

# Debian 12



**Debian 12 Server  
Installation Guide  
with Apache2,  
PostgreSQL and  
PhpPgAdmin**

## Table of contents

Illustrations table .....	3
1. Introduction .....	4
2. Preparing the installation .....	4
1.1 Preparing the ISO image : .....	4
3. Installing Debian 12.....	4
1.2 Launch installation:.....	4
1.3 Debian configuration : .....	5
1.4 Finalizing the Installation: .....	6
1.5 Installation Verification .....	6
1.6 Connecting via SSH: .....	7
1.7 Checking for the Absence of Xorg Server:.....	8
4. Module installation .....	9
1.8 Apache2 installation: .....	9
1.9 PostgreSQL installation : .....	10
1.9.1 Verify the installation:.....	10
1.9.2 Configuration : .....	11
1.9.3 Useful Changes: .....	12
1.10 PhpPgAdmin installation .....	15
1.10.1 Let's start by installing PHP: .....	15
1.10.2 PhpPgAdmin installation:.....	15
1.10.3 Test your database from PhpPgAdmin: .....	16
1.10.4 Test PHP files:.....	17
5. Security and Finalization: .....	18
1.11 Finalization:.....	18
1.12 Security Analysis:.....	21
6. Conclusion : .....	22
7. Controls and documentations: .....	22
1.13 Virtual machine management :.....	22
1.14 Web server management: .....	22
1.15 Database management:.....	23
1.16 File management: .....	24
1.17 System management: .....	24

1.18	Documentations:	24
------	-----------------	----

## Illustrations table

1-ISO-image-page	4
2-SHA512SUMS	4
3-Setup-user	5
4-Software-selection	6
5-IP-and-Ethernet-characteristics	7
6-/cat/fstab	7
7-Virtual-Machine-root	8
8-Virtual-Machine-user	8
9-Physical-Machine-root	8
10-Physical-machine-user	8
11-find-Xorg	8
12-Apache2-status	9
13-Apache2-Default-Page	10
14-Status-postgresql	11
15-Postgres-Database	12
16-Postgresql-pg_shadow	14
17-Postgresql-test-mv	14
18-Postgresql-test-pm	14
19-PhpPgAdmin-home	16
20-PhpPgAdmin-database	16
21-PhpPgAdmin-test	17
22-PhpPgAdmin-pagetest	18
23-rufus-installation	19
24-Status-Module	20
25-space-after-installation	21
26-Update-Upgrade	21

# 1. Introduction

This guide contains the steps required to install a non-graphical Debian 12 server on a virtual machine using Qemu/KVM, as well as the installation of the Apache2, PostgreSQL and PhpPgAdmin modules. The guide is designed so that anyone can install their own virtual machine. I've gone into detail so that even someone with basic computer knowledge can succeed.

## 2. Preparing the installation

### 1.1 Preparing the ISO image :

Download the virtual image called « [debian-12.5.0-amd64-netinst.iso](https://cdimage.debian.org/cdimage/release/current/amd64/iso-cd/debian-12.5.0-amd64-netinst.iso) » or the first .iso file available at <https://cdimage.debian.org/cdimage/release/current/amd64/iso-cd/>

- What to look for on the page :

#### Other questions?

See the Debian CD [FAQ](#) for lots more information about Debian CDs and installation.

The images here were put together by the [Debian CD team](#) , using debian-cd and other software.

Name	Last modified	Size
<a href="#">Parent Directory</a>	-	-
<a href="#">SHA256SUMS</a>	2024-02-10 17:41	302
<a href="#">SHA256SUMS.sign</a>	2024-02-10 21:13	833
<a href="#">SHA512SUMS</a>	2024-02-10 17:41	494
<a href="#">SHA512SUMS.sign</a>	2024-02-10 21:13	833
<a href="#">debian-12.5.0-amd64-netinst.iso</a>	2024-02-10 14:46	629M
<a href="#">debian-edu-12.5.0-amd64-netinst.iso</a>	2024-02-10 14:46	637M
<a href="#">debian-mac-12.5.0-amd64-netinst.iso</a>	2024-02-10 14:46	627M

#### 1-ISO-image-page

- Example of SHA512SUMS :

cdimage.debian.org/cdimage/release/current/amd64/iso-cd/SHA512SUMS	
33c08e56c83d13007e4a5511b9bf2c4926c4aa12fd5dd56d493c0653aebab380988c5bf1671dbaa75c582827797d98c4a611f7fb2b131fbde2c677d5258ec97cd6ec7d359515117558e91650fce1b98af0082cb2149b3d94fb480f657f51b06934d7b450fe563eeef023ccf59472b8faa870ee50c369f4c4ceceea8da0c60f968067c0874b8cc865f7c0ff802b86ba09013a20f0318e39b38e0e37af5c557ccd5145ae7a2f8be01492a5298386aba437b545471450f9ae80fda6d6dac5e454d	debian-12.5.0-amd64-netinst.iso debian-edu-12.5.0-amd64-netinst.iso debian-mac-12.5.0-amd64-netinst.iso

#### 2-SHA512SUMS

## 3. Installing Debian 12

### 1.2 Launch installation:

Use the following script to launch the Debian installation:

```
$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=tcp::8080-:80,hostfwd=tcp::5432-:5432 -
cdrom YOUR/PATH/NAME_OF_YOUR_IMAGE.iso
```

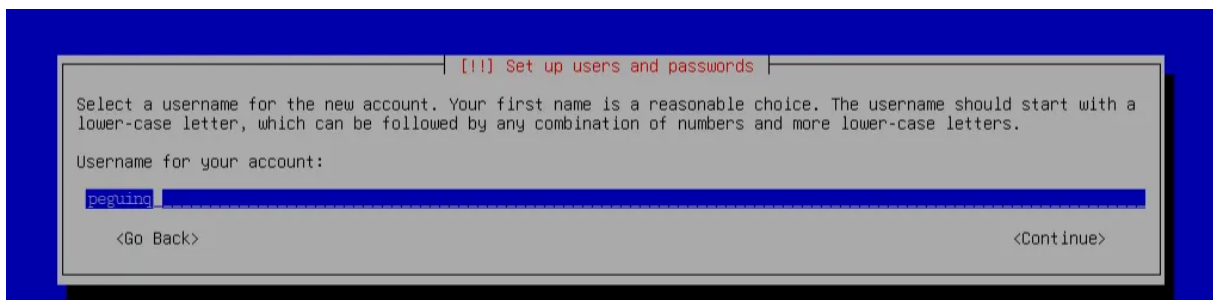
---

## 1.3 Debian configuration :

Throughout the guide, if you see 'Your\_login\_UGA', enter the name you would like to have. For me, this name is 'peguinq'.

