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.

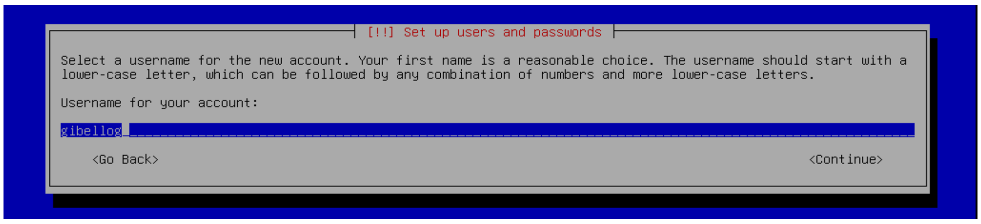
The rest of the settings configure the hardware, networking with port forwarding, and virtual machine display settings.

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 !

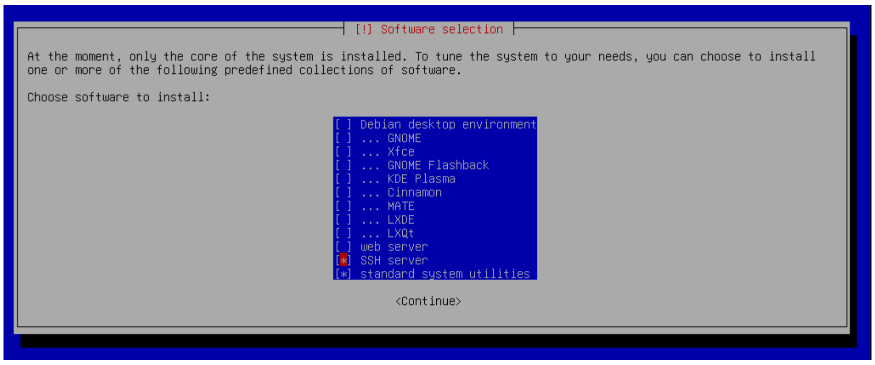
1. **Language** : English
2. **Location** : other/Europe/France
3. **Locales** : United States, en_US.UTF-8
4. **Keyboard** : French
5. **Hostname** : use server-"YOUR_UGA_LOGIN"

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



```
[!!] Set up users and passwords

Select a username for the new account. Your first name is a reasonable choice. The username should start with a
lower-case letter, which can be followed by any combination of numbers and more lower-case letters.

Username for your account:

gibellog

    <Go Back>                                                                                  <Continue>
```

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



```
[!] Software selection

At the moment, only the core of the system is installed. To tune the system to your needs, you can choose to install
one or more of the following predefined collections of software.

Choose software to install:

                                            [ ] Debian desktop environment
                                            [ ] ... GNOME
                                            [ ] ... Xfce
                                            [ ] ... GNOME Flashback
                                            [ ] ... KDE Plasma
                                            [ ] ... Cinnamon
                                            [ ] ... MATE
                                            [ ] ... LXDE
                                            [ ] ... LXQt
                                            [ ] web server
                                            [*] SSH server
                                            [*] standard system utilities

                                                      <Continue>
```

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

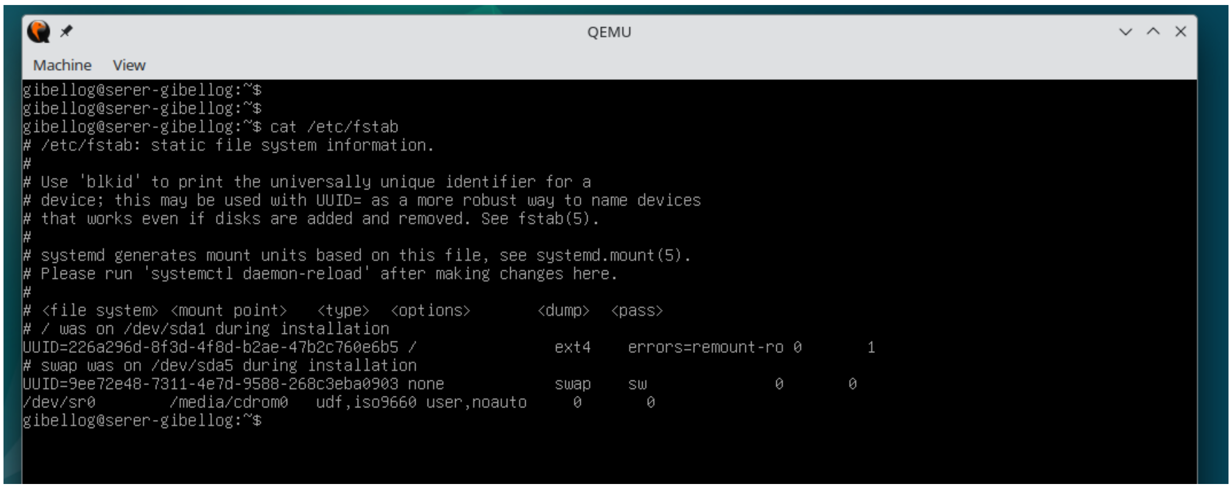
Before continuing, let's do a few checks to make sure that the installation of Debian 12 has been well doned :

