Installation guide of Debian 12 server equipped with Apache, PostgreSQL and PHP on a virtual machine

Ilyès Drouiche

June 2024

Table of Contents

Part 1: installation of Debian 12 and tools on your machine.

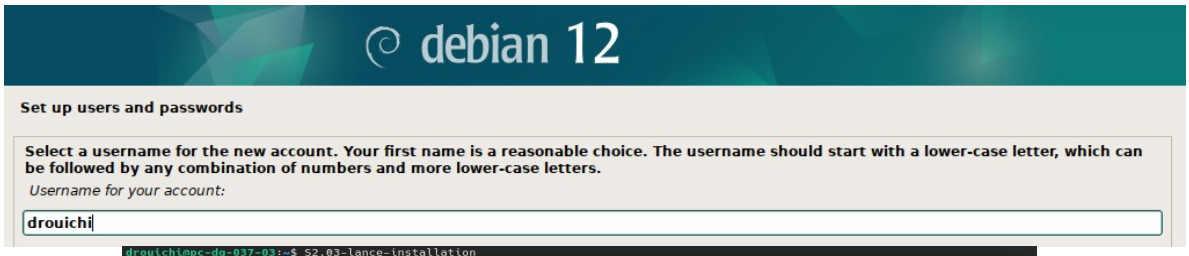
The main goal is to install the following system:

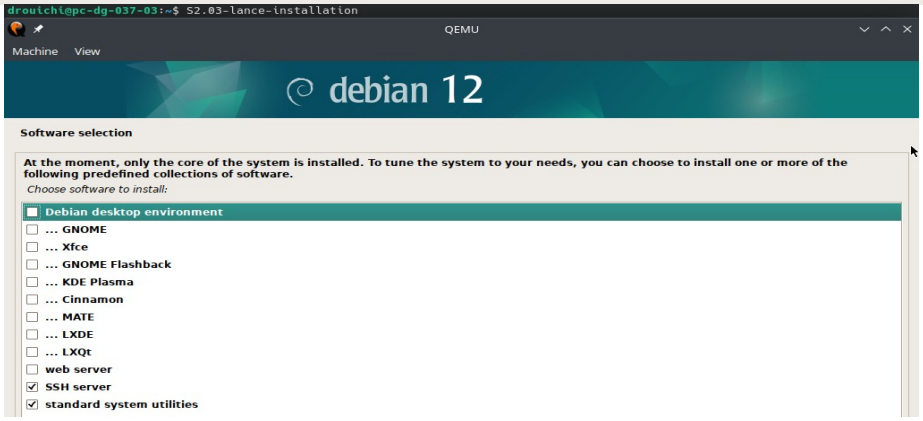
- Debian 12.x
- for x86 CPU 64 bytes
- with ISO image of type “netinst”

Step 1: install the Debian system without graphic interface.

Follow the parameters below to achieve the installation:

Language	English
Location	other/Europe/France
Locales	United_States, en_US.UTF8
Keyboard	French
Hostname	server-"YOUR NAME"
Root Password	A simple password
User Account – Full Name	Your full name
Username	Your name
User Password	A simple password
Partition disks	Guided – use entire disk
Partition disks	All files in one partition
Partition disks	Yes
Software Selection	The “Debian desktop” checkbox has to be ckecked and the “ssh server” checkbox has not to be ckecked
Install GRUB	Yes
Device for boot loader	/dev/sda





Go on the root shell with this command and type your root password :

```
drouichi@server-drouichi: $su -
```

Step 2: installation of SSH.

Normally, ssh is installed by default.To make you certain about that, you can type the following command and you must have the same screen as below:

```
root@server-drouichi:~# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Tue 2024-05-28 08:26:27 CEST; 23min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 490 (sshd)
      Tasks: 1 (limit: 4645)
     Memory: 6.7M
        CPU: 27ms
     CGroup: /system.slice/ssh.service
             └─490 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

May 28 08:26:27 server-drouichi systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
May 28 08:26:27 server-drouichi sshd[490]: Server listening on 0.0.0.0 port 22.
May 28 08:26:27 server-drouichi sshd[490]: Server listening on :: port 22.
May 28 08:26:27 server-drouichi systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
root@server-drouichi:~# _
```

Step 3: installation of Apache.

