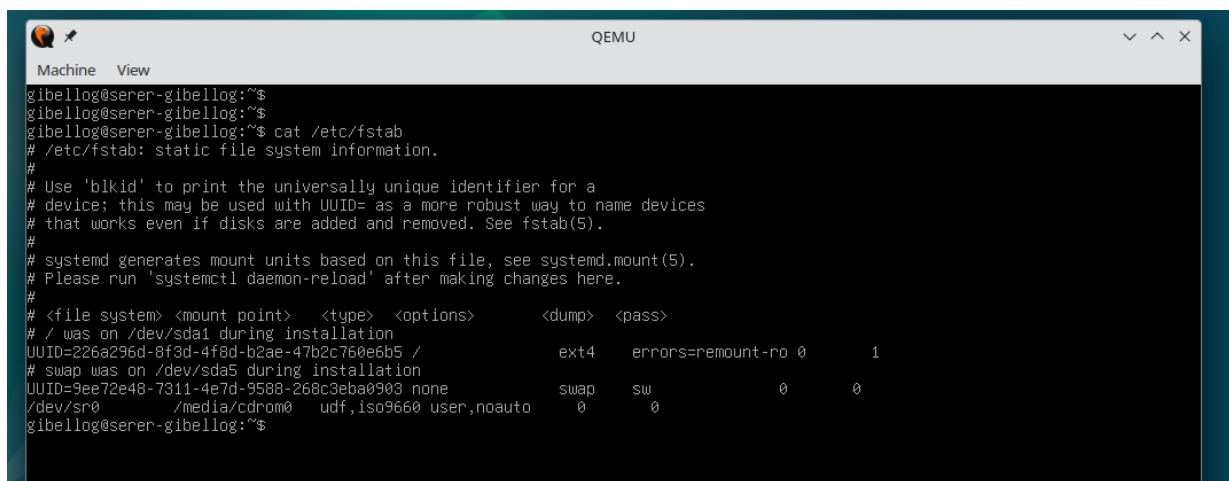
2 - Post-Installation Verification

Before continuing, let's do a few checks to make sure that the installation of Debian 12 has been well doned :

1.2 – Disk partitions check

First, check the file system table to ensure that the disk partitions are correctly configured. This can be done by running the following command in your virtual machine:

cat /etc/fstab

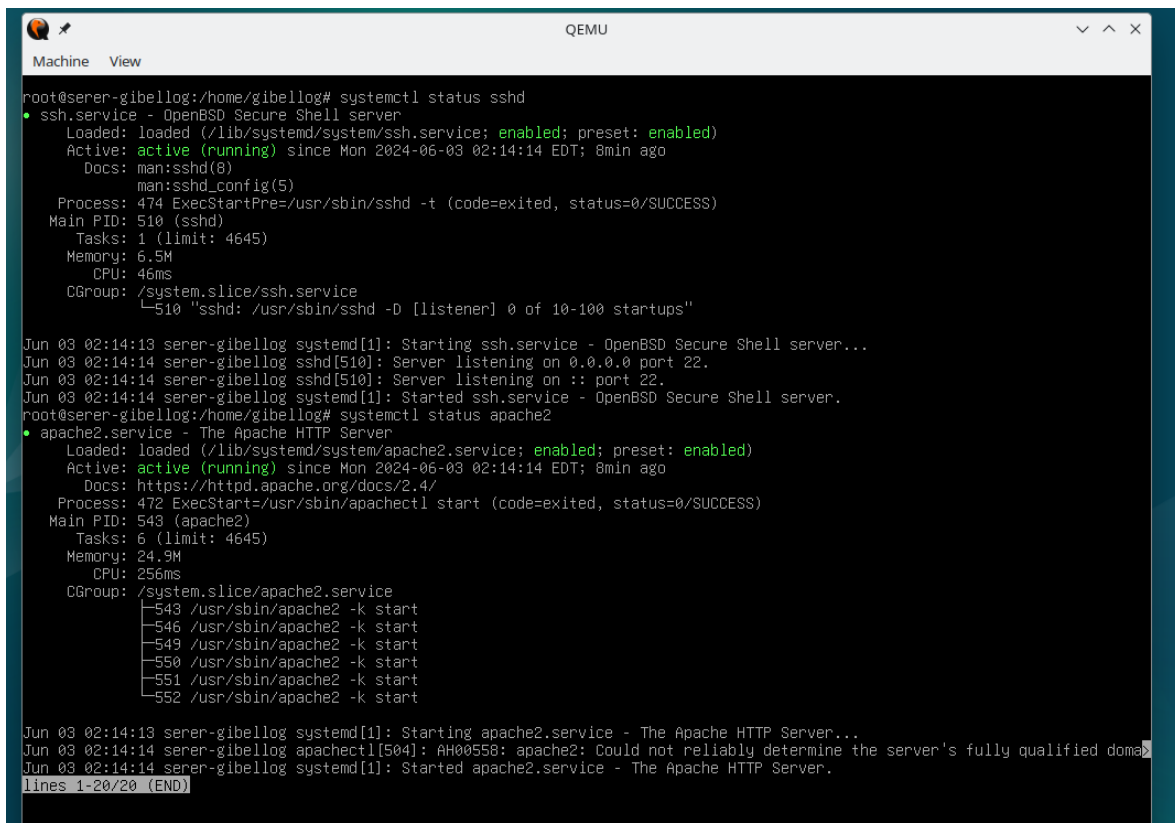The resulting file shows how the disk partitions are mounted at boot time :