2.1 - 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 :



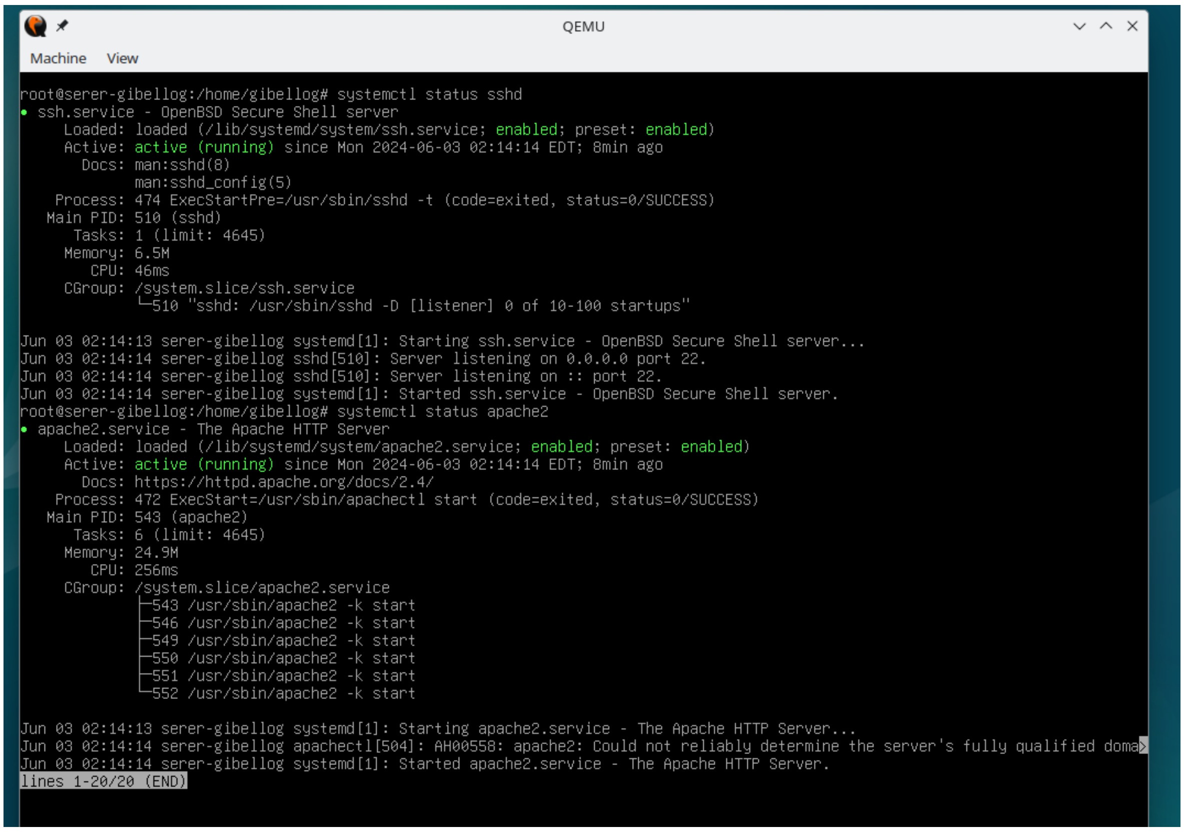
```
gibellog@serer-gibellog:~$
gibellog@serer-gibellog:~$
gibellog@serer-gibellog:~$ cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=226a296d-8f3d-4f8d-b2ae-47b2c760e6b5 /               ext4    errors=remount-ro 0       1
# swap was on /dev/sda5 during installation
UUID=9ee72e48-7311-4e7d-9588-268c3eba0903 none            swap    sw              0       0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
gibellog@serer-gibellog:~$
```

