Installation Guide: Debian Server with Apache, PostgreSQL, and PHP on QEMU/KVM

Author : Colombat Paolo E2

Objective :

This guide will teach you how to install and configure a Debian 12 server without an X server, inside a virtual machine using QEMU/KVM.

The system will include Apache, PostgreSQL, PHP, and phppgadmin all accessible from your host machine.

Table of content :

1. Preparing the Environment

1.1 Root shell

Some commands require being logged in as 'root'.

They will be marked by a #.

To start a root shell use this command :

sudo “name of the user with root privileges”

sudo - “name” if you want the root environment too.

1.2 ISO File

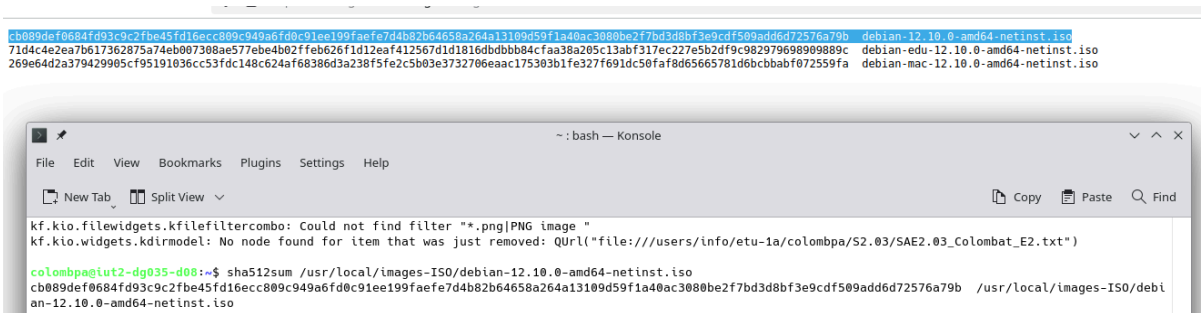
You can download the ISO image at this website :

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

On my pc the Debian 12 ISO is downloaded and available in:

```
colombpa@iut2-dg035-d08:~$ l /usr/local/images-ISO/
total 6178164
-rw-r--r-- 1 root staff         44 11 juil.  2023 CACHEDIR.TAG
-rw-r--r-- 1 root root   663748608 15 mars  13:03 debian-12.10.0-amd64-netinst.iso
-rw-r--r-- 1 root staff 5665497088  9 oct.  16:32 ubuntu-24.10-desktop-amd64.iso
colombpa@iut2-dg035-d08:~$
```

Verify the integrity of the ISO image for security:



We compare the site’s hashes with ours (obtained using the sha512sum command).

If both values match, the file is intact.

2. Installing Debian 12 on QEMU/KVM

2.1 Start the Installer

We start QEMU with this really long command :

```
qemu-system-x86_64 -machine q35 -cpu host -m 4G -enable-kvm
-device VGA,xres=1024,yres=768 -display gtk,zoom-to-fit=off -drive
$drive -device e1000,netdev=net0 -netdev
user,id=net0,hostfwd=tcp::2222-:22,hostfwd=tcp::4443-:443,hostfwd=t
cp::8080-:80,hostfwd=tcp::5432-:5432
```

Here are the parameters used (the most important ones will be in underlined):

qemu-system-x86_64: Starts a 64-bit virtual machine.

-machine q35: Uses the Q35 chipset (more modern than the default).

-cpu host: Emulates the host machine CPU for better performance.

-m 4G: We give 4 GB of RAM to the virtual machine.

-enable-kvm: Enables hardware acceleration (KVM), if available.

-device VGA,xres=1024,yres=768: Sets up a VGA display with a fixed resolution of 1024x768.

-display gtk,zoom-to-fit=off: Uses a GTK window for the VM, without automatic zoom.

-drive $drive: Specifies the disk image to use (stored in the $drive variable).

-device e1000,netdev=net0: Adds an Intel e1000 network card, linked to net0.

-netdev user,id=net0,...: Sets up user-mode networking with port forwarding.

Port forwarding *:*

(allows access to the servers running on the virtual machine from the host machine.)

hostfwd=tcp::8080-:80: HTTP traffic on port 8080.

hostfwd=tcp::4443-:443: HTTPS on 4443.

hostfwd=tcp::5432-:5432: PostgreSQL on 5432.

hostfwd=tcp::2222-:22: SSH on port 2222.