1.2 - Check the status of essential services

Next, check that the SSH, Apache and PostgreSQL services are running correctly. Use the following command and adapt them to check the status of each service :

systemctl status [ssh or apache2 or postgresql]

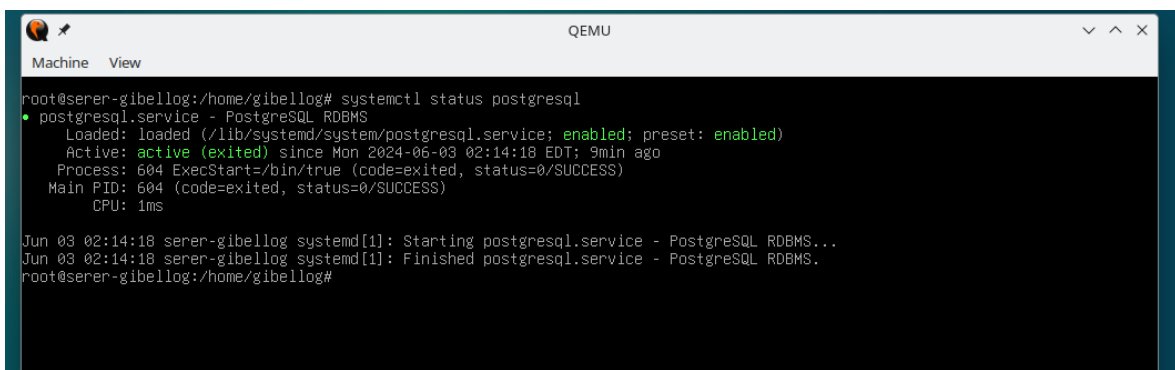If all went well, you should get this:

Info :
The systemctl status command is use for verifying the status of services on your Debian system. It provides detailed information about if a service is active and running, or if there are any errors.

1.2 - External connectivity Check

Check that the VM is connected to outside with the command :

ping pc-dg-033-01

If the ping is successful, it indicates that you have access to the external network.

Tip :
If you want to display the network configuration, use : ip addr

That's all for this part of the checklist. If all went well, you can move on to the next stage of our guide!

A problem ?

If you have a problem somewhere at this all stage, please refer to "Troubleshooting", problem N°2.

3 - Apache, PostgreSQL and PHP Installation

IMPORTANT !
For every installation, we use the APT (Advanced Package Tool) package manager and the packages provided by the Debian distribution.

3.1 - Apache installation

First, launch your virtual machine if it is not already running :

S2.03-lance-machine-virtuelle

Then connect as root, and install Apache2 with APT.