Install Apache with the following command:

```
root@server-drouichi: # apt install apache2
```

Verify if Apache is nicely installed and compare your screen with this :

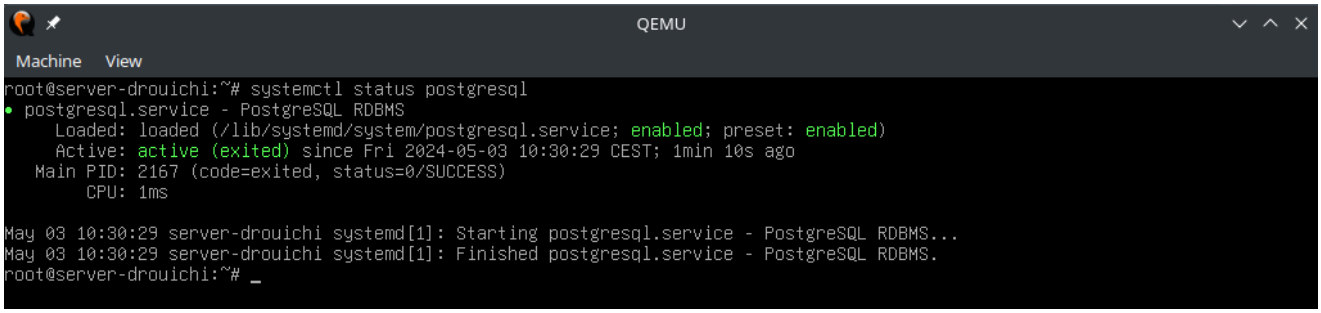
```
root@server-drouichi:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Fri 2024-05-03 10:39:16 CEST; 38s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2852 (apache2)
      Tasks: 55 (limit: 4645)
     Memory: 9.2M
        CPU: 34ms
     CGroup: /system.slice/apache2.service
             ├─2852 /usr/sbin/apache2 -k start
             ├─2854 /usr/sbin/apache2 -k start
             └─2855 /usr/sbin/apache2 -k start

May 03 10:39:16 server-drouichi systemd[1]: Starting apache2.service - The Apache HTTP Server...
May 03 10:39:16 server-drouichi apachectl[2851]: AH00558: apache2: Could not reliably determine the server's fully qualified do>
May 03 10:39:16 server-drouichi systemd[1]: Started apache2.service - The Apache HTTP Server.
root@server-drouichi:~#
```

Step 4: installation of PostgreSQL.

Install PostgreSQL with the following command:

```
root@server-drouichi: # apt install postgresql
```

Verify if postgresql is nicely installed:



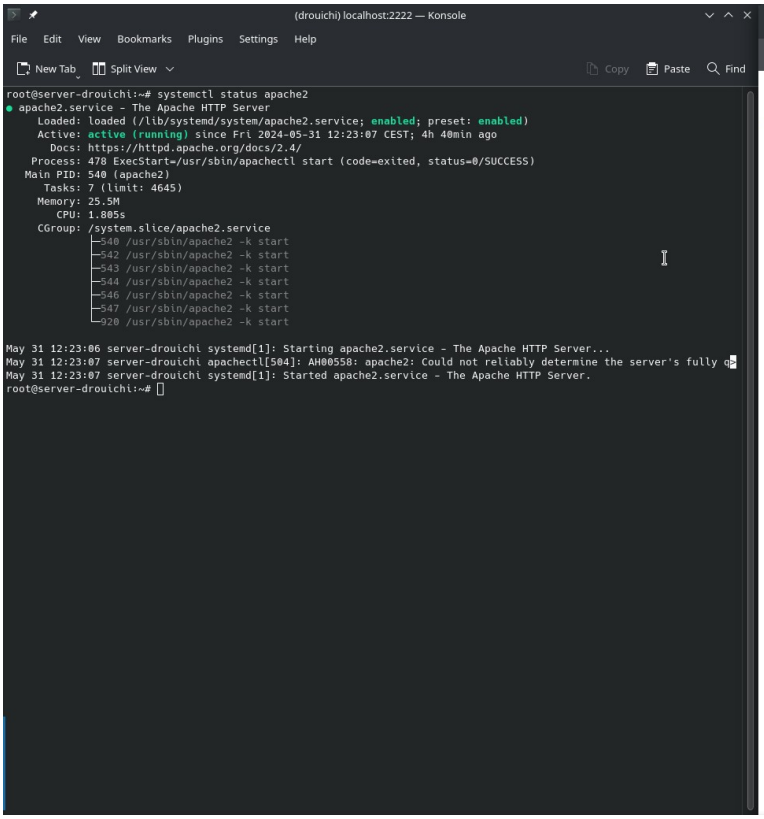
Step 5: the *cat /etc/fstab* command.