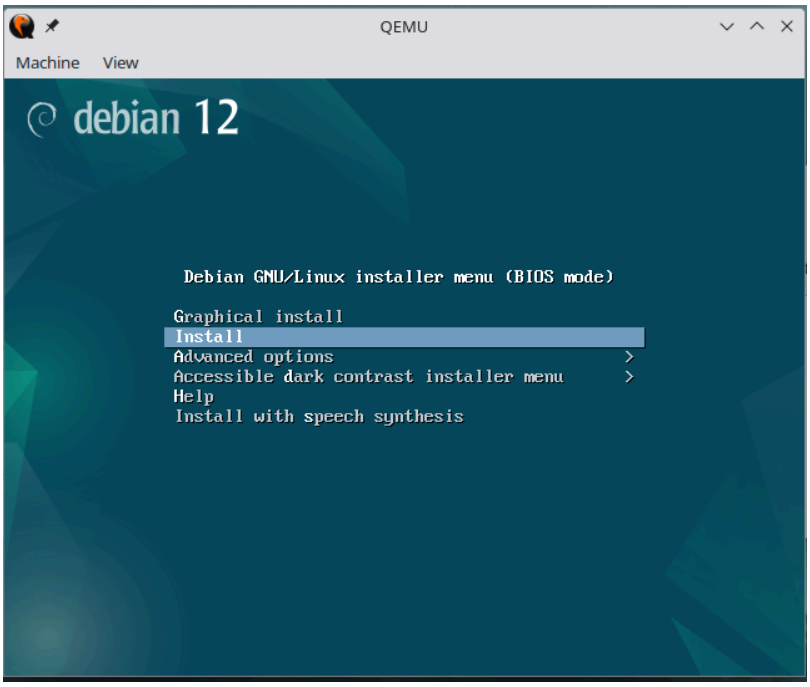
We will be able to connect to the server from our host machine with this command :

ssh user_name@localhost -p 2222

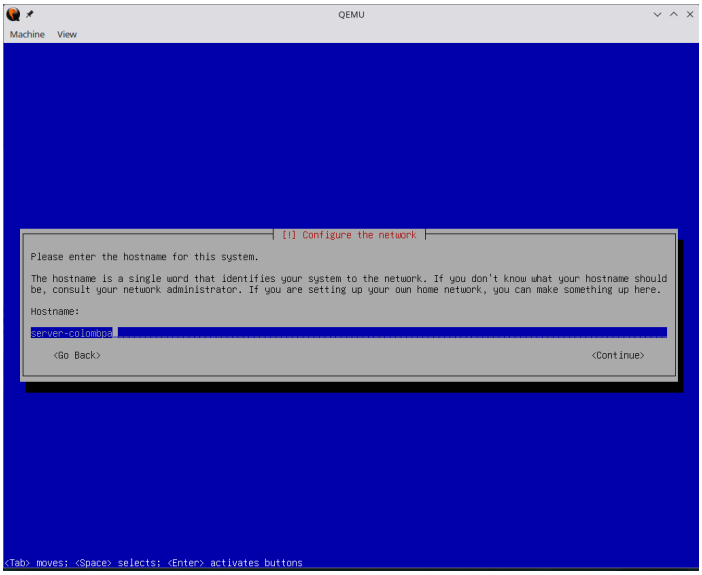
```
colombpa@iut2-dg037-d12:~$ ssh colombpa@localhost -p 2222
colombpa@localhost's password:
Linux server-colombpa 6.1.0-32-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.129-1 (2025-03-06) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Mar 24 13:52:51 2025 from 10.0.2.2
colombpa@server-colombpa:~$ su
```

2.2 Installation

We select “install” and not “graphical install” because we want Debian without Graphical User Interface (GUI).



