Quentin PEGUIN
07/06/24
BUT1-INFO-D1

Debian 12

Debian 12 Server Installation Guide with Apache2, PostgreSQL and PhpPgAdmin

IUT2 Université Grenoble Alpes

Table of contents

Illustrations table

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.

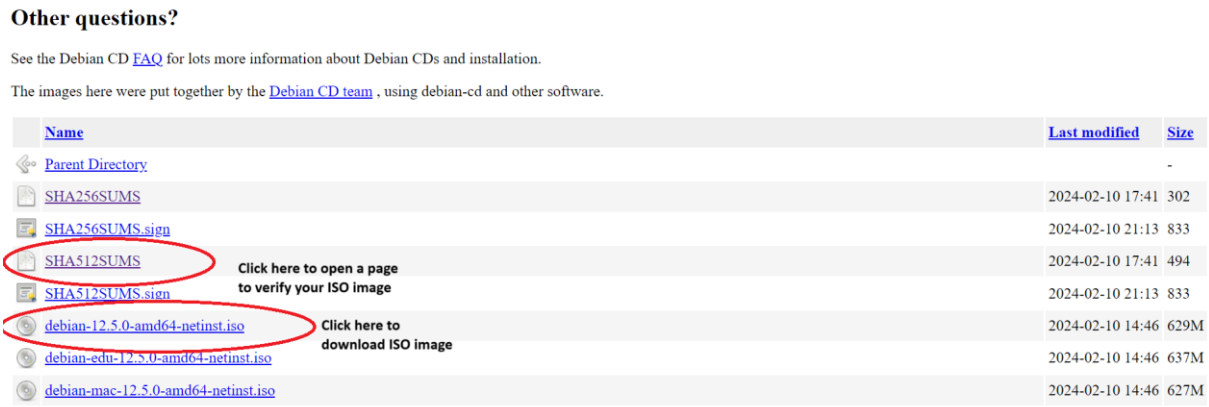
---

2. Preparing the installation

1.1 Preparing the ISO image :

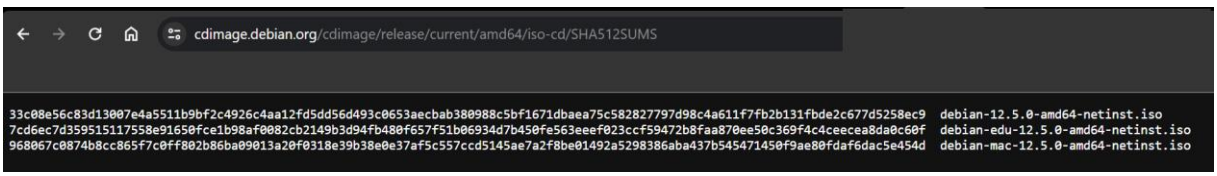
Download the virtual image called « debian-12.5.0-amd64-netinst.iso » or the first .iso file available at https://cdimage.debian.org/cdimage/release/current/amd64/iso-cd/

- What to look for on the page :



1-ISO-image-page

- Example of SHA512SUMS :



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’.

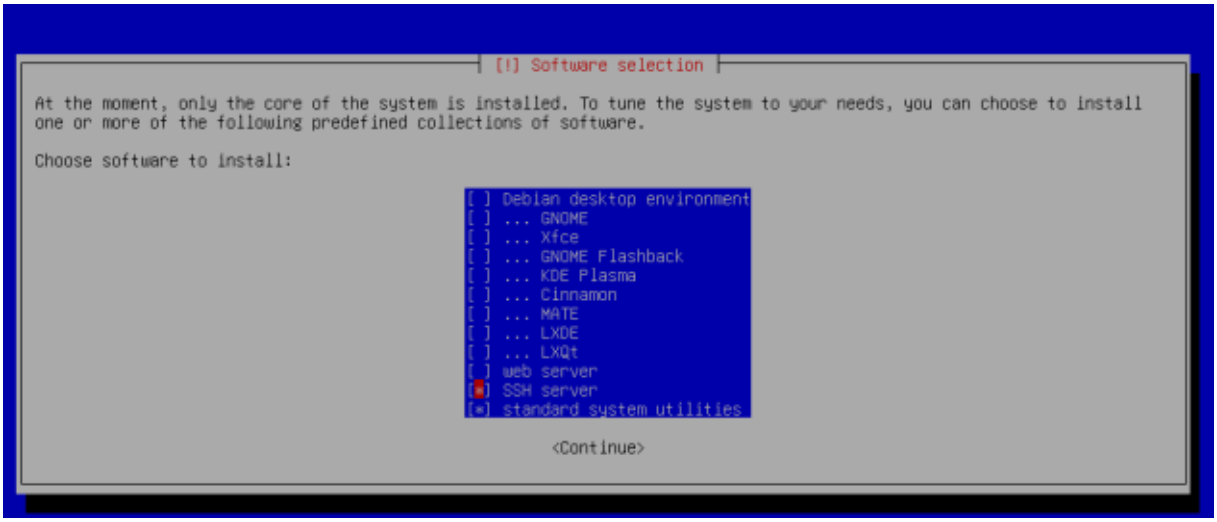
Follow the installation steps, accepting the default choices except those shown below:

- 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

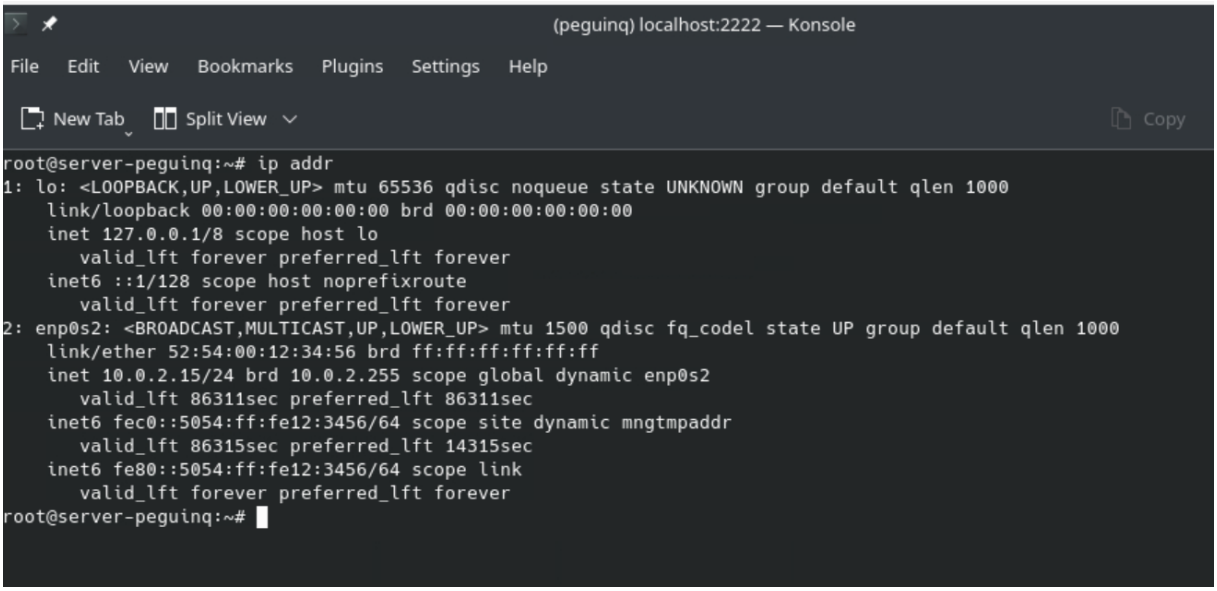
- Launch the virtual machine :