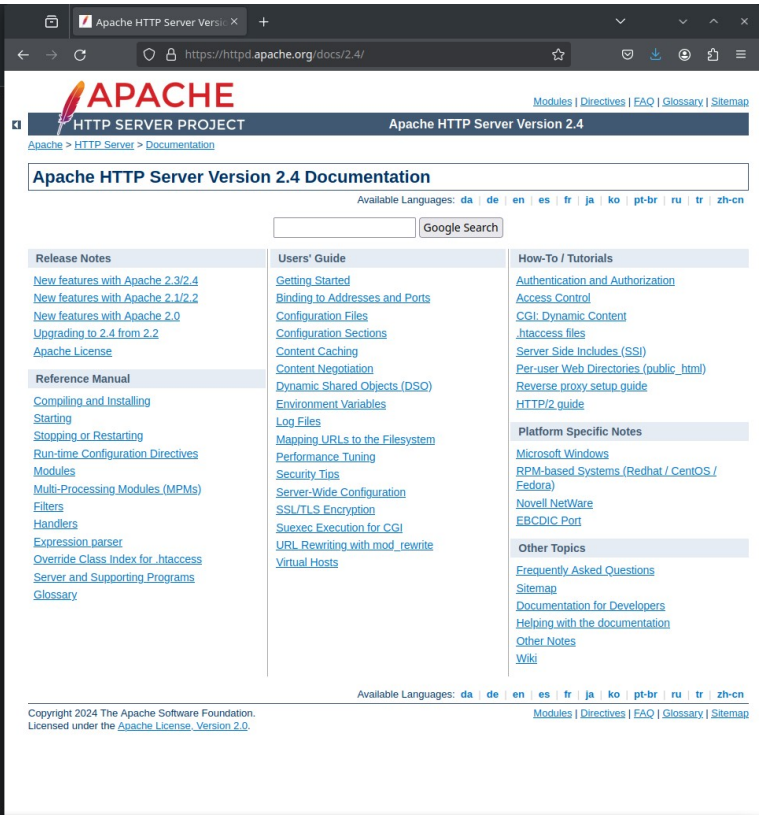
This command brings you the content of the fstab file which contains informations about the file system of your machine.

Part 2: Access to documentation

Step 1: Access to apache server documentation.

On the result of *systemctl status apache2* command, a link will be displayed next to the word “Docs”. This link refers to the documentation of the Apache server you just installed





Step 2: Access to php documentation.