2.2 - Check the status of essential services

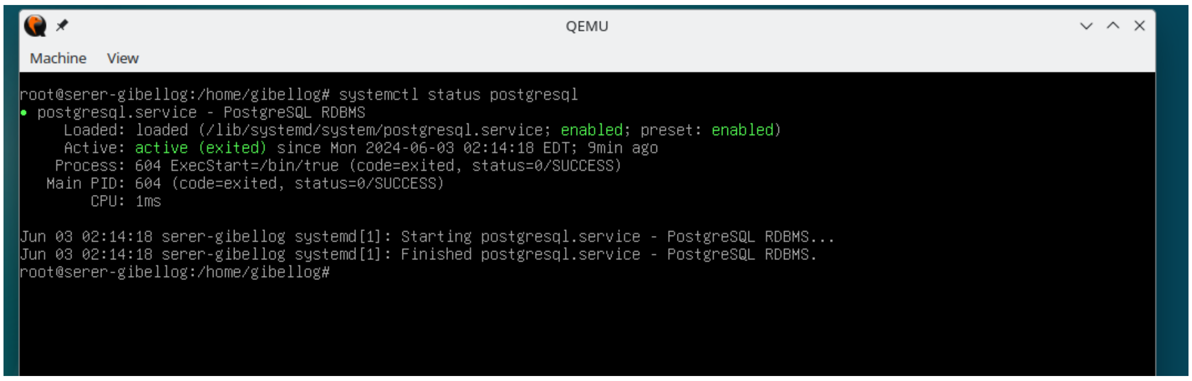
Next, check that the SSH, Apache and PostgreSQL services are running correctly. Use the following command and adapt them to check the status of each service :

```
systemctl status [ssh or apache2 or postgresql]
```

If all went well, you should get this:



```
root@serer-gibellog:/home/gibellog# systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Mon 2024-06-03 02:14:14 EDT; 8min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 474 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 510 (sshd)
      Tasks: 1 (limit: 4645)
     Memory: 6.5M
        CPU: 46ms
     CGroup: /system.slice/ssh.service
             └─510 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jun 03 02:14:13 serer-gibellog systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Jun 03 02:14:14 serer-gibellog sshd[510]: Server listening on 0.0.0.0 port 22.
Jun 03 02:14:14 serer-gibellog sshd[510]: Server listening on :: port 22.
Jun 03 02:14:14 serer-gibellog systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
root@serer-gibellog:/home/gibellog# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Mon 2024-06-03 02:14:14 EDT; 8min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 472 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 543 (apache2)
      Tasks: 6 (limit: 4645)
     Memory: 24.9M
        CPU: 256ms
     CGroup: /system.slice/apache2.service
             ├─543 /usr/sbin/apache2 -k start
             ├─546 /usr/sbin/apache2 -k start
             ├─549 /usr/sbin/apache2 -k start
             ├─550 /usr/sbin/apache2 -k start
             ├─551 /usr/sbin/apache2 -k start
             └─552 /usr/sbin/apache2 -k start

Jun 03 02:14:13 serer-gibellog systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jun 03 02:14:14 serer-gibellog apachectl[504]: AH00558: apache2: Could not reliably determine the server's fully qualified doma>
Jun 03 02:14:14 serer-gibellog systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-20/20 (END)
```



```
root@serer-gibellog:/home/gibellog# systemctl status postgresql
• postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; preset: enabled)
     Active: active (exited) since Mon 2024-06-03 02:14:18 EDT; 9min ago
    Process: 604 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 604 (code=exited, status=0/SUCCESS)
        CPU: 1ms

Jun 03 02:14:18 serer-gibellog systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
Jun 03 02:14:18 serer-gibellog systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
root@serer-gibellog:/home/gibellog#
```

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.