Follow the installation steps, accepting the default choices except those shown below:

- Language: English
- Location: other/Europe/France
- Locales: United States, en\_US.UTF-8
- Keyboard: French
- Hostname: use server-"YOUR\_UGA\_LOGIN"

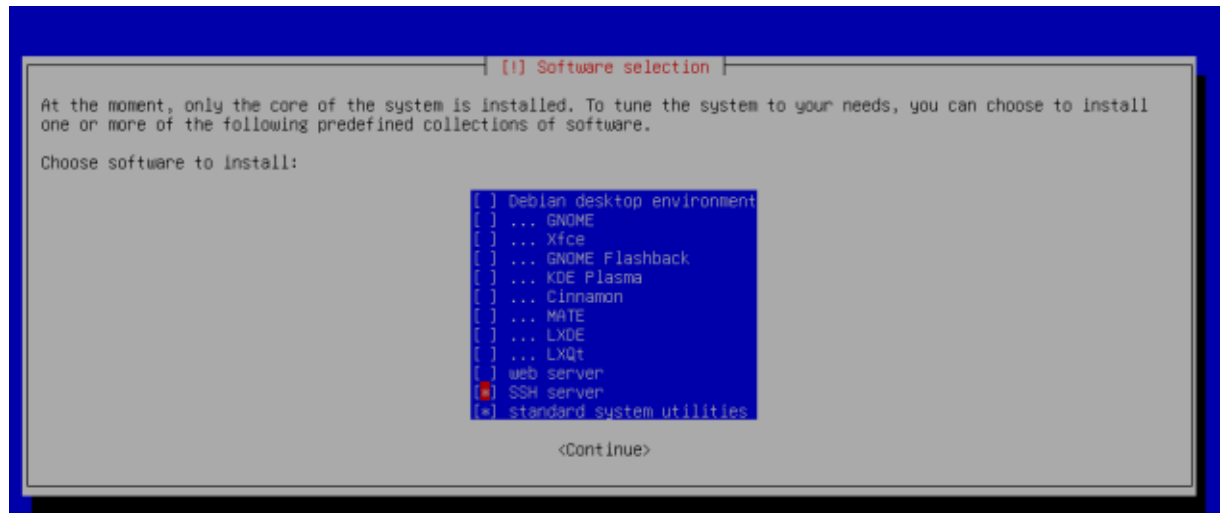


### 3-Setup-user

- Root Password: a simple password is advised, for example, "root." In my context, it does not pose a security problem but do it seriously. Check the "Show Password" box to ensure that the entered password is indeed the one you want.
- User Account - Full Name: your full name, for example, "Jean Toto"
- User Name: enter your UGA login name
- User Password: enter a simple password, for example, "etu." Check the "Show Password" box to ensure that the entered password is indeed the one you want.
- Partition disks: Guided - use entire disk
- Partition disks: All files in one partition
- Partition disks: Yes



- Software Selection: ensure "Debian desktop" is not checked and "ssh server" is checked



#### 4-Software-selection

- Install GRUB : Yes
- Device for boot loader : `/dev/sda`

## 1.4 Finalizing the Installation:

Update the packages:

```
#apt update && apt upgrade
```

Once the installation is complete, shut down the virtual machine:

```
#poweroff
```

## 1.5 Installation Verification

- Launch the virtual machine :

Use the following line to launch the virtual machine:

```
$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=tcp::8080-:80,hostfwd=tcp::5432-:5432
```

Your Debian server being installed, by typing the command below, you can see the Ethernet and IP characteristics:

```
#ip addr
```

```
(peguinq) localhost:2222 — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View Copy
root@server-peguinq:~# 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 86311sec preferred_lft 86311sec
    inet6 fec0::5054:ff:fe12:3456/64 scope site dynamic mngtmpaddr
        valid_lft 86315sec preferred_lft 14315sec
    inet6 fe80::5054:ff:fe12:3456/64 scope link
        valid_lft forever preferred_lft forever
root@server-peguinq:~#
```

### 5-IP-and-Ethernet-characteristics

Here you can see my IPV4 address: 10.0.2.15/24

My IPV6 address: fec0::5054:ff:feda:3456/64

My Ethernet address: 52:54:00:12:34:56

- Check the mount points on your virtual machine :

Type the command

```
#cat /etc/fstab
```

As shown below:

```
QEMU
Machine View
peguinq@server-peguinq:~$ 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=9c8eb2c1-a24c-4b61-a22d-3563d1dd5b6d / ext4 errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=a08a49f5-b14c-4f80-8bbc-3d10eda343fc none swap sw 0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
peguinq@server-peguinq:~$
```

### 6-/cat/fstab

## 1.6 Connecting via SSH:

You can connect to your virtual machine from your personal machine because in the configuration for launching your virtual machine you have defined ports. For SSH this

is port 2222. Therefore, you need to launch your virtual machine and type the command:

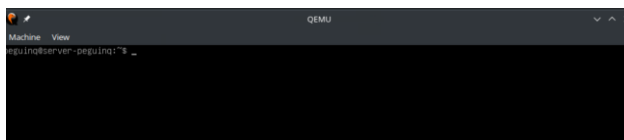
```
$ssh VOTRE_LOGIN@localhost -p 2222
```

From your physical machine.