1. With root user, create a file named info.php on the /var/www/html directory.

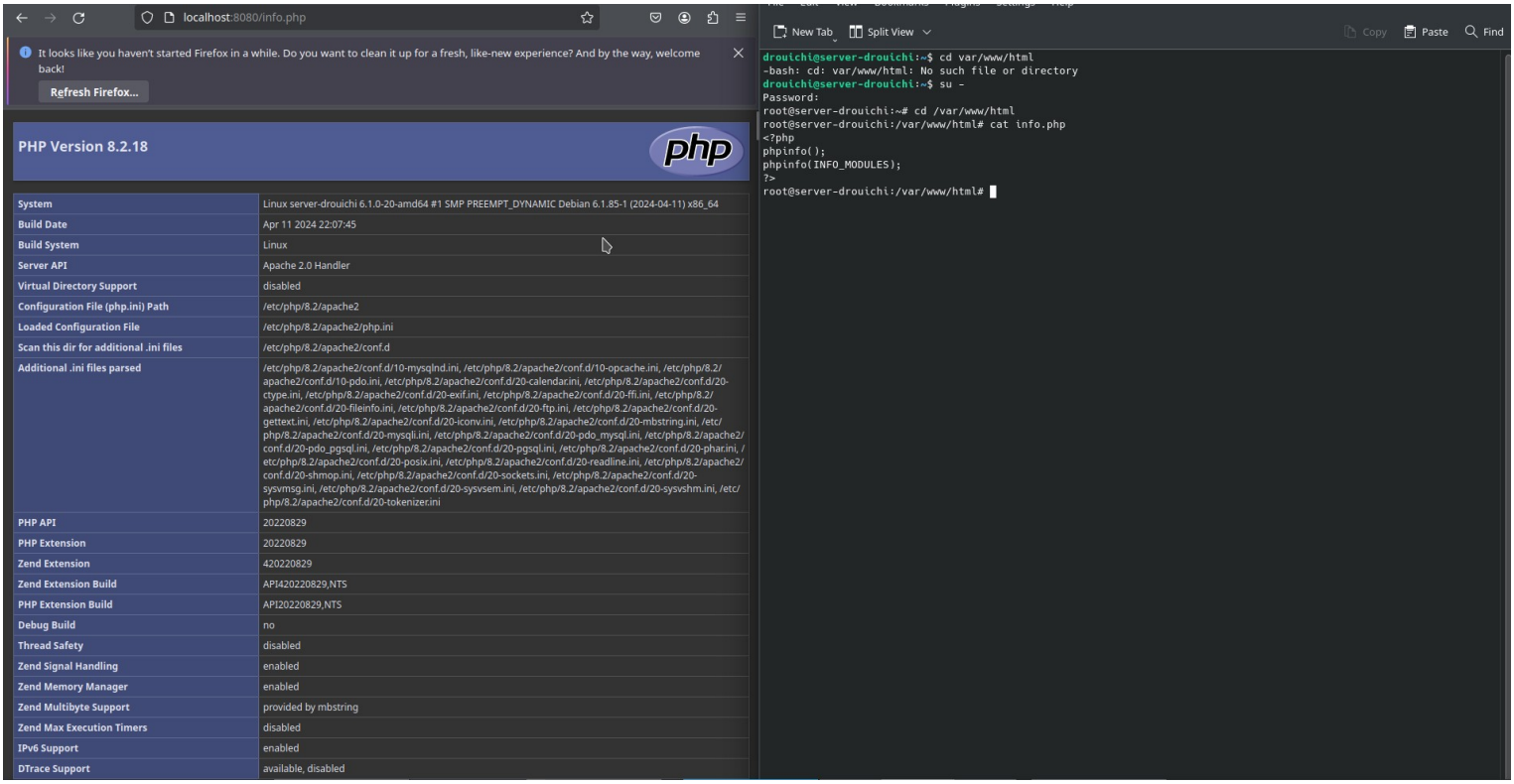
```
root@server-drouichi: # cd /var/www/html
root@server-drouichi:/var/www/html # touch info.php
```