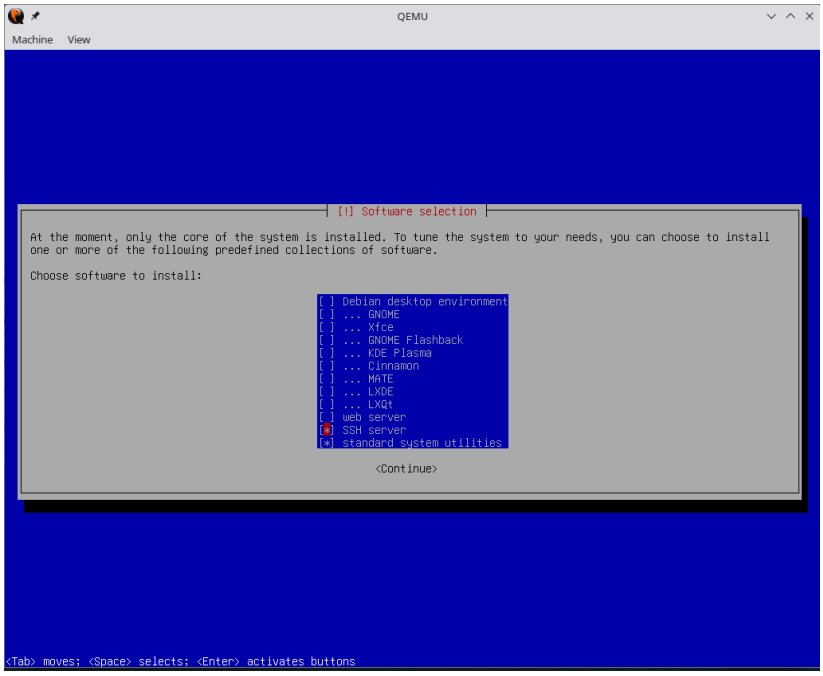
My user will be “serveur-colombpa”.



Use the options you want, but for this guide we use those.

(default when not specified)

- Language: English
- Location: Other > Europe > France
- Locale: en_US.UTF-8
- Keyboard: French
- Partitioning: Guided – Use entire disk – All files in one partition
- Software selection:



After installation if you need to restart the virtual machine, to shut down properly use this command:

 # systemctl poweroff

3. Ethernet and VM check

3.1 Basic check

We can verify Xorg is not installed:

```
colombpa@server-colombpa:/$ dpkg -l | grep xorg
colombpa@server-colombpa:/$
```

Indeed, it is unnecessary since we don't need it for a server.

We can also see our ethernet specifications :

```
colombpa@server-colombpa:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> 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: <BROADCAST,MULTICAST,UP,LOWER_UP> 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 86320sec preferred_lft 86320sec
    inet6 fec0::5054:ff:fe12:3456/64 scope site dynamic mngtmpaddr
       valid_lft 86325sec preferred_lft 14325sec
    inet6 fe80::5054:ff:fe12:3456/64 scope link
       valid_lft forever preferred_lft forever
colombpa@server-colombpa:~$ _
```

and finally check that we can access the external network :

```
colombpa@server-colombpa:/tmp$ ping www.example.org
PING a1519.dscr.akamai.net (2.21.34.104) 56(84) bytes of data.
```

ssh is pre-installed in debian, we can check if it's active

 # systemctl status apache2

```
root@server-colombpa:/home/colombpa# systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; preset: enabled)
     Active: active (exited) since Mon 2025-03-24 14:34:00 CET; 1min 37s ago
   Main PID: 2752 (code=exited, status=0/SUCCESS)
        CPU: 743us

Mar 24 14:34:00 server-colombpa systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
Mar 24 14:34:00 server-colombpa systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
root@server-colombpa:/home/colombpa# _
```

3.2 About the file system

The cat /etc/fstab command allows you to display the content of the /etc/fstab file.

It shows the list of partitions, disks, and other storage devices, along with their mount points and options :

4. Quick apt guide

These command are important to know :

 # apt update

This command updates the list of available packages from the repositories
(It does not update the software itself, only the information about what is available.)

 # apt uprgade

This command installs the new versions of the packages you already have, if they are available (based on the information retrieved by apt update).

 # apt clean

This command deletes temporary downloaded package files (frees up space after an install).

5. Installing Apache

Install with:

 # apt install apache2

Check that apache is running :