APT :
- To install with the Advanced Package Tool (APT), as a root do : apt install [software name]

Here the right command is :

 # apt install apache2

Use the following command to check if Apache has been started :

 # systemctl status apache2

If it is not, enter:

 # systemctl start apache2

Last check!

To check that the virtual machine's web server is working correctly, we're going to display the web page. Although it is impossible to display a web page directly on the virtual machine, we can do so from the host machine by redirecting a port on the host machine (for example, port 8080) to port 80 on the virtual machine (the default port for web servers).

This redirection is implemented by the launch script provided. So, by opening a web browser on the Linux host machine and accessing the URL http://localhost:8080, you should see the default page for the virtual machine's Apache server.
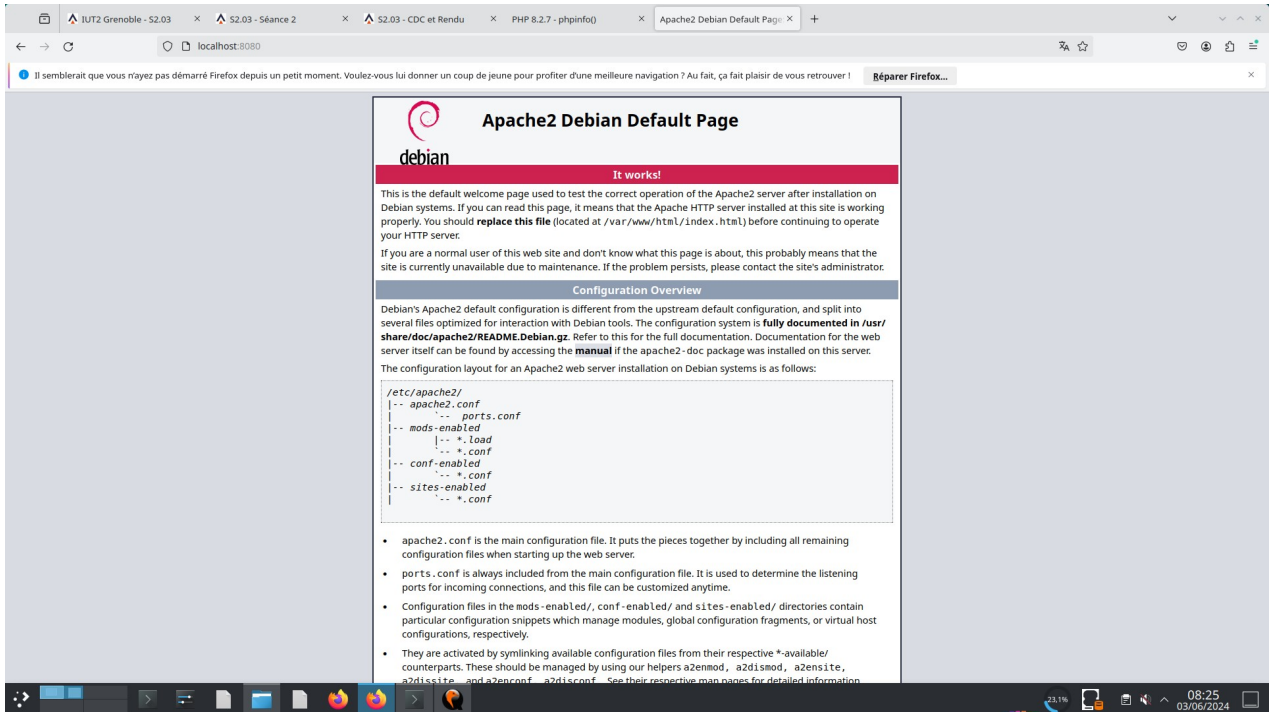
3.2 - PostgreSQL installation

To install it, use this command as you did before:

 # apt install postrgesql

When the postgresql package was installed, a new unix account was created: postgres. Connect to this account (run from a root shell) to check the installation :