2. Edit this file with the following content:

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

3. Access to the link http://localhost/8080/info.php since the host machine

A page with the main characteristics of your php installation will be displayed


localhost:8080/info.php
It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!
Refresh Firefox...
PHP Version 8.2.18
php
System
Linux server-drouichi 6.1.0-20-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.85-1 (2024-04-11) x86_64
Build Date
Apr 11 2024 22:07:45
Build System
Linux
Server API
Apache 2.0 Handler
Virtual Directory Support
disabled
Configuration File (php.ini) Path
/etc/php/8.2/apache2
Loaded Configuration File
/etc/php/8.2/apache2/php.ini
Scan this dir for additional .ini files
/etc/php/8.2/apache2/conf.d
Additional .ini files parsed
/etc/php/8.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/8.2/apache2/conf.d/10-opcache.ini, /etc/php/8.2/apache2/conf.d/10-pdo.ini, /etc/php/8.2/apache2/conf.d/20-calendar.ini, /etc/php/8.2/apache2/conf.d/20-ctype.ini, /etc/php/8.2/apache2/conf.d/20-exif.ini, /etc/php/8.2/apache2/conf.d/20-ffi.ini, /etc/php/8.2/apache2/conf.d/20-fileinfo.ini, /etc/php/8.2/apache2/conf.d/20-ftp.ini, /etc/php/8.2/apache2/conf.d/20-gettext.ini, /etc/php/8.2/apache2/conf.d/20-iconv.ini, /etc/php/8.2/apache2/conf.d/20-mbstring.ini, /etc/php/8.2/apache2/conf.d/20-mysqli.ini, /etc/php/8.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.2/apache2/conf.d/20-pdo_pgsql.ini, /etc/php/8.2/apache2/conf.d/20-pgsql.ini, /etc/php/8.2/apache2/conf.d/20-phar.ini, /etc/php/8.2/apache2/conf.d/20-posix.ini, /etc/php/8.2/apache2/conf.d/20-readline.ini, /etc/php/8.2/apache2/conf.d/20-shmop.ini, /etc/php/8.2/apache2/conf.d/20-sockets.ini, /etc/php/8.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.2/apache2/conf.d/20-sysvsem.ini, /etc/php/8.2/apache2/conf.d/20-sysvshm.ini, /etc/php/8.2/apache2/conf.d/20-tokenizer.ini
PHP API
20220829
PHP Extension
20220829
Zend Extension
420220829
Zend Extension Build
API420220829,NTS
PHP Extension Build
API20220829,NTS
Debug Build
no
Thread Safety
disabled
Zend Signal Handling
enabled
Zend Memory Manager
enabled
Zend Multibyte Support
provided by mbstring
Zend Max Execution Timers
disabled
IPv6 Support
enabled
DTrace Support
available, disabled
New Tab
Split View
Copy
Paste
Find
drouichi@server-drouichi:~$ cd var/www/html
-bash: cd: var/www/html: No such file or directory
drouichi@server-drouichi:~$ su -
Password:
root@server-drouichi:~# cd /var/www/html
root@server-drouichi:/var/www/html# cat info.php
<?php
phpinfo();
phpinfo(INFO_MODULES);
?>
root@server-drouichi:/var/www/html#

Part 3: SQL queries and security of your password.

Step 1: you can make SQL queries from the virtual machine.

1. First, go on the root and connect you with su – postgres.

```
root@server-drouichi: # su – postgres
```

You are now connected on your postgresql server

2. Create a database with the following command:

```
root@server-drouichi: # su – postgres
postgres@server-drouichi: $ create database ma_base;
```

create database ma_base creates an empty database on your postgresql server

3. Connect you on the base you just created:

```
postgres@server-drouichi: $ psql ma_base;
```

You are now connected on the database.

4. Create your first table with the following SQL command on the database you just created:

```
ma_base=# create table ma_table(prenom varchar, nom varchar, age int);
```

In this example, the table allows you to save the name, the full name and the age of someone.

5. Set a value for each field of the table

```
ma_base=# insert into ma_table values (“Drouiche”, “Ilyès”, 20);
```

Because of this SQL command, you will insert into the table you created before the following values at this order. You can so pay attention of the order.