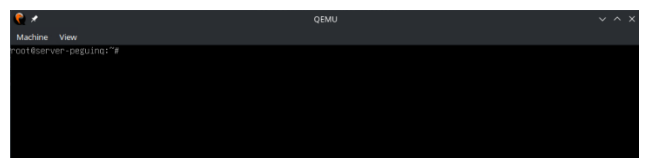
Remember to put your login in place of YOUR\_LOGIN!

- Here are the 4 useful terminals:

From your virtual machine

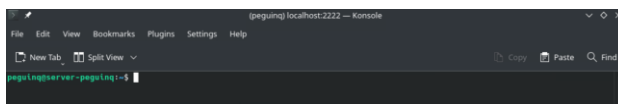


*8-Virtual-Machine-user*

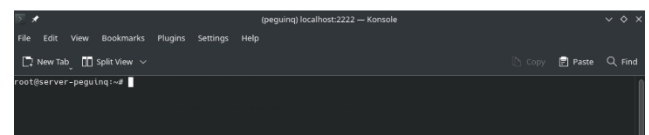


*7-Virtual-Machine-root*

From your physical machine :



*10-Physical-machine-user*



*9-Physical-Machine-root*

## 1.7 Checking for the Absence of Xorg Server:

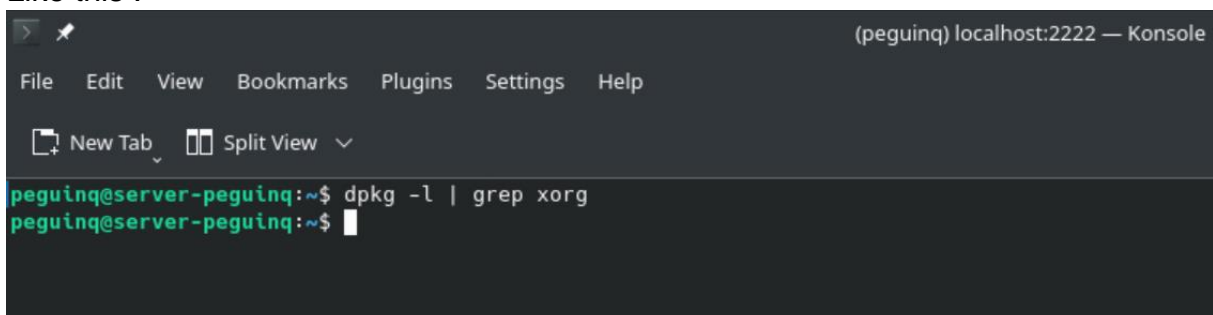
As our machine does not have a graphical server, let's verify.

- Type the command:

```
$dpkg -l | grep xorg
```

And you should not get any response (except for the command itself running during its search).

Like this :



*11-find-Xorg*



Here, we can notice we don't have graphical component.

---

## 4. Module installation

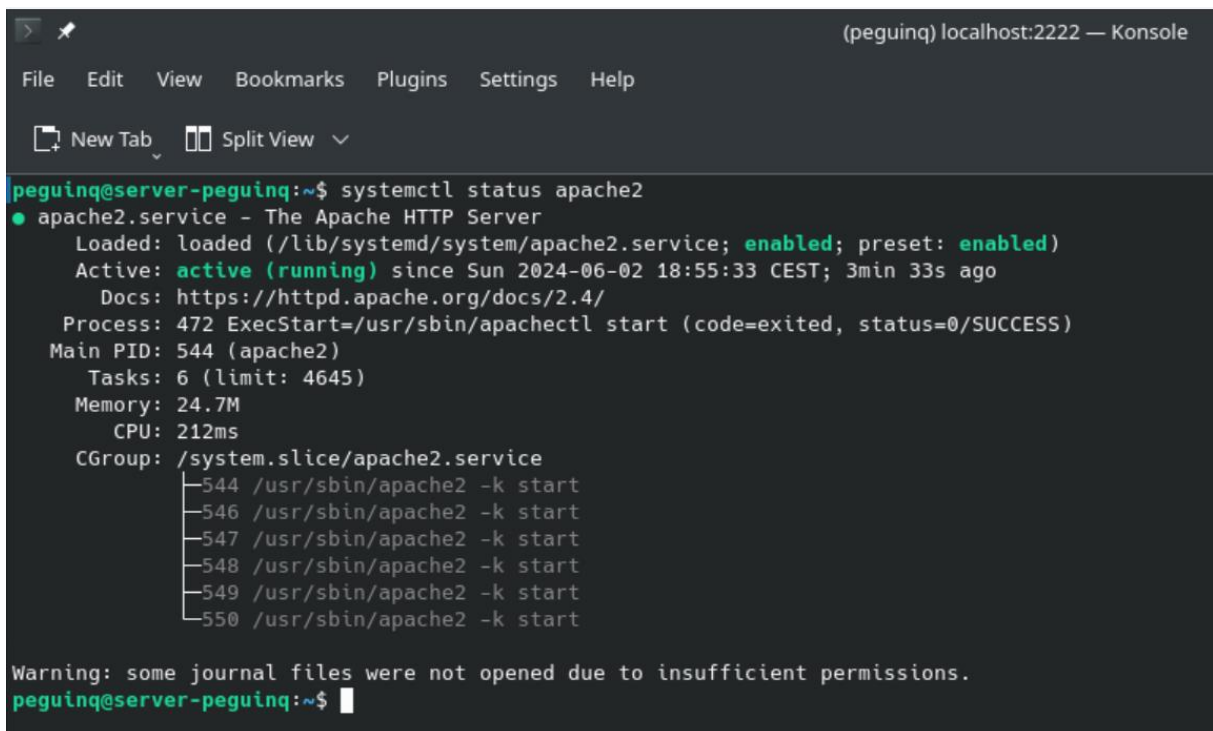
### 1.8 Apache2 installation:

- Install Apache2 using the following command:

```
#apt install apache2
```

- Check that the Apache service has been started:

```
$systemctl status apache2
```



The screenshot shows a terminal window titled "(peguinq) localhost:2222 — Konsole". The terminal output for the command `systemctl status apache2` is as follows:

```
peguinq@server-peguinq:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-06-02 18:55:33 CEST; 3min 33s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 472 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 544 (apache2)
    Tasks: 6 (limit: 4645)
   Memory: 24.7M
      CPU: 212ms
   CGroup: /system.slice/apache2.service
           └─544 /usr/sbin/apache2 -k start
             └─546 /usr/sbin/apache2 -k start
               └─547 /usr/sbin/apache2 -k start
                 └─548 /usr/sbin/apache2 -k start
                   └─549 /usr/sbin/apache2 -k start
                     └─550 /usr/sbin/apache2 -k start

Warning: some journal files were not opened due to insufficient permissions.
peguinq@server-peguinq:~$
```

*12-Apache2-status*

If everything is green, it means your Apache is running!

- From your workstation, open the URL:

<http://localhost:8080/>

- You should see this page open:



### 13-Apache2-Default-Page

This means it is working properly.

## 1.9 PostgreSQL installation :

```
#apt install postgresql
```

### 1.9.1 Verify the installation:

With the command:

```
$systemctl status postgresql
```

You should get a screenshot like this, showing that PostgreSQL is successfully installed and running.

```
(peguinq) localhost:2222 — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View
peguinq@server-peguinq:~$ systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; preset: enabled)
   Active: active (exited) since Sun 2024-06-02 18:55:36 CEST; 7min ago
   Process: 605 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 605 (code=exited, status=0/SUCCESS)
      CPU: 1ms

Warning: some journal files were not opened due to insufficient permissions.
peguinq@server-peguinq:~$
```

#### 14-Status-postgresql

If everything is good, you can start the configuration.

## 1.9.2 Configuration :

To see the list of default PostgreSQL databases, type:

```
$psql - l
```

(There will be a screenshot later to show how successfully you created your table)

The following steps are demonstrations and can be adapted and modified according to your wishes and knowledge. They show that the basic commands can be executed without error.

- Creating a User:

Next, let's create a superuser who will have permission to create databases by connecting with their password.

```
CREATE USER peguinq WITH SUPERUSER CREATEDB PASSWORD 'peguinq123';
```

(`peguinq` must be changed to the name you want to give to your user; `peguinq123` must be changed to the password you want to give to your user)

Once your user has been created, you can view it by typing :

```
\du
```

- Creating a Database:

Then we can create a database to verify that everything is working.

```
create database test owner peguinq;
```

Once you have created your database, you can check its creation with the command seen above (`psql -l`), the result of which is as follows.

```
(peguinq) localhost:2222 — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View
peguinq@server-peguinq:~$ psql -l
      List of databases
  Name | Owner  | Encoding | Collate | Ctype  | ICU Locale | Locale Provider | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8    | en_US.UTF-8 | en_US.UTF-8 |             | libc            | =c/postgres +
template0 | postgres | UTF8    | en_US.UTF-8 | en_US.UTF-8 |             | libc            | postgres=CTc/postgres +
template1 | postgres | UTF8    | en_US.UTF-8 | en_US.UTF-8 |             | libc            | =c/postgres +
test      | peguinq  | UTF8    | en_US.UTF-8 | en_US.UTF-8 |             | libc            | postgres=CTc/postgres
(4 rows)

peguinq@server-peguinq:~$
```

### 15-Postgres-Database

(test should be the name you want to give your database; peguinq should be the user created earlier)

Connect to your database and execute the following command:

```
\c test
```

This will connect you as the PostgreSQL user, and you can modify the database as you wish.

- Creating a Table:

```
CREATE TABLE utilisateur (
    id SERIAL PRIMARY KEY,
    nom VARCHAR(50),
    prenom VARCHAR(50),
    age INT
);
```

- Values insertion:

```
INSERT INTO utilisateur (nom, prenom, age) VALUES
('Dupont', 'Jean', 30),
('Martin', 'Marie', 25),
('Dubois', 'Pierre', 35);
```

## 1.9.3 Useful Changes:

To be able to access your database from your Linux workstation, some modifications need to be made to the PostgreSQL configuration on your virtual machine.

- First Modification:

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

(Use Ctrl + W to search within the document)

Find the line:

```
listen_addresses = 'localhost'
```

And replace it with:

```
listen_addresses = '*'
```

Remember to uncomment the line! This is a common mistake.

- Second Modification:

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

Add to the file:

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

After making these changes, restart your machine:

```
#service postgresql restart
```

- To test that everything is working correctly:

```
psql -h localhost -U peguing postgres
```

To connect (Password: 'your\_password')

You can check user with password in the pg\_shadow table:

```
Postgres=#select * from pg_shadow;
```

Here there is my own screenshot :

```
postgres=# select * from pg_shadow ;
username | usesysid | usecreatedb | usesuper | userepl | usebypassrls |          | valuntil | useconfig
-----+-----+-----+-----+-----+-----+-----+-----+-----
postgres |      10 | t           | t        | t        | t            |          |          |
peguinq  |   16387 | t           | f        | f        | f            | SCRAM-SHA-256$4096:VcvZiPHvn4lW5B8GF1D.jsQ==$PjsZ+UouRLD
+fdi//If52vCFZxHF+f8+HG8D7uknr0s=:eej+RCdc99vBEc1xH1haSXpNkREXZi2ULLCbXNN/yAY= |          |
(2 rows)

(END)
```