 # su - postgres
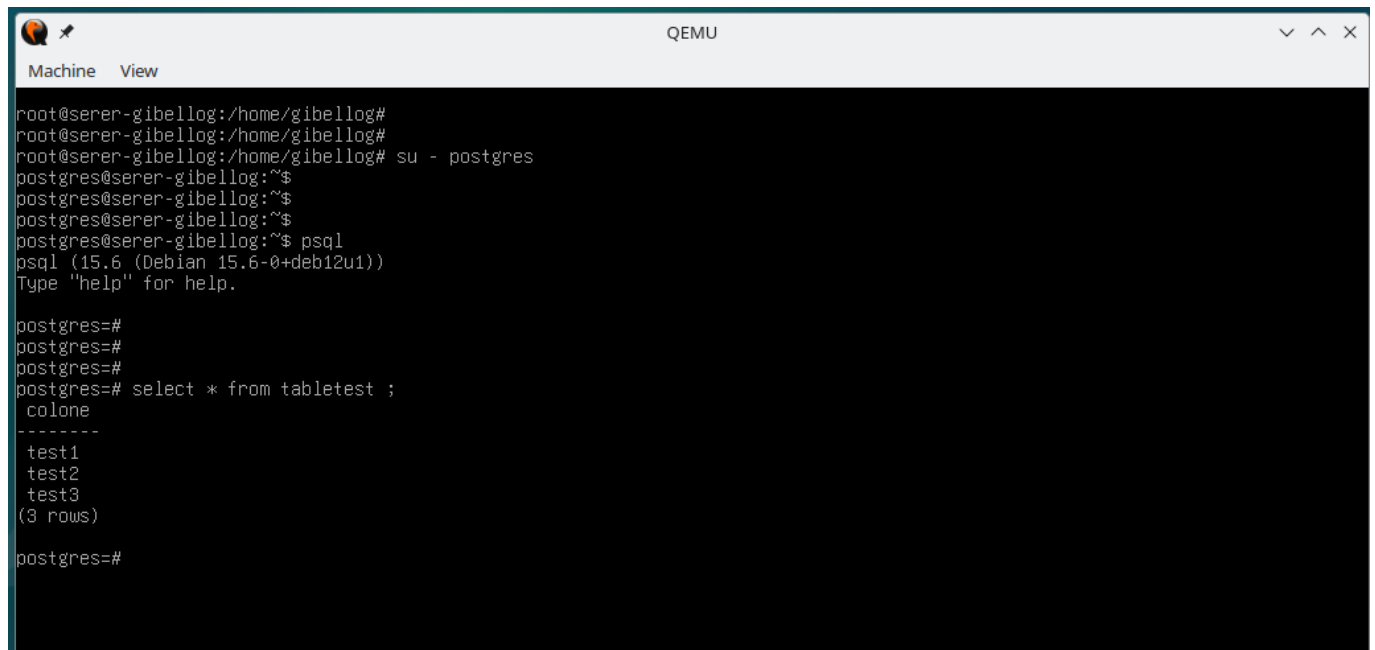
Now, lets create a test table, enter this command and create a psql user with your login:

psql

You can now create a database whose owner is your user by entering :

CREATE DATABASE [Database name] OWNER [User login] ;

After that, you can use « CREATE TABLE [Name]; » to create a table and « INSERT INTO [Table] [Data]; » to insert rows.

```
root@serer-gibellog:/home/gibellog#
root@serer-gibellog:/home/gibellog#
root@serer-gibellog:/home/gibellog# su - postgres
postgres@serer-gibellog:~$
postgres@serer-gibellog:~$
postgres@serer-gibellog:~$
postgres@serer-gibellog:~$ psql
psql (15.6 (Debian 15.6-0+deb12u1))
Type "help" for help.

postgres=#
postgres=#
postgres=#
postgres=# select * from tabletest ;
 colone
--------
 test1
 test2
 test3
(3 rows)

postgres=#
```

You can check that the new database has been created with :

```
gibellog@postgres=> SELECT datname AS database_name, pg_catalog.pg_get_userbyid(datdba) AS owner
FROM pg_catalog.pg_database;
 database_name |  owner
---------------+----------
 postgres      | postgres
 test          | gibellog
 template1     | postgres
 template0     | postgres
(4 rows)
```