2.3 - 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
```

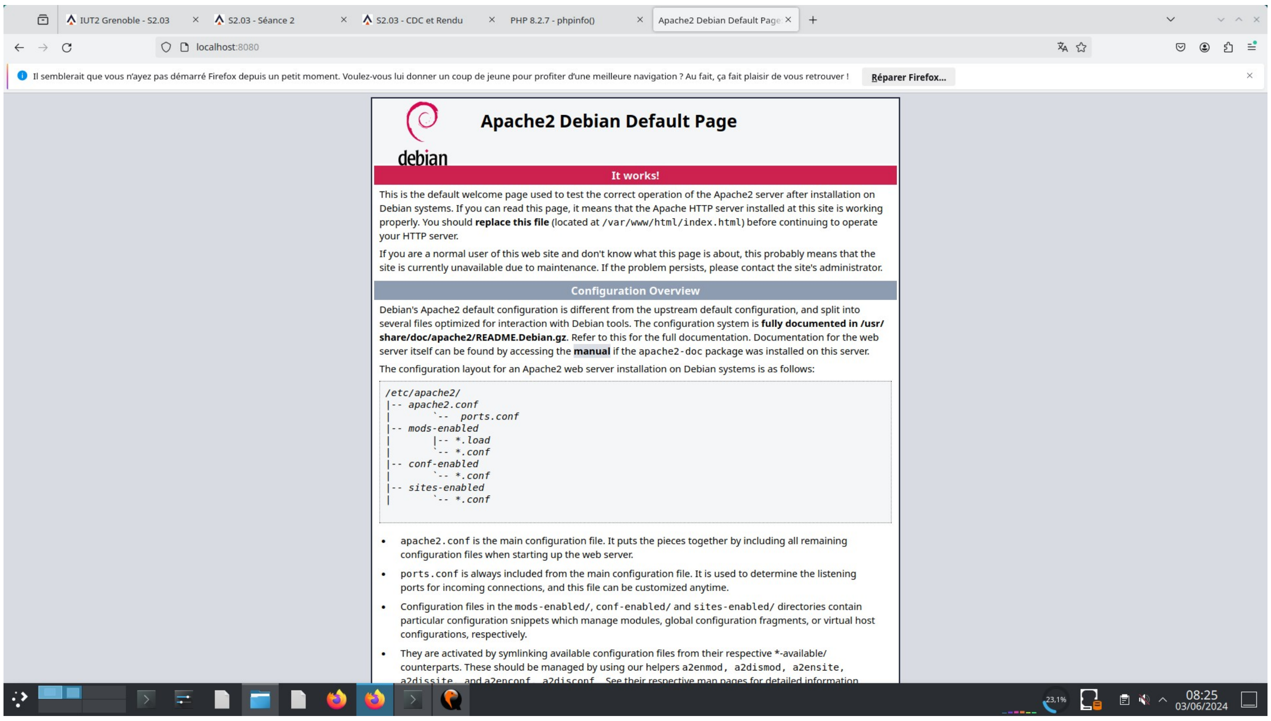
If it is not, enter:

```
# 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.



Il semblerait que vous n'ayez pas démarré Firefox depuis un petit moment. Voulez-vous lui donner un coup de jeune pour profiter d'une meilleure navigation ? Au fait, ça fait plaisir de vous retrouver !

Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information

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
```

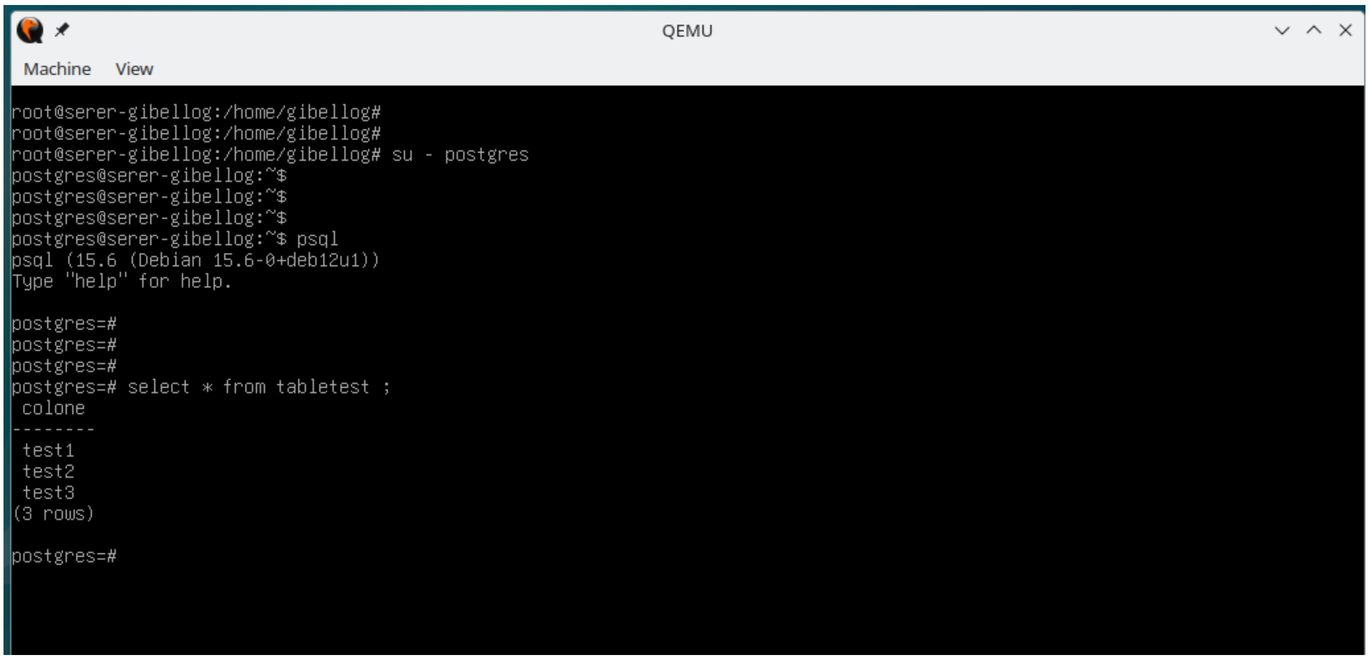
Now, lets create a test table, enter this command and create a psql user with your login:

```
psql
```

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

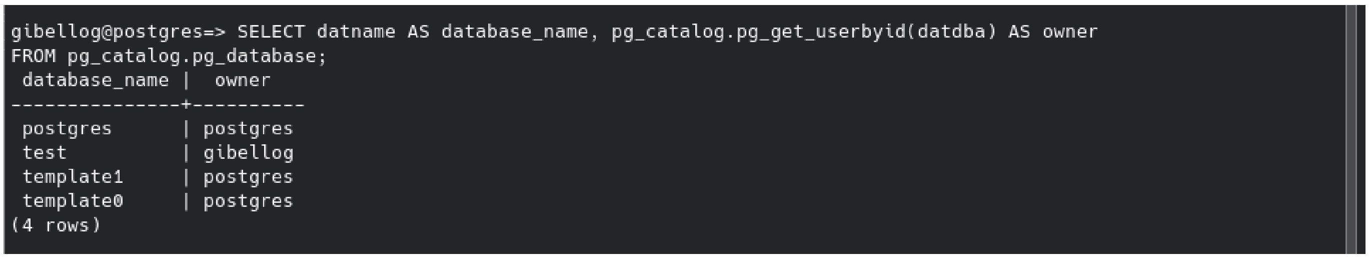
```
CREATE DATABASE [Database name] OWNER [User login] ;
```

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 :

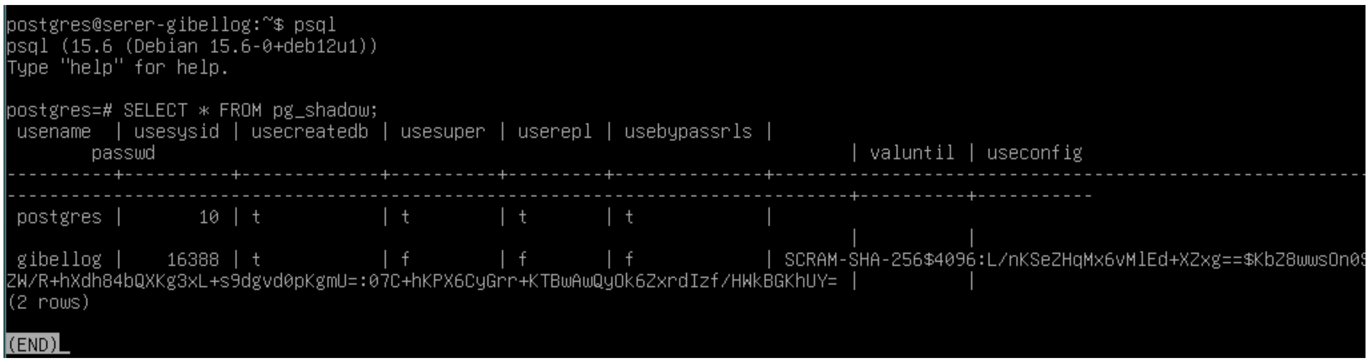