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
```



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

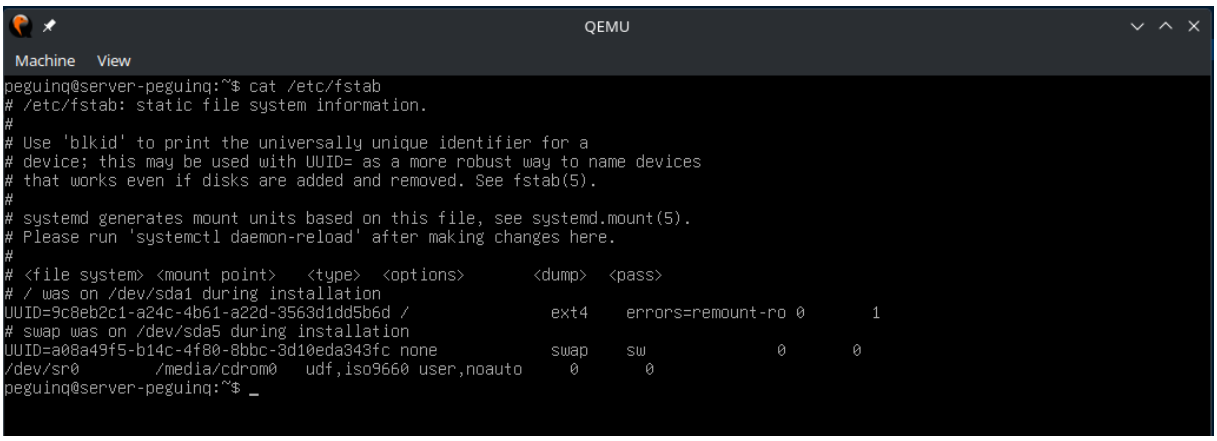
My Ethernet address: 52:54:00:12:34:56

- Check the mount points on your virtual machine :

Type the command

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

As shown below:



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:

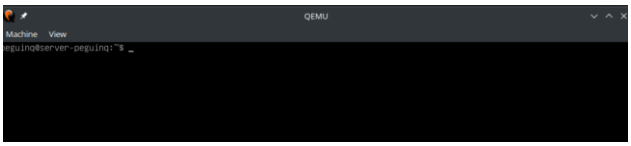
```
$ssh VOTRE_LOGIN@localhost -p 2222
```

From your physical machine.

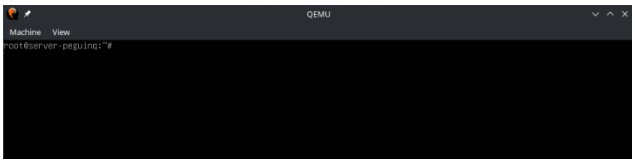
Remember to put your login in place of YOUR_LOGIN!

- Here are the 4 useful terminals:

From your virtual machine

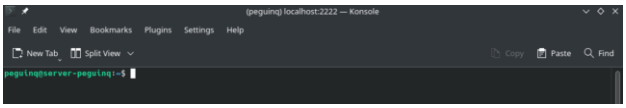


8-Virtual-Machine-user

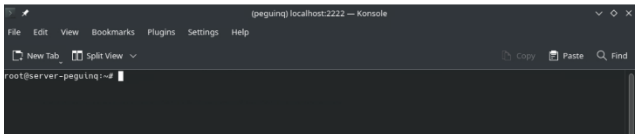


7-Virtual-Machine-root

From your physical machine :



10-Physical-machine-user



9-Physical-Machine-root

1.7 Checking for the Absence of Xorg Server:

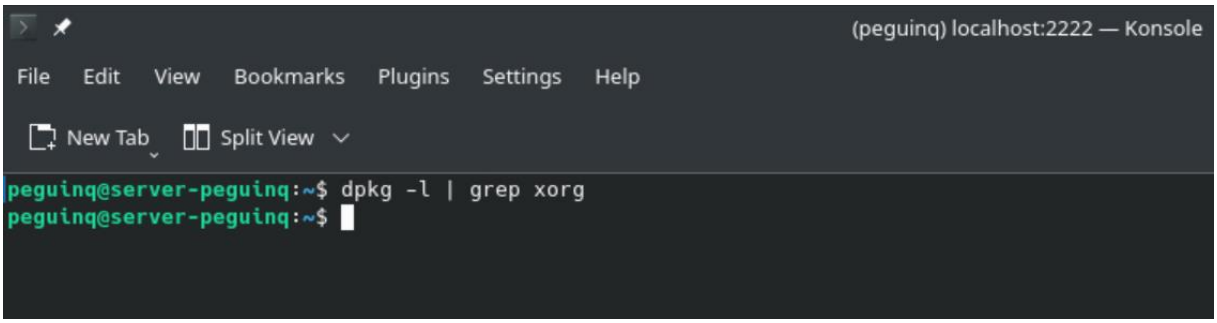
As our machine does not have a graphical server, let's verify.

- 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 dont have graphical component.

---

4. Module installation

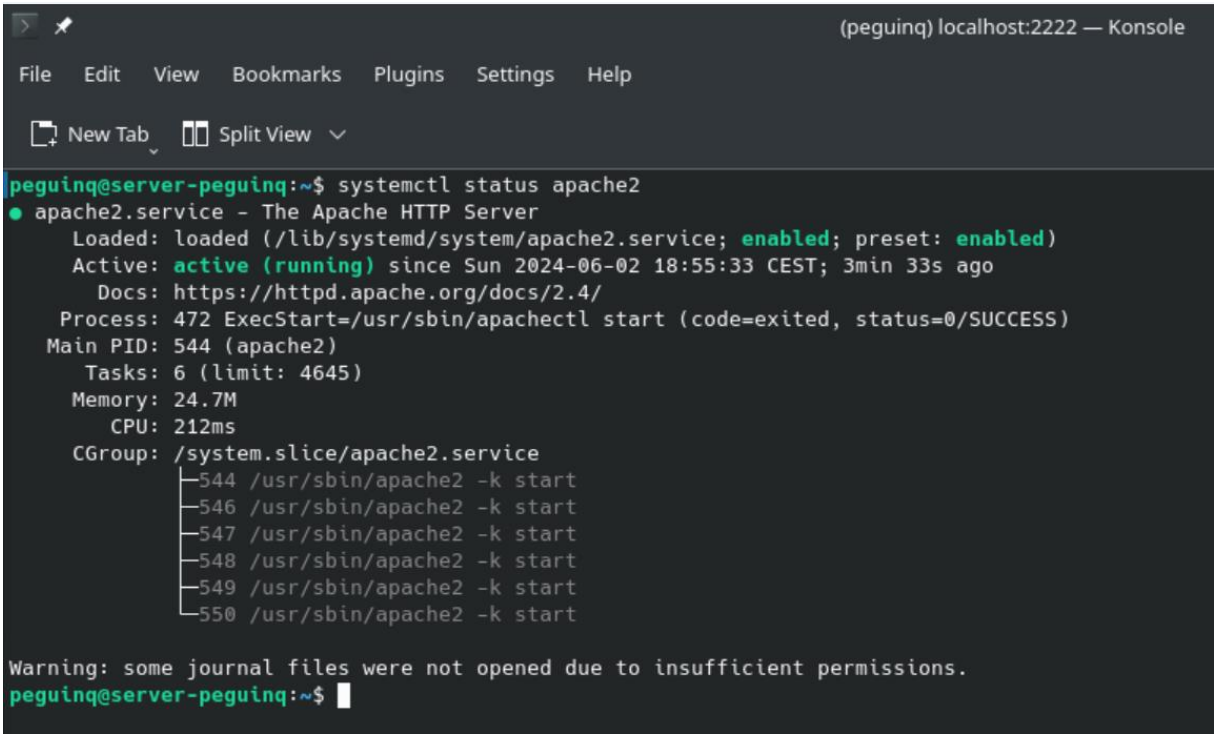
1.8 Apache2 installation:

- Install Apache2 using the following command:

```
#apt install apache2
```

- Check that the Apache service has been started:

```
$systemctl status apache2
```



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 :

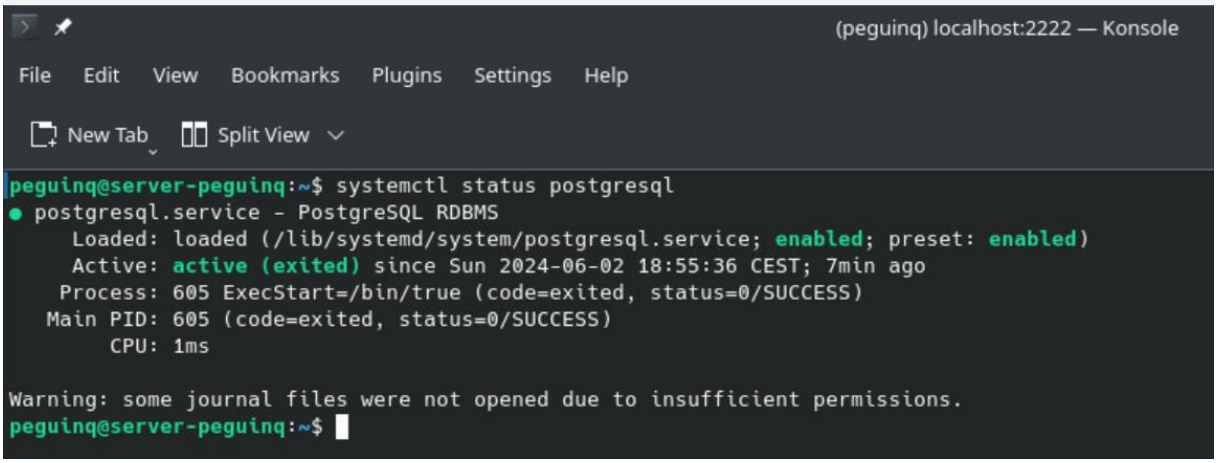
```
#apt install postgresql
```

1.9.1 Verify the installation:

With the command:

```
$systectl status postgresql
```

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



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 :

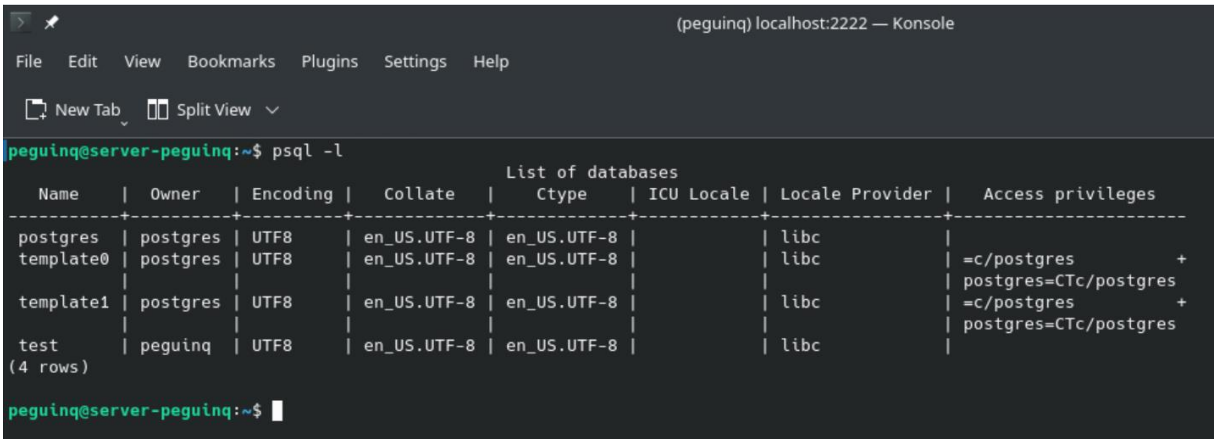
```
\du
```

- 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.



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 peguinq 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 :

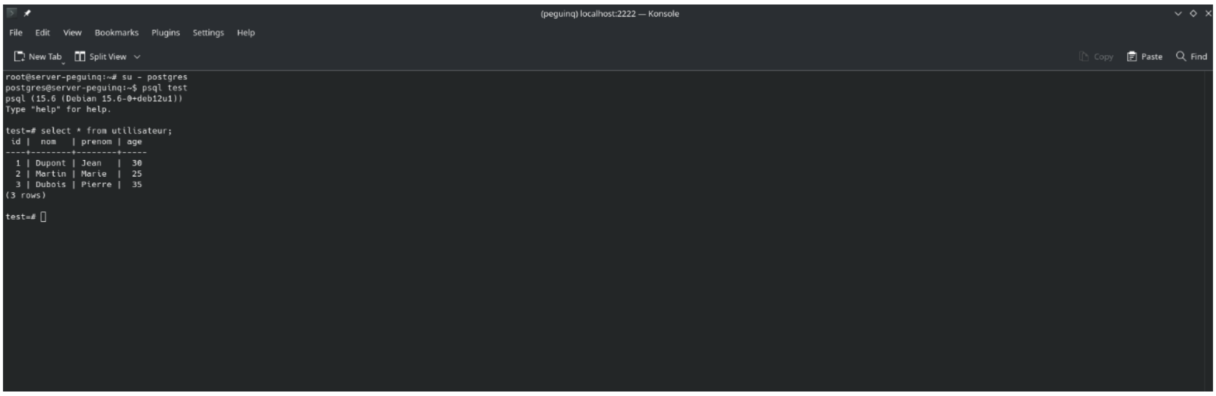
16-Postregresql-pg_shadow

To log out from PostgreSQL:

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

You can test from your virtual machine as below:

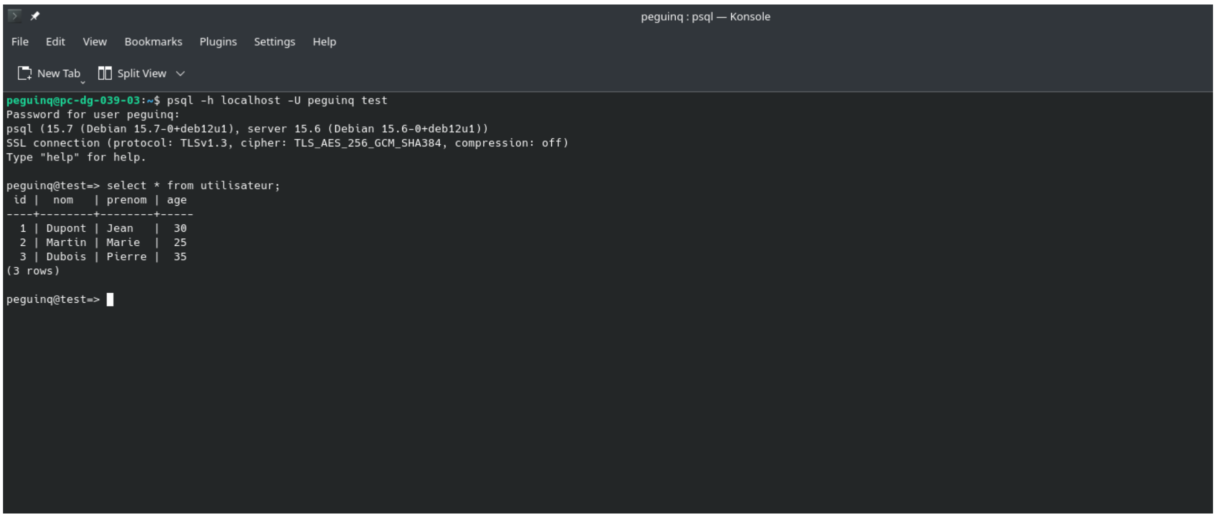
```
test=#select * from utilisateur;
```



17-Postgresql-test-mv

Or from your personal machine as I did below:

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



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
```