Verify Password Encryption :

List the contents of the pg_shadow system table to verify that passwords are hashed using SHA-256. To do this, use :

SELECT * FROM pg_shadow;

```
postgres@serer-gibellog:~$ psql
psql (15.6 (Debian 15.6-0+deb12u1))
Type "help" for help.

postgres=# SELECT * FROM pg_shadow;
 usename  | usesysid | usecreatedb | usesuper | userepl | usebypassrls |
       passwd                                                           | valuntil | useconfig
----------+----------+-------------+----------+---------+--------------+-------------------------------------------------------------------+----------+----------
-------------------------------------------------------------------------------------------+----------+----------
 postgres |       10 | t           | t        | t       | t            |
                                                                       |          |
 gibellog |    16388 | t           | f        | f       | f            | SCRAM-SHA-256$4096:L/nKSeZHqMx6vMlEd+XZxg==$KbZ8wwsOn09
ZW/R+hXdh84bQXKg3xL+s9dgvd0pKgmU=:07C+hKPX6CyGrr+KTBwAwQyOk6ZxrdIzf/HWkBGKhUY= |          |
(2 rows)

(END)
```

Ensure that the passwords are hashed with the SCRAM-SHA-256 algorithm like you can see on this screenshot.

Now, Remember to exit psql with : "exit"

Configuring PostgreSQL to be accessible from your Linux workstation :

To do this, we're going to modify a few lines in some files. Don't worry, we'll take you through all the steps involved.

First, from your virtual machine, edit the configuration file, for example with nano :

```
# nano /etc/postgresql/15/main/postgresql.conf
```

In the « CONNECTIONS AND AUTHENTICATION » section, find the following line to uncomment and update :

```
listen_addresses = '*'
```

IMPORTANT !
Don't forget to save with ctrl+O and ctrl+X to exit.

Now that the server is listening for connection requests from non-local IP addresses, we need to define an authentication rule that will be used for these requests. To do this, edit the authentication rules file :

```
# nano /etc/postgresql/15/main/pg_hba.conf
```

and add the following rule to allow only connections authenticated by a password stored with a strong hash function:

```
#IPv4 remote connections:
host  all all 0.0.0.0/0 scram-sha-256
```

Validate these new configurations by restarting your server with this command :

```
# service postgresql restart
```

Almost done !
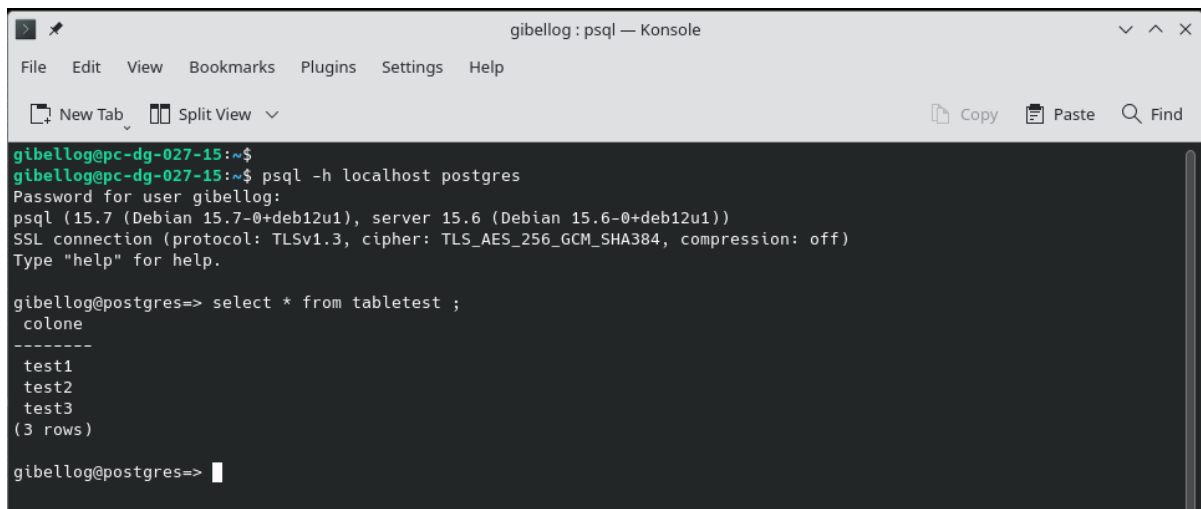
Connect from a terminal on your station, again using psql :

psql -h localhost postgres

Try to ask your database's table with :

SELECT * FROM [table name] ;

You should see this :

If this works, the PostreSQL installation is complete. We can move on to the next step!