```
gibellog@postgres=> SELECT datname AS database_name, pg_catalog.pg_get_userbyid(datdba) AS owner
FROM pg_catalog.pg_database;
 database_name |  owner
---------------+----------
 postgres      | postgres
 test          | gibellog
 template1     | postgres
 template0     | postgres
(4 rows)
```

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$4096:L/nKSeZHqMx6vMlEd+XZxg==$KbZ8wwsOn0
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 !

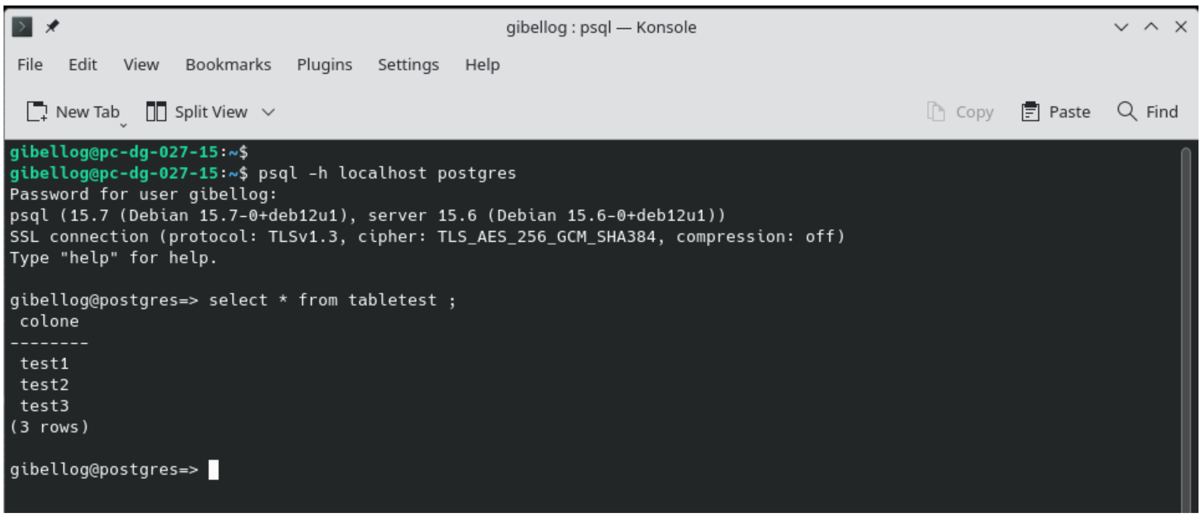
Connect from a terminal on your station, again using psql :

```
psql -h localhost postgres
```

Try to ask your database’s table with :

```
SELECT * FROM [table name] ;
```

You should see this :



```
gibellog@pc-dg-027-15:~$
gibellog@pc-dg-027-15:~$ psql -h localhost postgres
Password for user gibellog:
psql (15.7 (Debian 15.7-0+deb12u1), server 15.6 (Debian 15.6-0+deb12u1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

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

gibellog@postgres=> 
```

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:

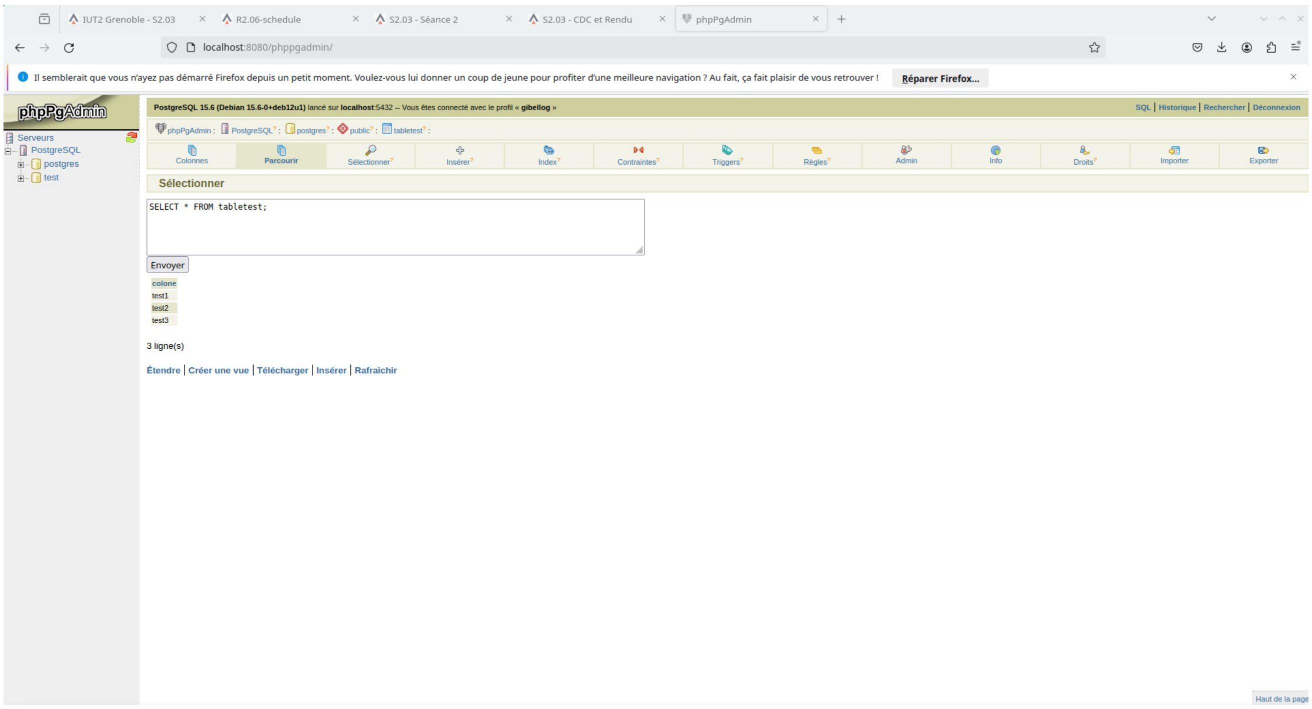
```
Alias /phppgadmin /usr/share/phppgadmin