6. Now, you can make SQL query to print the content of your table:

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

ma_base=# select * from ma_table;
  prenom  |  nom  | age
----------+-------+-----
 Drouiche | Ilyès |  20
(1 row)

ma_base=# _
```

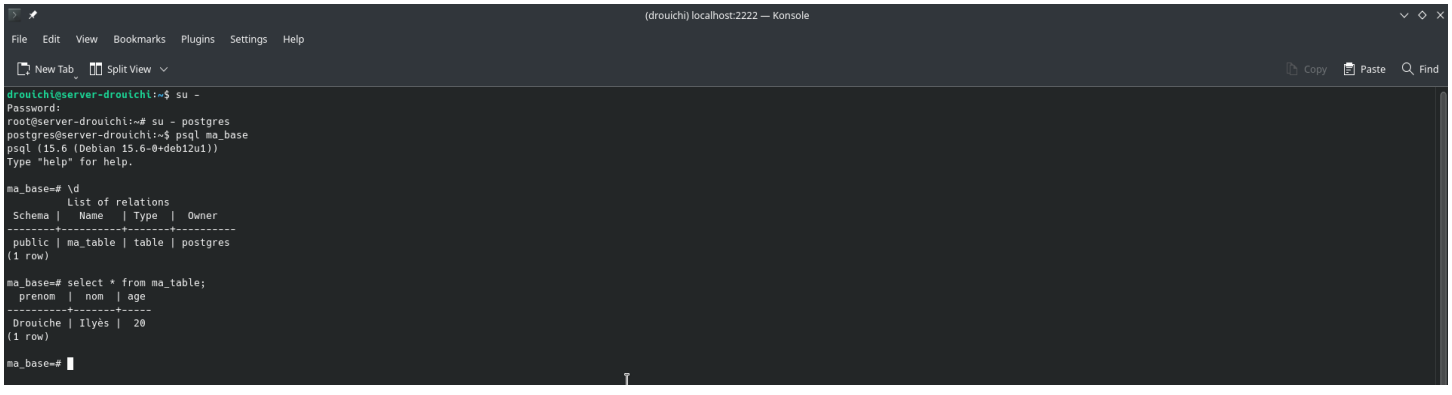
Step 2: try to work on your database in your host machine connected by ssh.

Connect you with ssh from your host machine with the following command:

```
drouichI@pc-dg-037-03:~$ ssh drouichi@localhost -p 2222
```

Replace « drouichi » by your username.

2222 is the port number (-p for port) of the ssh protocol which secures the TCP connection. On the file system of your virtual machine, you can find this: hostfwd=tcp::2222-:22 (S2.03-commun). It shows the port number where the user has to connect.



Step 3: vizualize the list of all your database on your postgresql server.

```
postgres@server-drouichi:~$ psql -l
                                                   List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    | ICU Locale | Locale Provider |   Access privileges
-----------+----------+----------+-------------+-------------+------------+-----------------+-----------------------
 ma_base   | drouichi | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            |
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            | =c/postgres          +
           |          |          |             |             |            |                 | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            | =c/postgres          +
           |          |          |             |             |            |                 | postgres=CTc/postgres
(4 rows)

postgres@server-drouichi:~$ _
```

You can see the database you created before.

Step 4: secure your password with the SHA-256 encryption.

To secure your password, you have to modify two configuration files which are *pg_hba.conf* and *postgresql.conf*

```
postgres@server-drouichi= # nano /etc/postgresql/15/main/postgresql.conf
```

Thanks to this command, you can edit the postgresql.conf file and do the following actions:

1. Research the CONNECTIONS AND AUTHENTICATION section

2. Decomment the listen address line and replace it by

```
listen_adresses = “*’"
```

Now, 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.

3. To do this, edit the authentication rules file:

```
postgres@server-drouichi= # nano /etc/postgresql/15/main/pg_hba.conf
```

4. Add the following rule to only accept authenticated connections by a SHA-256 password.

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

5. Go back to the root and confirm these new configurations by restarting your server.

```
postgres@server-drouichi= #exit
root@server-drouichi: #service postgresql restart
```

6. After the restart of the machine, define a password with this command.