 # systemctl status apache2

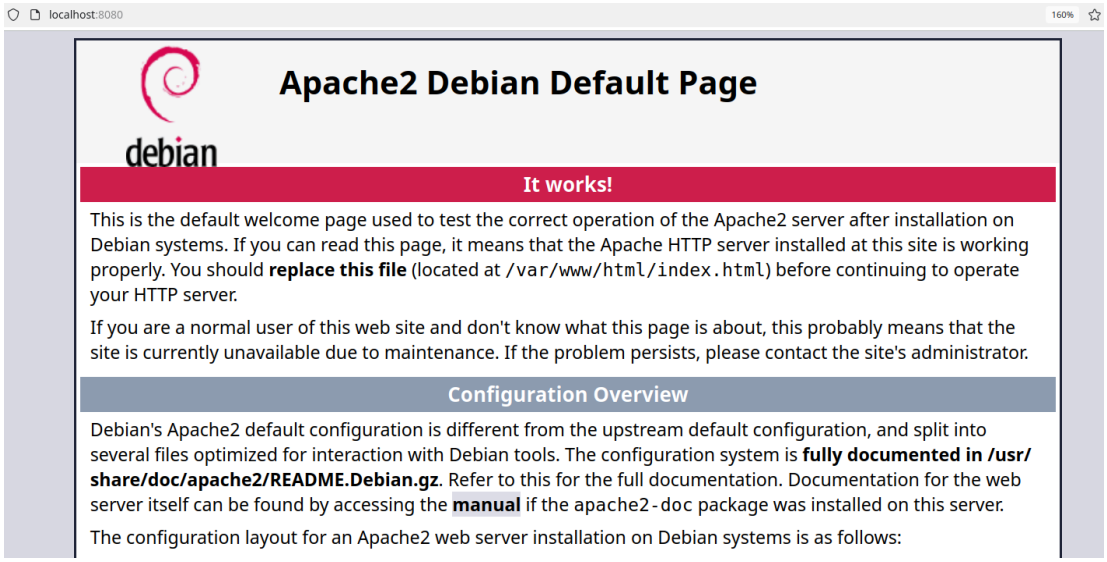
```
root@server-colombpa:/home/colombpa# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-03-24 14:08:09 CET; 3min 50s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 1044 (apache2)
      Tasks: 55 (limit: 4642)
     Memory: 9.2M
        CPU: 44ms
     CGroup: /system.slice/apache2.service
             ├─1044 /usr/sbin/apache2 -k start
             ├─1046 /usr/sbin/apache2 -k start
             └─1047 /usr/sbin/apache2 -k start

Mar 24 14:08:09 server-colombpa systemd[1]: Starting apache2.service - The Apache HTTP Server...
Mar 24 14:08:09 server-colombpa apachectl[1043]: AH00558: apache2: Could not reliably determine the server's fully qualified do>
Mar 24 14:08:09 server-colombpa systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-16/16 (END)
```

If not, start it :

 # systemctl start apache2

We can see that apache is active so it's now accessible from our host machine:

http://localhost:8080 on firefox



6. Installing PostgreSQL

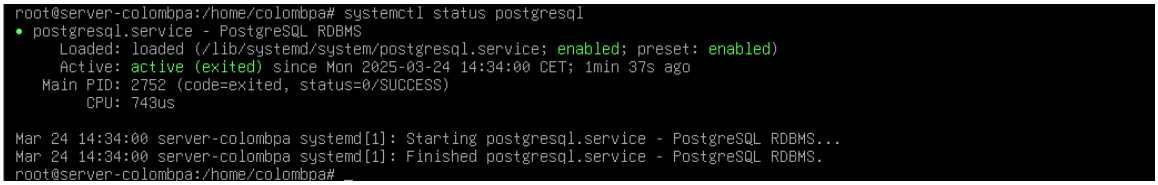
6.1 Installation

Install with:

```
# apt install postgresql
```

Check that postgresql is running :

```
# systemctl status postgresql
```



6.2 Configuration

⚠️ By default, remote (TCP) connections are not allowed.

We need to edit the postgresql.conf file, I suggest you use the pre-installed software nano.

You can find it here : */etc/postgresql/15/main/postgresql.conf*

In the “connections and authentification” section, find the following line, uncomment it, and update it :

```
listen_addresses = '*'
```

The server now listens for connection requests from non-local IP addresses.

You 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
```

Then add the following rule to allow only connections authenticated with a password stored using a Secure Hash Algorithm (*scram-sha-256*) function:

```
host  all  all  0.0.0.0/0  scram-sha-256
```

Don’t forget to restart the postgres server to confirm changes.

 # service postgresql restart

In the system table *pg_shadow*, we can see that my password is well protected by *scram-sha-256*.