3.3 - PHP installation

We will now install PHP on the virtual machine.

> **Info :**
> PHP is a widely-used scripting language for web development.

To install php on your virtual machine, we'll use APT as usual:

 # apt install php

After installing PHP, restart the Apache2 service to apply the changes.

 # /etc/init.d/apache2 stop

and

 # /etc/init.d/apache2 start

Now we'll test the PHP installation by placing a file named info.php in the /var/www/html/ directory. To do this, modify the file as you would during PostgreSQL installation:

 # nano /var/www/html/info.php

And enter the following code :

```
<?php
phpinfo();
phpinfo(INFO_MODULES);
?>
```

> **IMPORTANT !**
> Don't forget to save with ctrl+O and ctrl+X to exit.

After that, access http://localhost:8080/info.php from the host machine. A page containing the main features of your PHP installation should appear.

3.4 - PHPpgAdmin installation

Now install PHPpgAdmin on the virtual machine by using APT.

> **Info :**
> PhpPgAdmin is a web-based administration tool for managing PostgreSQL databases.

Once installed, we have to configure Apache for PhpPgAdmin to make it accessible via your web server.

Open the Apache configuration file for PhpPgAdmin as a root:

```
# nano /etc/apache2/conf-enabled/phppgadmin.conf
```

We need to add a few lines to configure PHPpg properly. Add these lines of code directly to the file:

```
Alias /phppgadmin /usr/share/phppgadmin

<Directory /usr/share/phppgadmin>
   <IfModule mod_dir.c>
      DirectoryIndex index.php
   </IfModule>
   AllowOverride None

   # Only allow connections from localhost:
   #Require local

   <IfModule mod_php.c>
     php_flag magic_quotes_gpc Off
     php_flag track_vars On
     #php_value include_path .
   </IfModule>
   <IfModule !mod_php.c>
      <IfModule mod_actions.c>
        <IfModule mod_cgi.c>
          AddType application/x-httpd-php .php
          Action application/x-httpd-php /cgi-bin/php
        </IfModule>
        <IfModule mod_cgid.c>
          AddType application/x-httpd-php .php
          Action application/x-httpd-php /cgi-bin/php
        </IfModule>
      </IfModule>
   </IfModule>
</Directory>
```

> **IMPORTANT !**
> Don't forget to save with ctrl+O and ctrl+X to exit.

After that restart the Apache2 service to apply the changes like as before.

You can now access PhpPgAdmin via your web browser. Open the following URL in your browser: http://localhost:8080/phppgadmin/