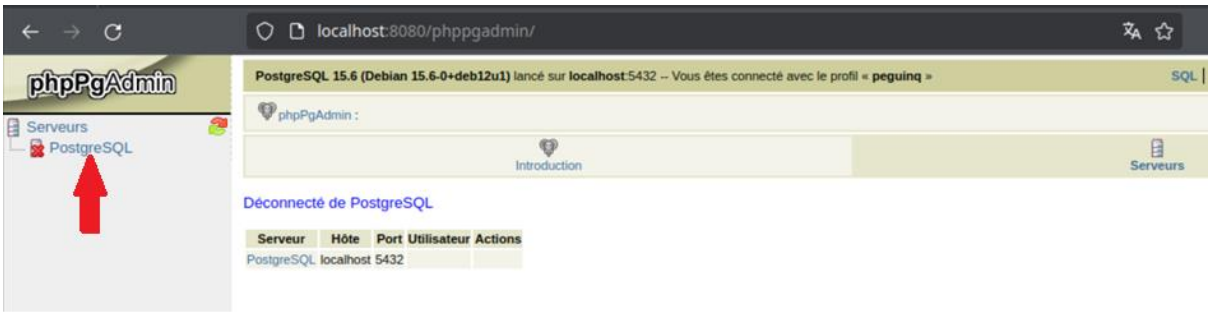
to:

```
Allow from all
```

1.10.3 Test your database from PhpPgAdmin:

Go to the following URL:

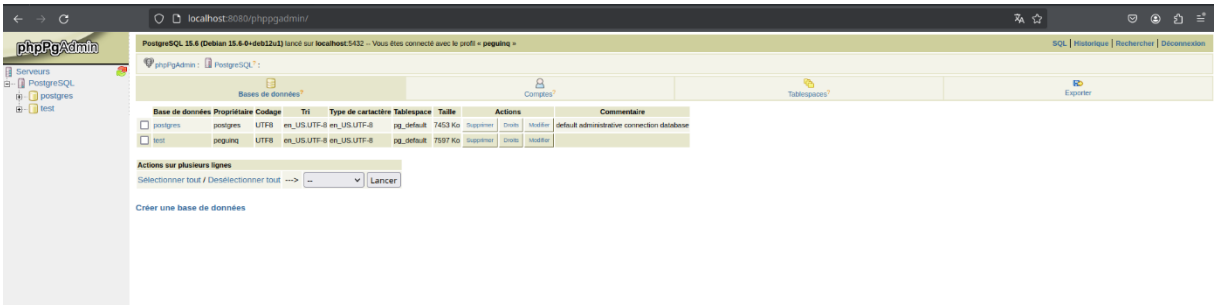
http://localhost:8080/phppgadmin/



19-PhpPgAdmin-home

You will need to log in with your PostgreSQL credentials.