```
 usename  | usesysid | usecreatedb | usesuper | userepl | usebypassrls |                                                                passwd
----------+----------+-------------+----------+---------+--------------+---------------------------------------------------------------------------------------------------------------------------------------
 postgres |       10 | t           | t        | t       | t            |
 colombpa |    16388 | f           | f        | f       | f            | SCRAM-SHA-256$4096:0vJB0xPvGNaa7XLVK+4G1g==$3mjCBafDOpj0BM2/6I3cWOHPXHUmBWQwkH3ZgH63cQw=:eHp5P+hSEiAXxU1ZEDyMDYOslaU65+iYF/aZ+PlQlZk=
(2 rows)
```

6.3 Connexion

We switch to postgres user :

su - postgres

This command connects to mybase using the postgres user and the localhost host :

psql -h localhost -U postgres -d mybase

-h: Specifies the host address (server where PostgreSQL is installed) it can be an IP or an hostname.

-U: Specifies the PostgreSQL user to use for the connection.

Now, you're connected with superuser privilege.

We can list our databases:

\l

```
postgres=# \l
                                               List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    | ICU Locale | Locale Provider |   Access privileges
-----------+----------+----------+-------------+-------------+------------+-----------------+-----------------------
 paolobase | colombpa | 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)
```

For this guide, I have created a user “colombpa”, and a base “paolobase” with a table “champions”.

```
root@server-colombpa:~# psql -h localhost -U colombpa paolobase
Password for user colombpa:
psql (15.12 (Debian 15.12-0+deb12u2))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

paolobase=> SELECT * FROM champions;
 id | champion
----+----------
  1 | yasuo
  2 | lee sin
(2 rows)
```

Thanks to the config we did, we can also access to our server with the host machine :

```
colombpa@iut2-dg035-d01:~$ psql -h localhost -U colombpa paolobase
Password for user colombpa:
psql (15.12 (Debian 15.12-0+deb12u2))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

paolobase=> SELECT * FROM champions;
 id | champion
----+----------
  1 | yasuo
  2 | lee sin
(2 rows)
```

If you want to leave postgres :

```
\q
```

7. Installing PHP

Install PHP :

```
# apt install php-common libapache2-mod-php php-cli
```

Optional :

We can create a php page :

```
echo "<?php
phpinfo();
phpinfo(INFO_MODULES);
?>" > /var/www/html/info.php
```

And access from browser on host machine : http://localhost:8080/info.php

8. Installing PhpPgAdmin

8.1 Installation

So that apt can find PhpPgAdmin, we need to edit the source.list file :

```
nano /etc/apt/sources.list
```

Now just add :

"deb http://deb.debian.org/debian/ bookworm-backports contrib main non-free non-free-firmware"

Don't forget to use *# apt update* If you want to confirm changes.

Then, you can install the package:

```
apt install phppgadmin
```

8.2 Configuration

Find Connection.php file:

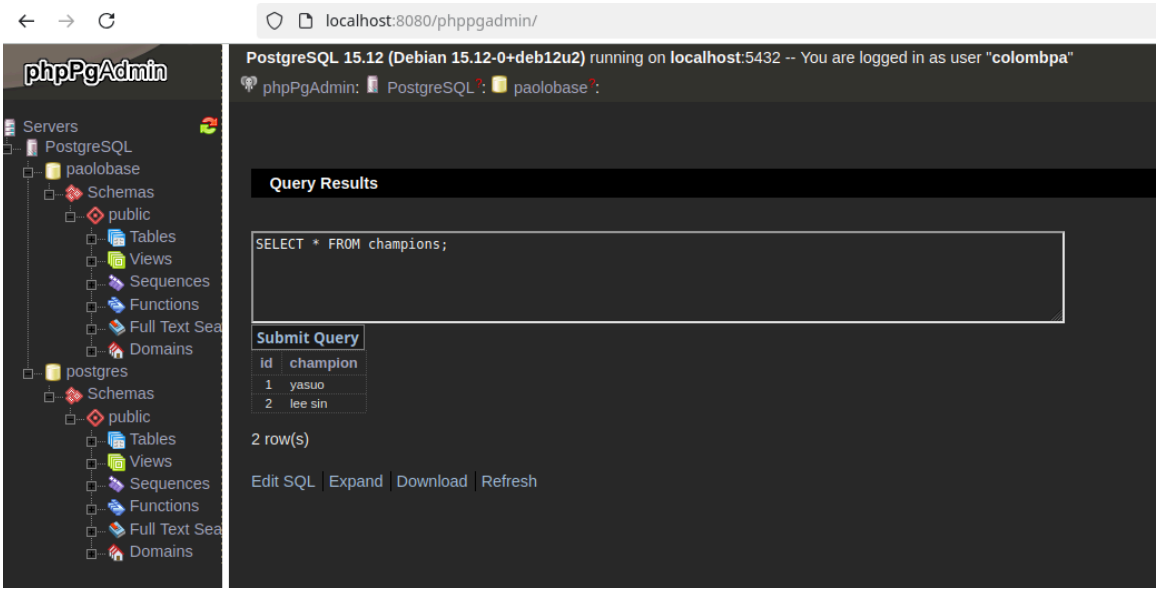
```
find / -name "Connection.php"
```

Then edit it.