If its working, you can acces to your SQL database and even select the table you created before, you should see a page like this one:
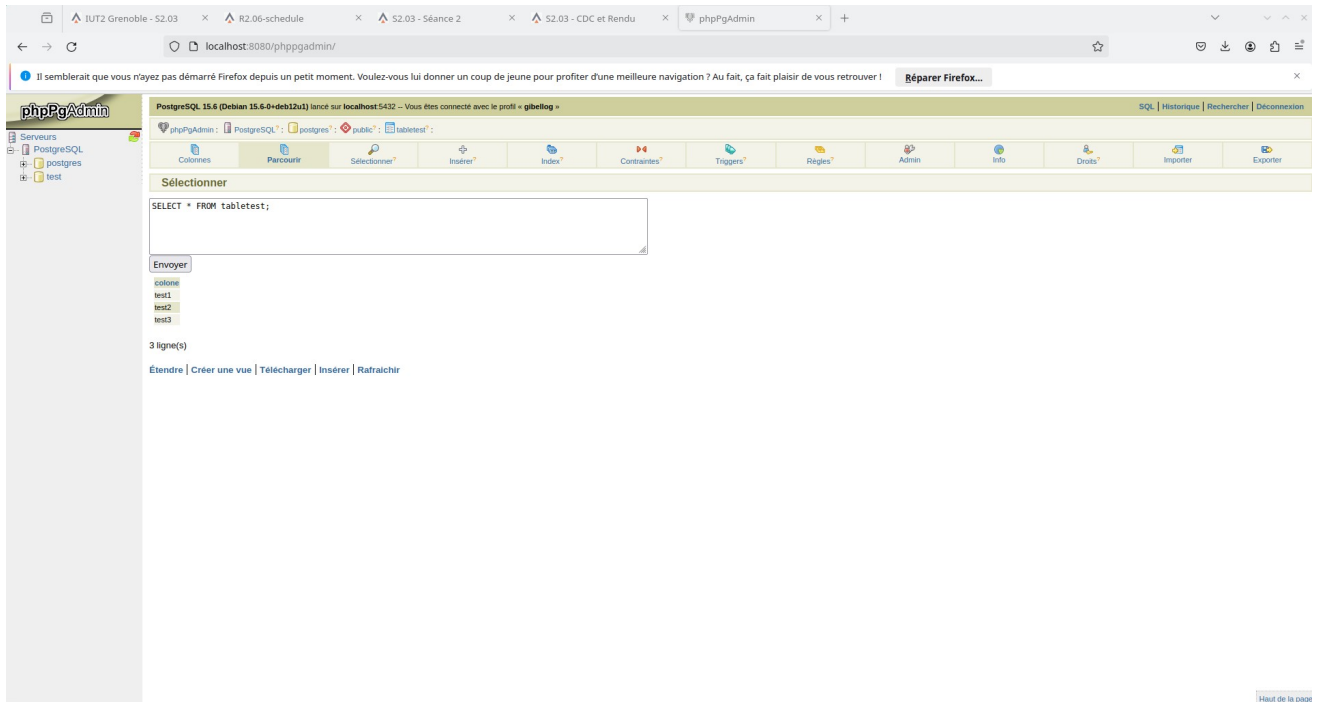
A problem ?

If you have a problem somewhere at this all stage, please refer to "Troubleshooting", problem N°3.

3.5 - To conclude

To conclude these installations, here is a good security tip for correcting software security flaws that may appear in Debian, Apache, PHP or PostgreSQL:

Update your Debian packages regularly. You can start by updating the list of available packages with :

```
# apt update
```

Update installed packages with :

```
# apt upgrade
```

And don't forget to clean up any unnecessary files with :

```
# apt clear
```

This last command deletes downloaded cache files, freeing up disk space. For example, you can check your remaining disk space with :

```
# df -h
```

You should obtain this :

```
root@serer-gibellog:~#
root@serer-gibellog:~#
root@serer-gibellog:~#
root@serer-gibellog:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           392M  488K  392M   1% /run
/dev/sda1       3.0G  1.6G  1.3G  56% /
tmpfs           2.0G  1.1M  2.0G   1% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           392M     0  392M   0% /run/user/1000
root@serer-gibellog:~#
```

This installation allows us, for example, to host web pages or PHP files such as this one, which can be retrieved from your Linux workstation at : /users/info/www/intranet/enseignements/S2.03/page_sae_S2.03.php

Then install it on our virtual machine's Apache server after executing the command :

```
# /sbin/blkid
```