```
postgres=# \password
```

7. Now, you can consult the *pg_shadow* table and see that the password has been hashed by the function SHA-256

```
postgres=# select * from pg_shadow;
 usename  | usesysid | usecreatedb | usesuper | userepl | usebypassrls |                                                                  passwd                                                                   | valuntil | useconfig
----------+----------+-------------+----------+---------+--------------+---------------------------------------------------------------------------------------------------------------------------------------+----------+-----------
 postgres |       10 | t           | t        | t       | t            |                                                                                                                                       |          |
 drouichi |    16388 | f           | f        | f       | f            | SCRAM-SHA-256$4096:WT4q3mOs/foWl52noikCKg==$eX3nqLsKUVJuAQxGHdJcC+DMxraHXnog4BDk93yr44I=:khPDiR/gPLx8TFZbBrHKW+UqGzbAOPXrIiJ6a66HR5k= |          |
(2 rows)
```

Part 4: Using the different tools and finalize

Step 1: making SQL queries with PhpPgAdmin.

1. Edit the *Connection,php* file

```
root@server-drouichi:# nano /usr/share/phppgadmin/classes/database/Connection,php
```

2. On this file, change the line

```
case ‘14’ :  return ‘Postgres’;break;
```

by

```
case ‘15’ :  return ‘Postgres’;break;
```

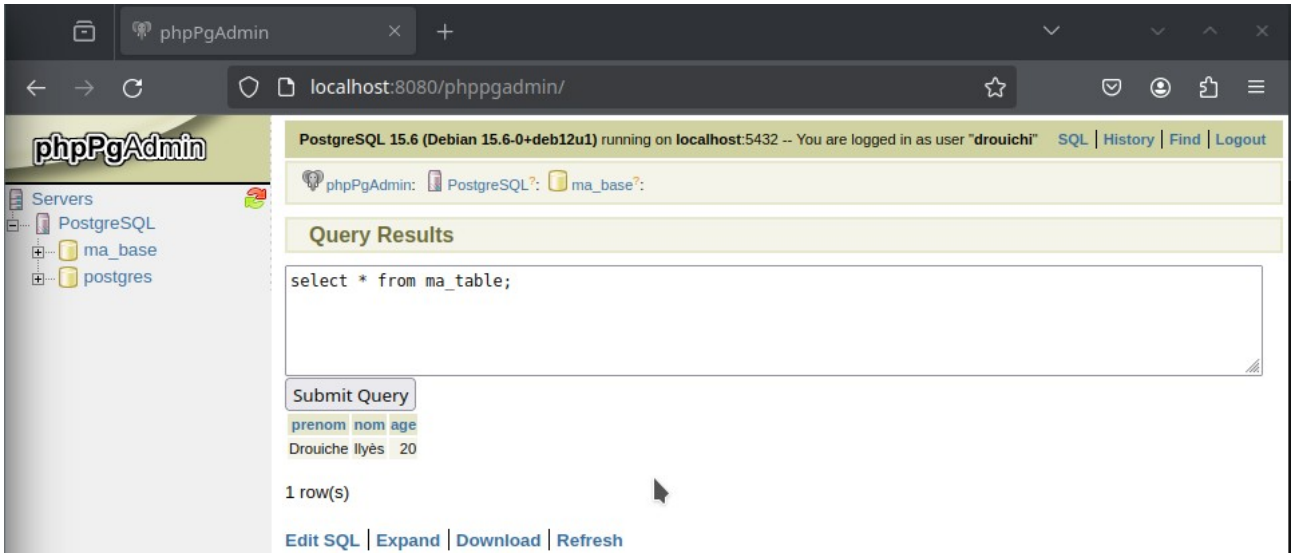
3. Now, you can access PhpPgAdmin with this url on a web browser