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

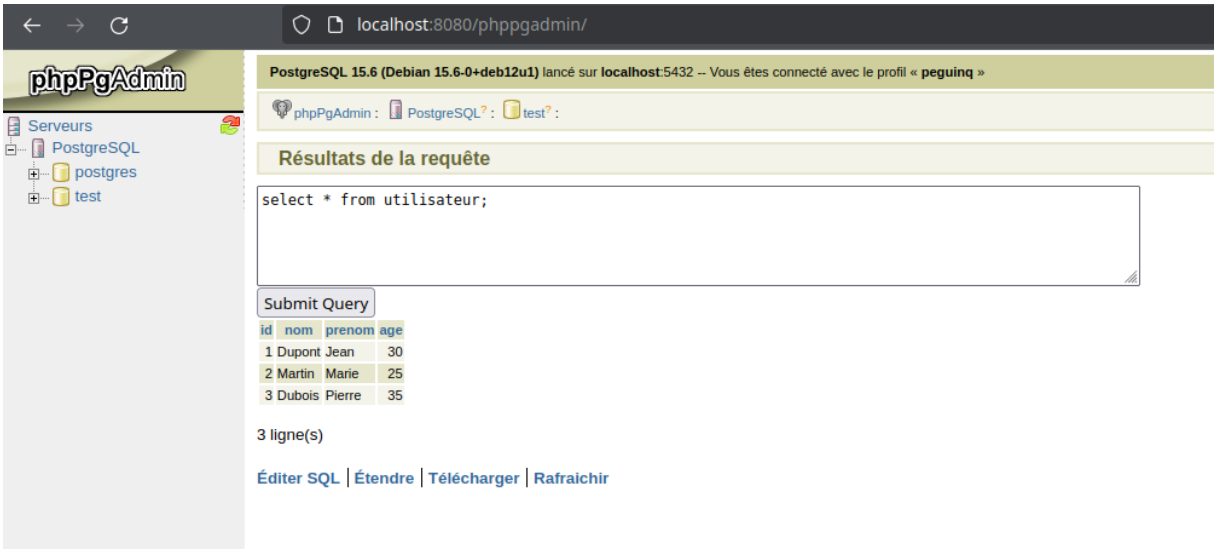


20-PhpPgAdmin-database

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 peguinq@localhost:~/page_sae_S2.03.php
```

Don't forget to replace "`peguinq`" with your login.

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

http://localhost:8080/page_de_test.php

from your browser on your personal machine.

22-PhpPgAdmin-pagetest

---

5. Security and Finalization:

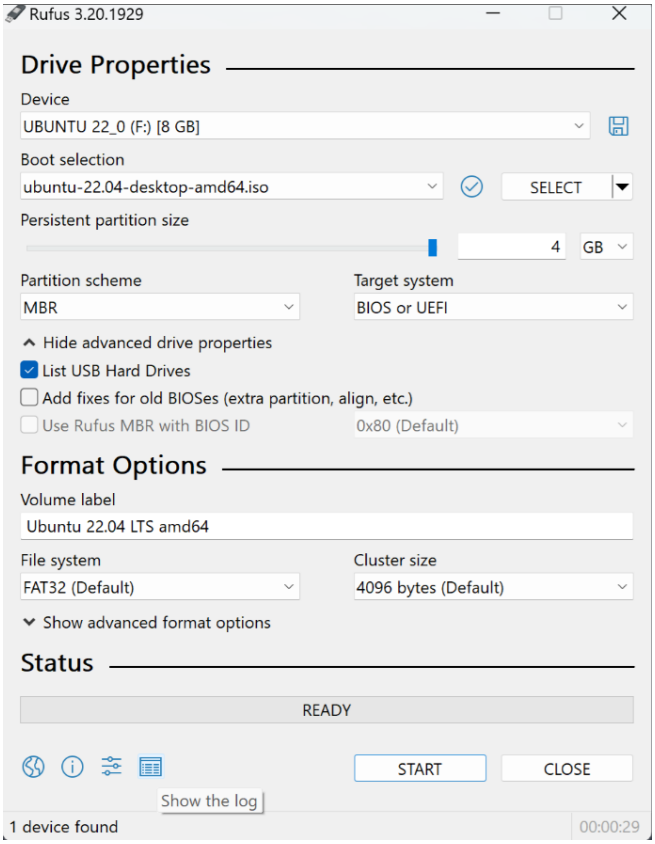
1.11 Finalization:

Boot on a USB-Stick:

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

Download and install Rufus.ie

- 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 BIOSS settings.
- Set the USB drive as the primary boot device.
- Save the changes and exit the BIOS.

You can restart the different services:

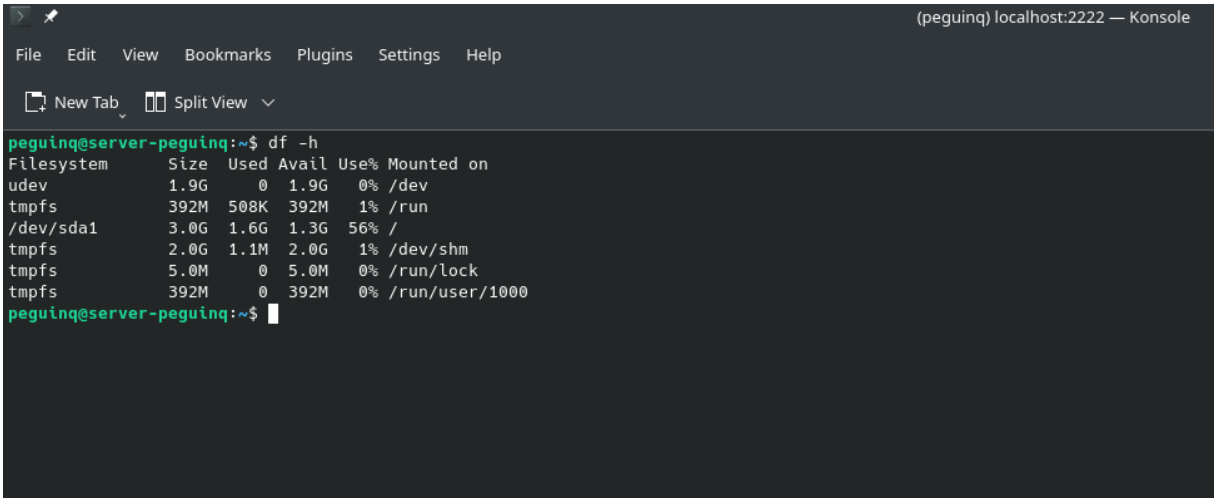
```
$systemctl status (postgres,ssh,apache2)
```