From your Linux workstation, you can view the following web page:
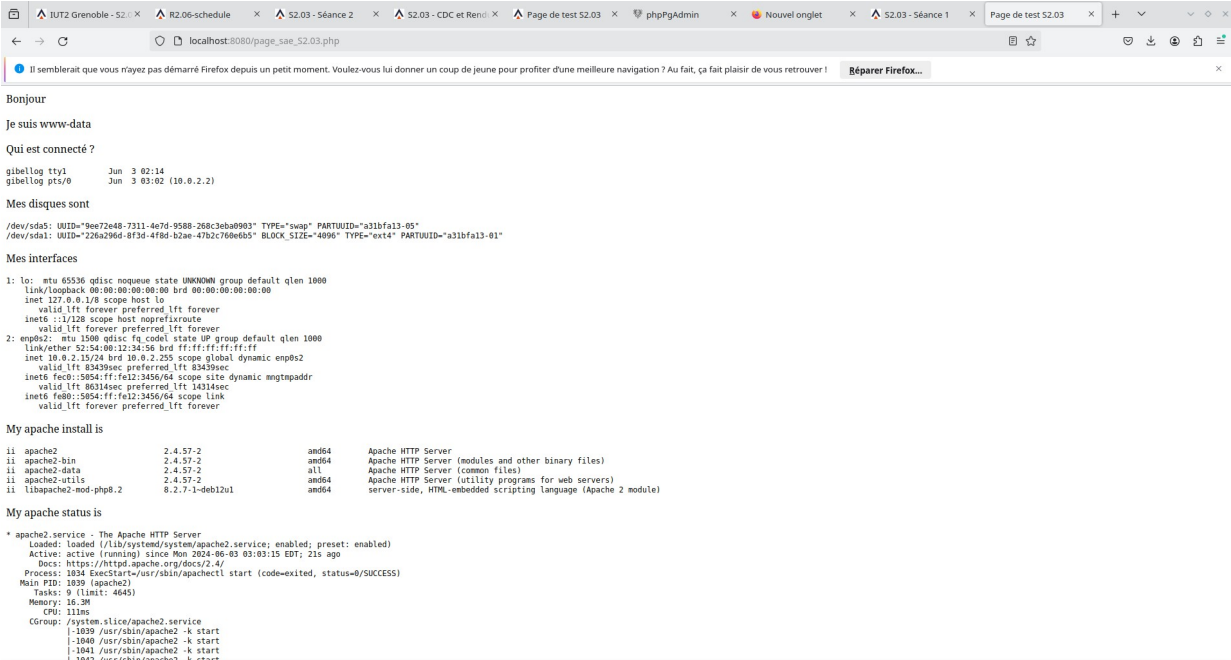
4 - Troubleshooting

IMPORTANT !
In this section, we address most of the likely problems that may arise during the process. If you haven't had any, you can skip it!

x Problem n°1 : My ISO image was not pre-installed correctly.

✓ **Solution:** If no images exist, create a new local disk image. Run the following commands:

```
if [ ! -e "$image_nfs" ] && [ ! -e "$image_locale" ]; then
        echo "Création d'une image disque locale sur cette station Linux ..."
        image="$image_locale"
        qemu-img create "$image" 4G
        sync
        echo "Fini."
fi
```

x Problem n°2 : One of my post-installation verification does not give the expected result.

✓ **Solution:** In this case, the VM has been incorrectly configured at installation. The best thing to do is to start the installation all over again.

x Problem n°3 : The /http://localhost:8080/phppgadmin/ page is not found by my browser.

✓ **Solution:** This problem is probably due to the version of PHPpgAdmin installed on your virtual machine, which doesn't work with PostgreSQL version 15. To solve this issue, enter this command as root :

```
nano /usr/share/phppgadmin/classes/database/Connection.php
```

Find the line that specifies the PostgreSQL version and modify it to include version 15:

```
case '14': return 'Postgres'; break;
```

Change to :

```
case '15': return 'Postgres'; break;
```

Don't forget to save with ctrl+O and ctrl+X to exit.

<u>Sources bibliography :</u>

- https://debian-facile.org/wiki
- <u>https://www.debian.org/releases/stable/installmanual</u>
- <u>https://httpd.apache.org/docs/2.4/en/install.html</u>
- https://www.php.net/manual/en/install.unix.php
- <u>https://www.postgresql.org/</u>

As well as all the BUT informatique courses at Grenoble Alpes University, available online at :
https://chamilo.iut2.univ-grenoble-alpes.fr/

<u>Message from the author :</u>

I hope this detailed installation guide has helped you and that you've enjoyed it, it's been a pleasure for me to write it. I wish you all the best with your virtual machine !

GIBELLO Grégoire