```
Replace:
case '14': return 'Postgres';
with:
case '15': return 'Postgres';
```

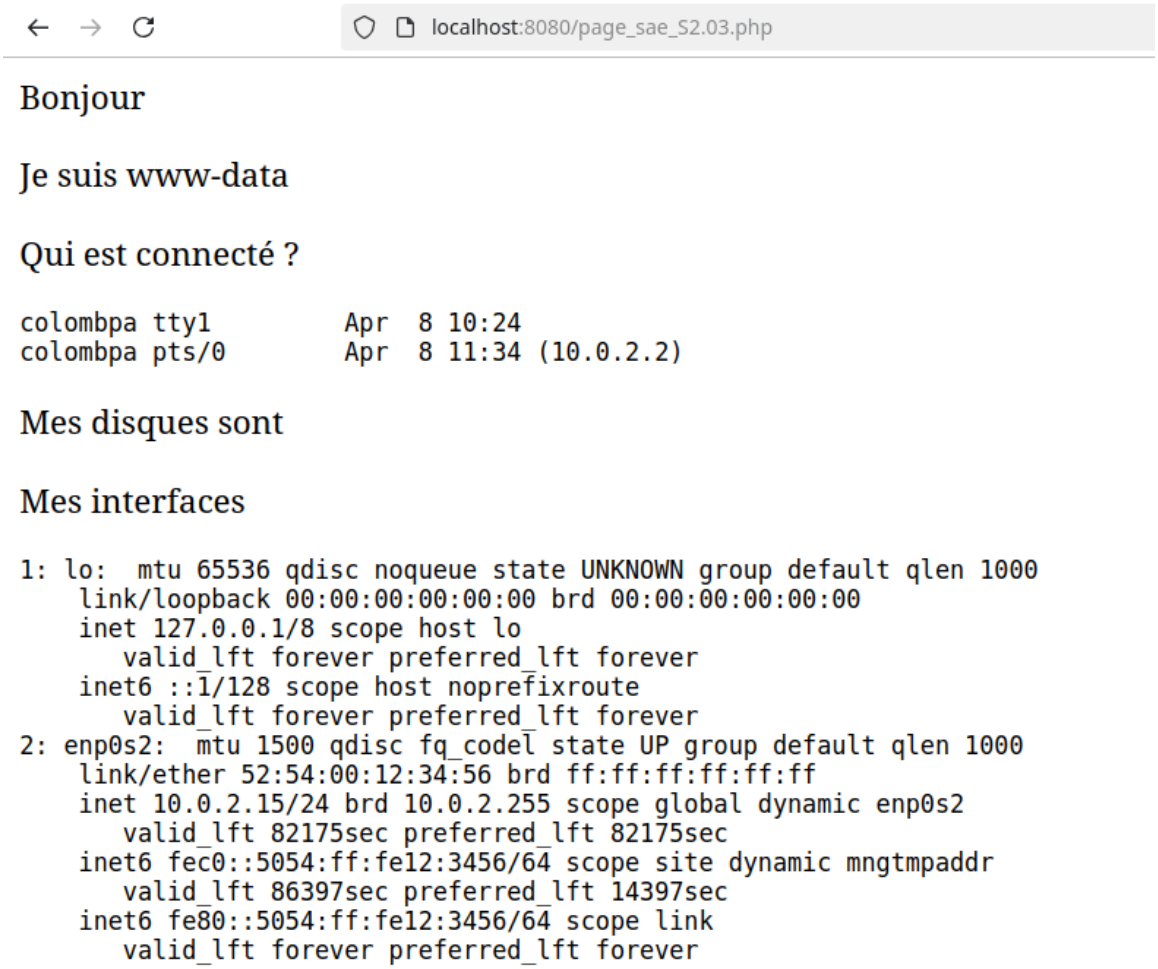
8.3 Connection

I can now access the PhpPgAdmin web interface and use it to view and edit my table on my host machine :



8. Final informations

If you want to see information about my virtual machine on a web php page:



About the disk usage :

After all these installations, we can check the disk usage with this command :

df -h

```
root@server-colombpa:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           392M  476K  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
```

Thank you for reading my guide 👍