<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:



Il semblerait que vous n'ayez pas démarré Firefox depuis un petit moment. Voulez-vous lui donner un coup de jeune pour profiter d'une meilleure navigation ? Au fait, ça fait plaisir de vous retrouver !

PostgreSQL 15.6 (Debian 15.6-0+deb12u1) lancé sur localhost:5432 -- Vous êtes connecté avec le profil « gibellog »

```
SELECT * FROM tabletest;
```

colone
test1
test2
test3

3 ligne(s)

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
```

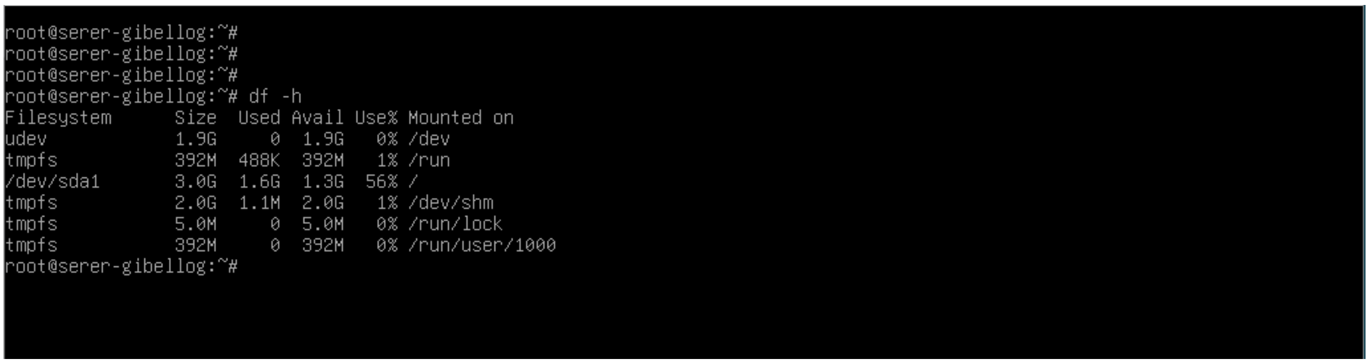
And don't forget to clean up any unnecessary files with :

```
# 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 :