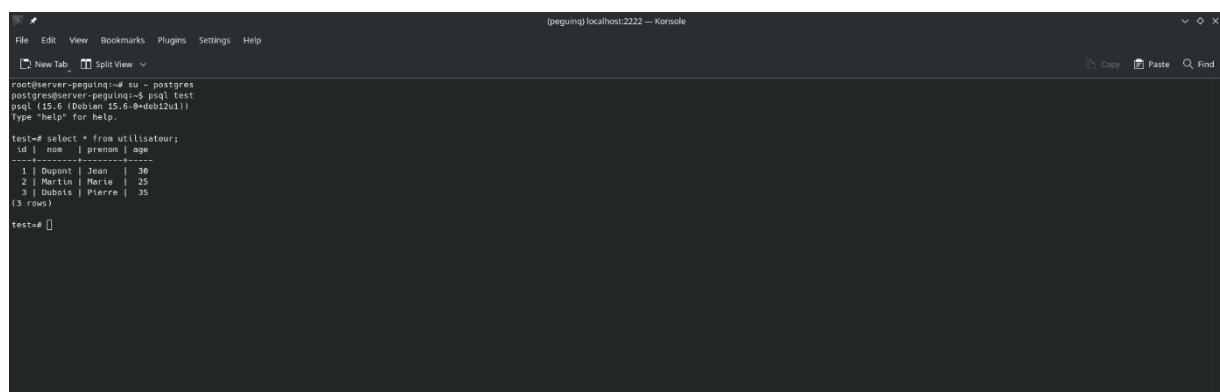
## 16-Postgresql-pg\_shadow

To log out from PostgreSQL:

```
postgres@server-peguinq:~$ logout
```

You can test from your virtual machine as below:

```
test=#select * from utilisateur;
```



```
(peguinq) localhost:2222 --- Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View
root@server-peguinq:~# su - postgres
postgres@server-peguinq:~$ psql test
psql (15.6 (Debian 15.6-0+deb12u1))
Type "help" for help.

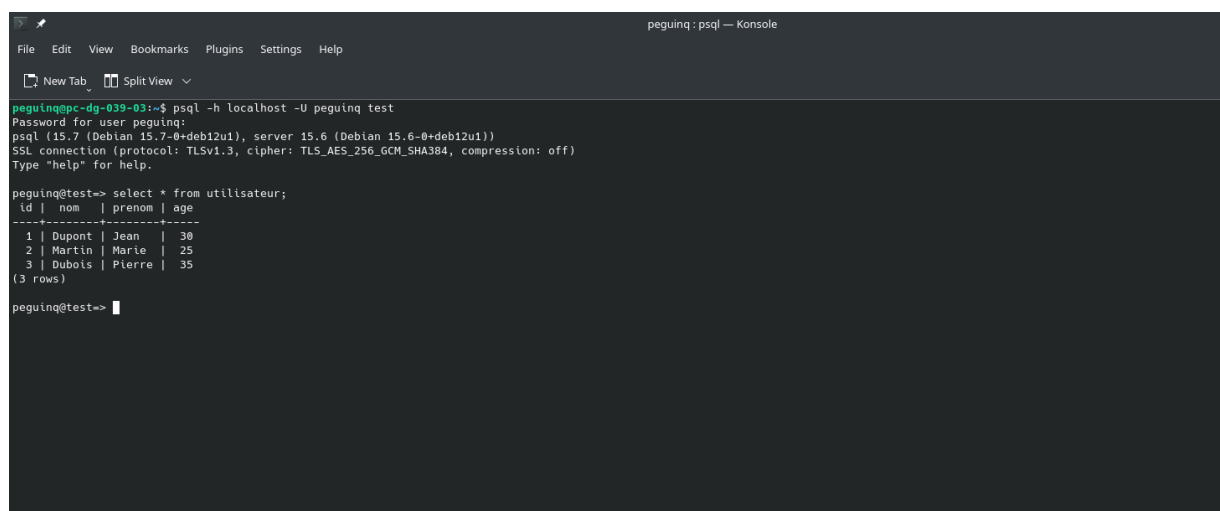
test=# select * from utilisateur;
 id | nom | prenom | age
----+----+-----+---
  1 | Dupont | Jean   | 30
  2 | Martin | Marie  | 25
  3 | Dubois | Pierre | 35
(3 rows)

test=#
```

## 17-Postgresql-test-mv

Or from your personal machine as I did below:

```
test=#select * from utilisateur;
```



```
peguinq: psql --- Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View
peguinq@pc-dg-039-03:~$ psql -h localhost -U peguinq test
Password for user peguinq:
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.

peguinq@test=> select * from utilisateur;
 id | nom | prenom | age
----+----+-----+---
  1 | Dupont | Jean   | 30
  2 | Martin | Marie  | 25
  3 | Dubois | Pierre | 35
(3 rows)

peguinq@test=>
```

## 18-Postgresql-test-pm



---

## 1.10 PhpPgAdmin installation

### 1.10.1 Let's start by installing PHP:

This is done with the command:

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

For PHP, we won't install any additional modules, but you can find more information on installing PHP on Debian at the following address:

If you need to restart or stop PHP, here are the commands:

```
#systemctl stop apache2  
#systemctl start apache2
```

### 1.10.2 PhpPgAdmin installation:

Next, you need to install PhpPgAdmin:

```
#apt install phppgadmin
```

Find the file:

```
#find . -name Connection.php
```

Edit the file:

```
nano /VOTRE/CHEMIN/Connection.php
```

Change:

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

To

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

- Edit the Apache configuration for PhpPgAdmin:

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

Change

```
Require local
```

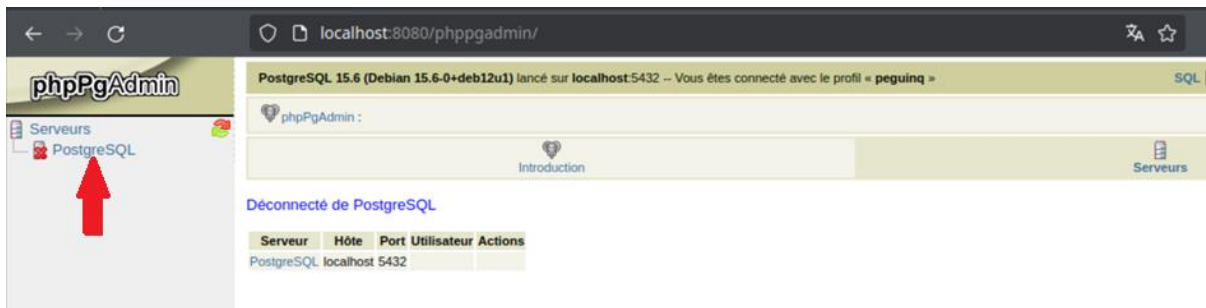
to:

```
Allow from all
```

### 1.10.3 Test your database from PhpPgAdmin:

Go to the following URL:

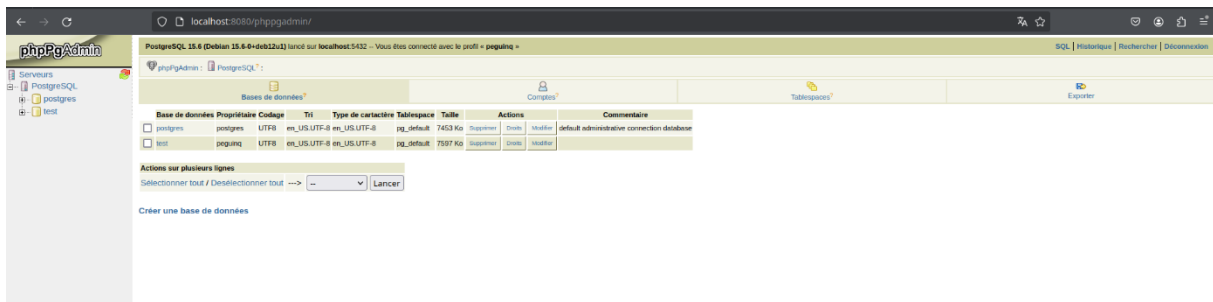
<http://localhost:8080/phpPgAdmin/>



19-PhpPgAdmin-home

You will need to log in with your PostgreSQL credentials.

Then you will arrive at this page; click on your database.

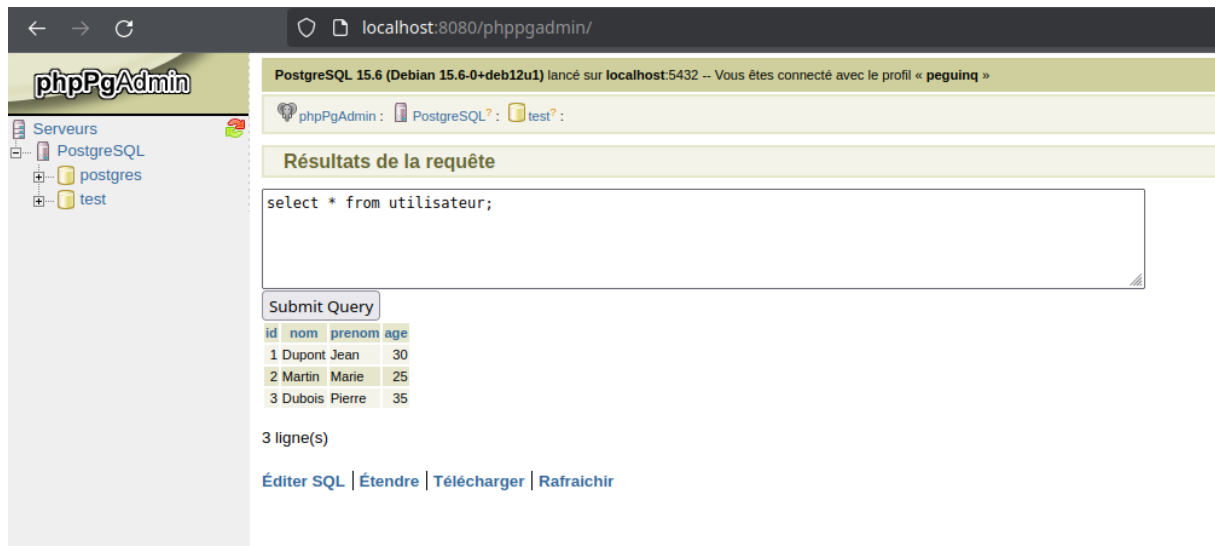


20-PhpPgAdmin-database

By clicking on your database (for me, test) and then on SQL, you can enter a command.

Personally, I entered the same as before:

```
Select * from utilisateur ;
```



21-PhpPgAdmin-test

## 1.10.4 Test PHP files:

- To see your PHP configuration:

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

Type on your virtual machine:

```
#/sbin/blkid
```

Download the file below to your personal machine (make sure it is in PHP format and not HTML):

[page\\_sae\\_S2.03.php](#)

Then use this command from your personal machine:

```
$scp -P 2222 ~/page_sae_S2.03.php penguin@localhost:~/page_sae_S2.03.php
```

Don't forget to replace "penguin" with your login.