```
localhost:8080/phppgadmin/
```

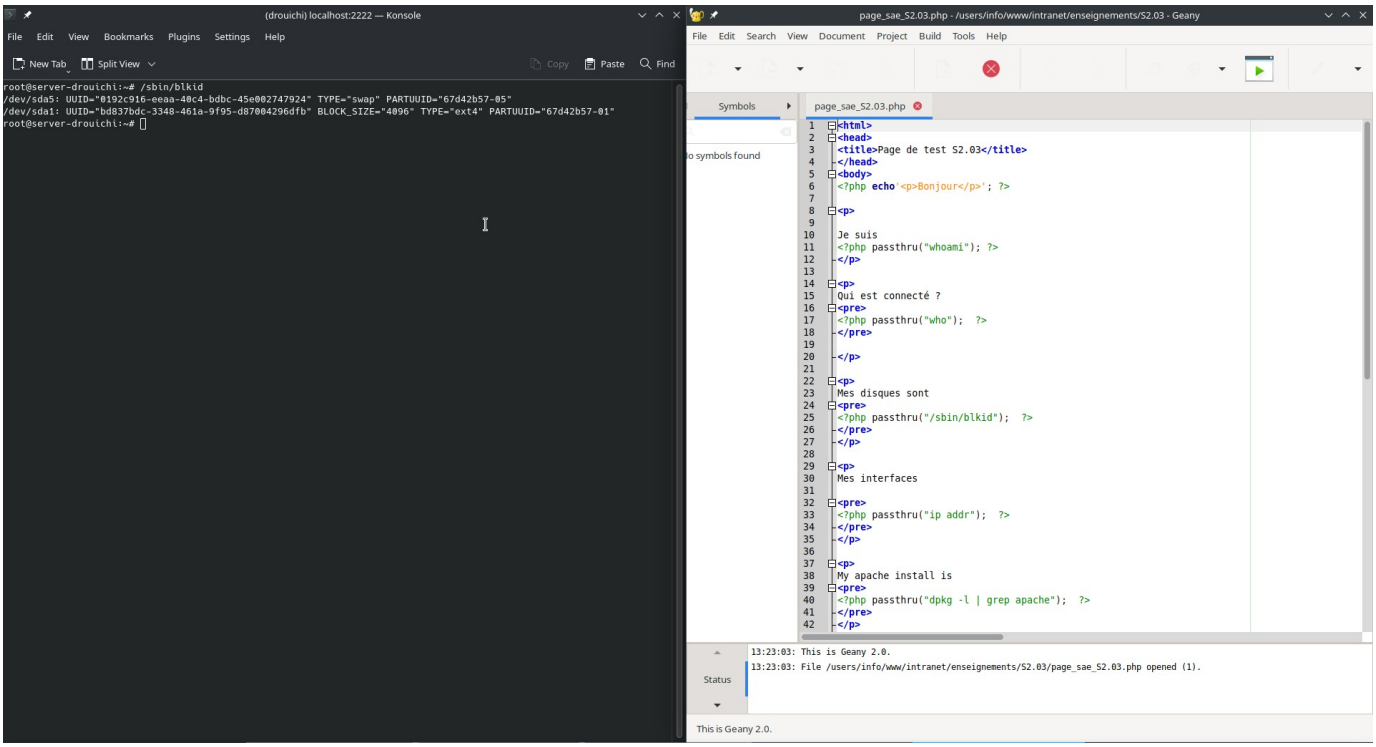
4. Before making a query, you have to grant the select permissions on the table you want to query like that

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

postgres=# \q
postgres@server-drouichi:~$ psql ma_base
psql (15.6 (Debian 15.6-0+deb12u1))
Type "help" for help.

ma_base=# grant select on ma_table to drouichi;
GRANT
```

5. Now, you can make a select query on the PhpPgAdmin page



Step 2: Query a PHP file.



Step 3: display the remaining storage space on the machine after all your installations.

```
drouichi@server-drouichi:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           392M  484K  392M   1% /run
/dev/sda1       3.0G  2.0G  804M  72% /
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
```

Now you have a fully functional Debian 12 system equipped with Apache, PostgreSQL and PHP

Part 5: Appendices

For more information, you can also ckeck out the following links.

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