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



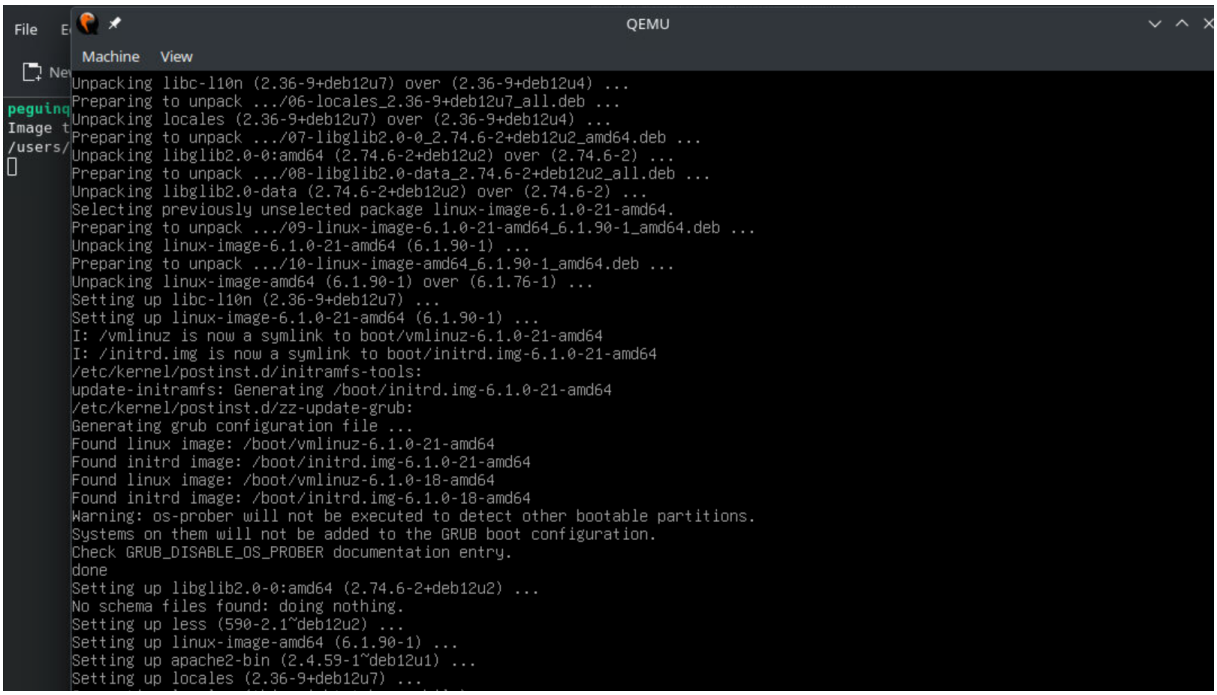
25-space-after-installation

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 :



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 :

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	To launch your 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 –cdrom YOUR/PATH/NAME_OF_YOUR_iMAGE.iso	To launch the installation of your virtual machine
ssh YOUR_LOGIN@localhost -p 2222	Connect to VM via SSH
poweroff	Shut Down Virtual Machine

1.14 Web server management:

apt install apache2	Install Apache2
systemct stop apache2	Stop Apache
systemct start apache2	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	https://www.debian.org/doc/index.fr.html
ISO Image	https://cdimage.debian.org/cdimage/release/current/amd64/iso-cd/
SSH	https://wiki.debian.org/fr/SSH
Apache2	https://httpd.apache.org/docs/2.4/fr/
PostgreSQL	https://www.postgresql.org/docs/