Place this file in /var/www/html/ and access it at:

[http://localhost:8080/page\\_de\\_test.php](http://localhost:8080/page_de_test.php)

from your browser on your personal machine.

```
localhost:8080/page_sae_s2.03.php

Bonjour

Je suis www-data

Qui est connecté ?

peguinq tty1 May 31 13:47
peguinq pts/0 May 31 13:58 (10.0.2.2)
peguinq pts/1 May 31 14:18 (10.0.2.2)

Mes disques sont

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 83397sec preferred_lft 83397sec
    inet6 fec0::5054:fff:fe12:3456/64 scope site dynamic mngtmpaddr
        valid_lft 86129sec preferred_lft 14129sec
    inet6 fe80::5054:fff: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-php 2:8.2+93 all server-side, HTML-embedded scripting language (Apache 2 module)
ii libapache2-mod-php8.2 8.2.18-1-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 Fri 2024-05-31 13:46:27 CEST; 50min ago
Docs: https://httpd.apache.org/docs/2.4/
Process: 477 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
Main PID: 520 (apache2)
Tasks: 13 (limit: 4645)
Memory: 35.2M
CPU: 735ms
CGroup: /system.slice/apache2.service
├─ 520 /usr/sbin/apache2 -k start
├─ 520 /usr/sbin/apache2 -k start
├─ 531 /usr/sbin/apache2 -k start
├─ 532 /usr/sbin/apache2 -k start
├─ 534 /usr/sbin/apache2 -k start
├─ 1357 /usr/sbin/apache2 -k start
├─ 1358 /usr/sbin/apache2 -k start
├─ 1359 /usr/sbin/apache2 -k start
└─ 1360 /usr/sbin/apache2 -k start
```

22-PhpPgAdmin-pagetest

## 5. Security and Finalization:

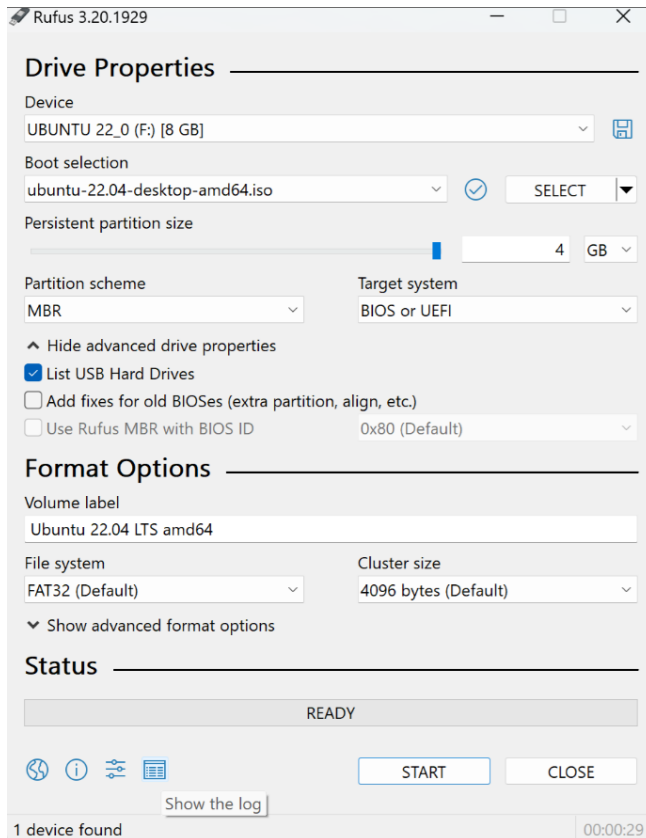
### 1.11 Finalization:

Boot on a USB-Stick:

To have your virtual machine in your USB-stick you need to

Download and install [Rufus.ie](https://rufus.ie)

- Put your USB-Stick in your computer
- Open Rufus and select your USB drive, then choose the file.iso you downloaded.
- Click on start to launch the process.



### 23-rufus-installation

- Insert the USB in the computer where you want to launch your virtual machine.
- Restart your computer and enter in the BIOS settings.
- Set the USB drive as the primary boot device.
- Save the changes and exit the BIOS.

You can restart the different services:

```
$systemctl status (postgres,ssh,apache2)
```

```

postgres@server-peguin:~$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Mon 2024-05-27 16:10:41 CEST; 53min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 472 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 491 (sshd)
    Tasks: 1 (limit: 4645)
   Memory: 8.4M
      CPU: 85ms
   CGroup: /system.slice/ssh.service
           └─491 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

postgres@server-peguin:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Mon 2024-05-27 16:30:07 CEST; 33min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 926 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 932 (apache2)
    Tasks: 9 (limit: 4645)
   Memory: 21.8M
      CPU: 335ms
   CGroup: /system.slice/apache2.service
           └─932 /usr/sbin/apache2 -k start
             └─933 /usr/sbin/apache2 -k start
               └─934 /usr/sbin/apache2 -k start
                 └─935 /usr/sbin/apache2 -k start
                   └─936 /usr/sbin/apache2 -k start
                     └─937 /usr/sbin/apache2 -k start
                       └─938 /usr/sbin/apache2 -k start
                         └─939 /usr/sbin/apache2 -k start
                           └─940 /usr/sbin/apache2 -k start

postgres@server-peguin:~$ systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; preset: enabled)
   Active: active (exited) since Mon 2024-05-27 16:10:43 CEST; 53min ago
   Process: 534 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
  Main PID: 534 (code=exited, status=0/SUCCESS)
      CPU: 893us

```

#### 24-Status-Module

Then you can check your remaining space as shown below:

```
$df -h
```



```
(peguinq) localhost:2222 — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View
peguinq@server-peguinq:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G   0    1.9G   0% /dev
tmpfs           392M  508K  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
peguinq@server-peguinq:~$
```

25-space-after-installation

## 1.12 Security Analysis:

Security updates: Make sure that security updates are applied regularly:

```
#apt update && apt upgrade
```