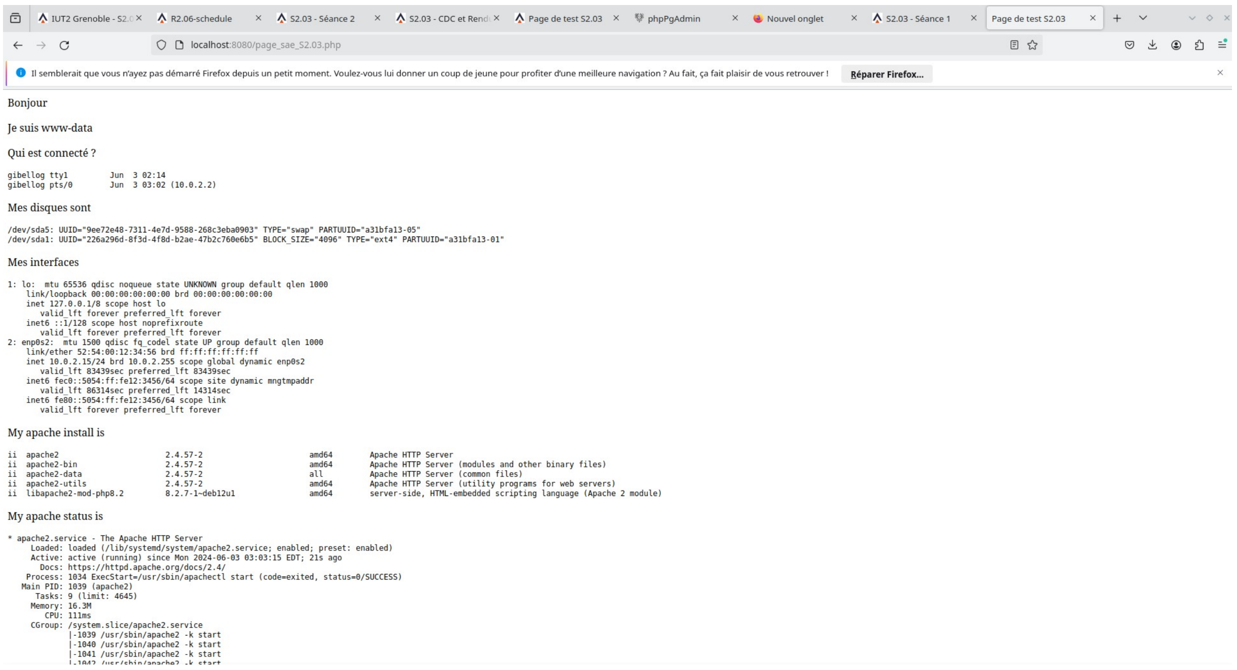
```
root@serer-gibellog:~#
root@serer-gibellog:~#
root@serer-gibellog:~#
root@serer-gibellog:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           392M  488K  392M   1% /run
/dev/sda1       3.0G  1.6G  1.3G  56% /
tmpfs           2.0G  1.1M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



Il semblerait que vous n'ayez pas démarré Firefox depuis un petit moment. Voulez-vous lui donner un coup de jeune pour profiter d'une meilleure navigation ? Au fait, ça fait plaisir de vous retrouver !

Bonjour

Je suis www-data

Qui est connecté ?

```
gibellog tty1        Jun  3 02:14
gibellog pts/0       Jun  3 03:02 (10.0.2.2)
```

Mes disques sont

```
/dev/sda5: UUID="9ee72e48-7311-4e7d-9588-268c3eba0903" TYPE="swap" PARTUUID="a31bfa13-05"
/dev/sda1: UUID="226a296d-8f3d-4f8d-b2ae-47b2c760e6b5" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="a31bfa13-01"
```

Mes interfaces

```
1: lo:  mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s2:  mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:12:34:56 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s2
       valid_lft 83439sec preferred_lft 83439sec
    inet6 fec0::5054:ff:fe12:3456/64 scope site dynamic mngtmpaddr
       valid_lft 86314sec preferred_lft 14314sec
    inet6 fe80::5054:ff:fe12:3456/64 scope link
       valid_lft forever preferred_lft forever
```

My apache install is

```
ii  apache2                  2.4.57-2                     amd64        Apache HTTP Server
ii  apache2-bin              2.4.57-2                     amd64        Apache HTTP Server (modules and other binary files)
ii  apache2-data             2.4.57-2                     all          Apache HTTP Server (common files)
ii  apache2-utils            2.4.57-2                     amd64        Apache HTTP Server (utility programs for web servers)
ii  libapache2-mod-php8.2    8.2.7-1~deb12u1              amd64        server-side, HTML-embedded scripting language (Apache 2 module)
```

My apache status is

```
* apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Mon 2024-06-03 03:03:15 EDT; 21s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 1034 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 1039 (apache2)
      Tasks: 9 (limit: 4645)
     Memory: 16.3M
        CPU: 111ms
     CGroup: /system.slice/apache2.service
             |-1039 /usr/sbin/apache2 -k start
             |-1040 /usr/sbin/apache2 -k start
             |-1041 /usr/sbin/apache2 -k start
             |-1042 /usr/sbin/apache2 -k start
```

4 - Troubleshooting

4.1 - 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.

4.2 - Conclusion and Bibliography

Sources bibliography :

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

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

Message from the author :

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