Here we can't see but I write this line and we can saw all the update and upgrade :

```
File Edit View QEMU
Machine View
peguinq@server-peguinq:~$ apt update && apt upgrade
Unpacking libc-l10n (2.36-9+deb12u7) over (2.36-9+deb12u4) ...
Preparing to unpack .../06-locales_2.36-9+deb12u7_all.deb ...
Unpacking locales (2.36-9+deb12u7) over (2.36-9+deb12u4) ...
Preparing to unpack .../07-libglib2.0-0_2.74.6-2+deb12u2_amd64.deb ...
Unpacking libglib2.0-0:amd64 (2.74.6-2+deb12u2) over (2.74.6-2) ...
Preparing to unpack .../08-libglib2.0-data_2.74.6-2+deb12u2_all.deb ...
Unpacking libglib2.0-data (2.74.6-2+deb12u2) over (2.74.6-2) ...
Selecting previously unselected package linux-image-6.1.0-21-amd64.
Preparing to unpack .../09-linux-image-6.1.0-21-amd64_6.1.90-1_amd64.deb ...
Unpacking linux-image-6.1.0-21-amd64 (6.1.90-1) ...
Preparing to unpack .../10-linux-image-amd64_6.1.90-1_amd64.deb ...
Unpacking linux-image-amd64 (6.1.90-1) over (6.1.76-1) ...
Setting up libc-l10n (2.36-9+deb12u7) ...
Setting up linux-image-6.1.0-21-amd64 (6.1.90-1) ...
I: /vmlinuz is now a symlink to boot/vmlinuz-6.1.0-21-amd64
I: /initrd.img is now a symlink to boot/initrd.img-6.1.0-21-amd64
/etc/kernel/postinst.d/initramfs-tools:
update-initramfs: Generating /boot/initrd.img-6.1.0-21-amd64
/etc/kernel/postinst.d/zz-update-grub:
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-6.1.0-21-amd64
Found initrd image: /boot/initrd.img-6.1.0-21-amd64
Found linux image: /boot/vmlinuz-6.1.0-18-amd64
Found initrd image: /boot/initrd.img-6.1.0-18-amd64
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
Setting up libglib2.0-0:amd64 (2.74.6-2+deb12u2) ...
No schema files found: doing nothing.
Setting up less (590-2.1~deb12u2) ...
Setting up linux-image-amd64 (6.1.90-1) ...
Setting up apache2-bin (2.4.59-1~deb12u1) ...
Setting up locales (2.36-9+deb12u7) ...
```

26-Update-Upgrade

## 6. Conclusion :

This guide provides the essential steps to install and configure a Debian 12 server with Apache, PostgreSQL, and PHP on a Qemu/KVM virtual machine. If you have any problem look the documentation on the problem.

## 7. Controls and documentations:

### 1.13 Virtual machine management :

<pre>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=tcp::8080- :80,hostfwd=tcp::5432-:5432</pre>	To launch your virtual machine
<pre>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=tcp::8080- :80,hostfwd=tcp::5432-:5432 -cdrom YOUR/PATH/NAME_OF_YOUR_IMAGE.iso</pre>	To launch the installation of your virtual machine
<pre>ssh YOUR_LOGIN@localhost -p 2222</pre>	Connect to VM via SSH
<pre>poweroff</pre>	Shut Down Virtual Machine

### 1.14 Web server management:

<pre>apt install apache2</pre>	Install Apache2
<pre>systemctl stop apache2</pre>	Stop Apache
<pre>systemctl start apache2</pre>	Start Apache

systemctl status apache2	Verify Apache2 Service
https://localhost:4443/	Connect to Apache Pages (HTTPS)
http://localhost:8080/	Connect to Apache Pages (HTTP)
http://localhost:8080/phpPgadmin/	Connect to PhpPgAdmin
http://localhost:8080/page_sae_S2.03.php/	Connect to PHP Test Page

## 1.15 Database management:

### System :

apt install postgresql	Install PostgreSQL
psql -h localhost -U YOUR_LOGIN postgres	Connect to PostgreSQL from Localhost
service postgresql restart	Restart PostgreSQL Service

### In postgres :

psql -h localhost -U YOUR_LOGIN postgres	Connect to PostgreSQL
logout	Disconnect from PostgreSQL
systemctl status postgresql	Verify PostgreSQL Service
CREATE USER YOUR_NAME WITH SUPERUSER CREATEDB PASSWORD 'YOUR_PASSWORD';	Create PostgreSQL Superuser, Create Database and Password
\du	List PostgreSQL Users
CREATE DATABASE YOUR_DATABASE;	Create PostgreSQL Database
\c YOUR_DATABASE	Connect to PostgreSQL Database
CREATE TABLE YOUR_TABLE (...);	Create Table in PostgreSQL
INSERT INTO YOUR_TABLE (...);	Insert Values into Table
psql -h localhost -U YOUR_LOGIN postgres	Connect to PostgreSQL from Localhost

SELECT * FROM YOUR_TABLE;	Query a Table
---------------------------	---------------

## 1.16 File management:

nano NAME_OF_FILE.xxx	Edit File with Nano
find . -name NAME.xxx	Search
df -h	Check Free Space
scp -P 2222 ~/YOUR_FILE YOUR_LOGIN@localhost:~/YOUR_FILE	Secure Copy to VM

## 1.17 System management:

apt install php-common libapache2-mod-php php-cli	Install PHP
apt update && apt upgrade	Update and Upgrade System
cat /etc/fstab	Search for Graphical Component
dpkg -l	List package installed
ip addr	Ethernet and IP Characteristics

## 1.18 Documentations:

Debian 12	<a href="https://www.debian.org/doc/index.fr.html">https://www.debian.org/doc/index.fr.html</a>
ISO Image	<a href="https://cdimage.debian.org/cdimage/release/current/amd64/iso-cd/">https://cdimage.debian.org/cdimage/release/current/amd64/iso-cd/</a>
SSH	<a href="https://wiki.debian.org/fr/SSH">https://wiki.debian.org/fr/SSH</a>
Apache2	<a href="https://httpd.apache.org/docs/2.4/fr/">https://httpd.apache.org/docs/2.4/fr/</a>
PostgreSQL	<a href="https://www.postgresql.org/docs/">https://www.postgresql.org/docs/</a>