Installation Guide
Debian 12 Server with Apache, PostgreSQL and PHP

Author
Lucas FOSSE

Table des matières

Debian 12 Server Installation

This section details the process of installing Debian 12 on a virtual machine using Qemu/KVM. You will learn how to prepare the environment by downloading and verifying the ISO image, configuring and launching the virtual machine, and following the steps to install Debian without a graphical interface. The goal is to set up a minimal installation of Debian 12 that will serve as a foundation for installing additional services like Apache, PostgreSQL, and PHP.

Preparing the Installation

Before starting the installation, download the ISO image from here:

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

If the ISO image is already installed, compare the image to the link above. To do this, run the command:

```
$ sha512sum /usr/local/images-ISO/debian-12.5.0-amd64-netinst.iso
```

Debian System Installation

Installation

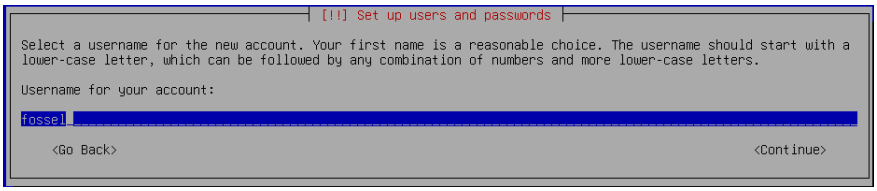
To start the virtual machine installation, run the command

```
$ S2.03-lance-installation
```

Proceed through the installation steps, and when nothing is specified, choose the default option.Language : English

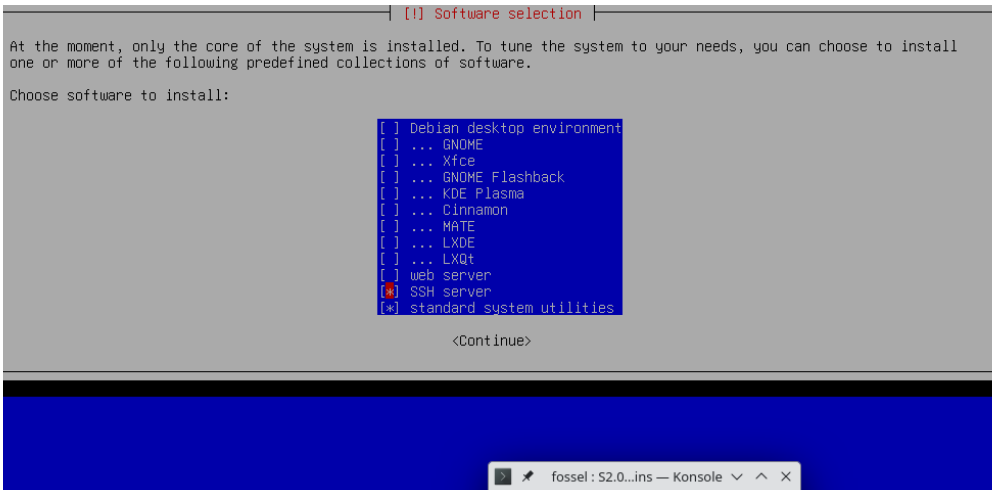
- **Location** : other/Europe/Franfe
- **Locales** : United States, en_US.UTF-8
- **Keyboard** : French
- **Hostname** : server-<your_uga_login>
- **Root Password** : root
- **User account** : Full name
- **User Name :** <your_uga_login>

Example :



- **Password** : etu
- **Partition disks** : Guided – use entire disk
- **Partition disks :** All files in one partition
- **Partition disks** : Yes

- **Software Selection** : Ensure ssh server is selected and Debian desktop is not selected



- **Install GRUB** : Yes
- **Device :** /dev/sda

Once the installation is complete, shut down the virtual machine before proceeding to the next step. To do this, as the root user, run the following command in your virtual machine's shell:

```
# poweroff
```

Moving the Disk Image

You have created the image on the local disk of the Linux workstation. To move it to the erebus4 server, first ensure that the virtual machine is properly shut down and then run the command

```
$ S2.03-déplace-image-disque-sur-erebus4.
```

Verifying the Debian Server

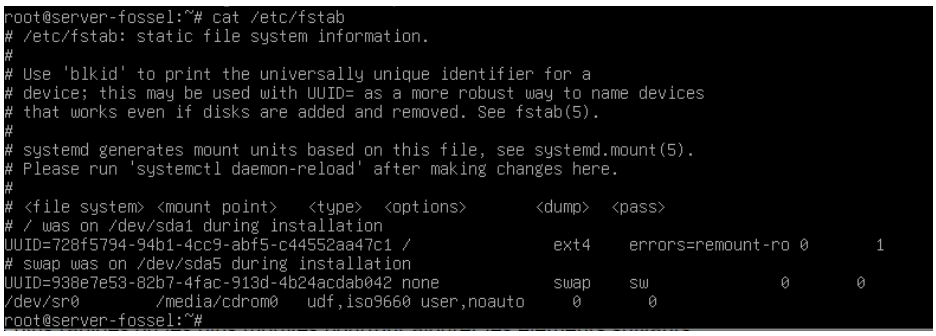
To start your virtual machine, run

```
S2.03-lance-machine-virtuelle
```

in your host shell.

Check /etc/fstab

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



Verifying Network Configuration

Check the network configuration of your virtual machine and ensure you can reach the outside using the command

```
ip addr
```

Verifying Absence of the Xorg Server

To verify that the Xorg server is not installed, run the command:

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

Port Forwarding and SSH Access

To allow access to servers running on your virtual machine from clients running on your workstation, the following port forwarding rules are set up (script S2.03-commun)

Network device	VM Port	Port on Linux station	Example of use from the Linux station
SSH	22	2222	$ ssh toto@localhost -p 2222
HTTP	80	8080	URL: http://localhost:8080/
HTTPS	443	4443	URL: https://localhost:4443/
PostgreSQL	5432	5432	$ psql -h localhost -U postgres postgres

Installation of Apache, PostgreSQL and PHP

This section details the process of installing Apache, PostgreSQL, and PHP on a Debian 12 server environment. You will learn how to set up and configure these essential components to create a robust web hosting and database management system.

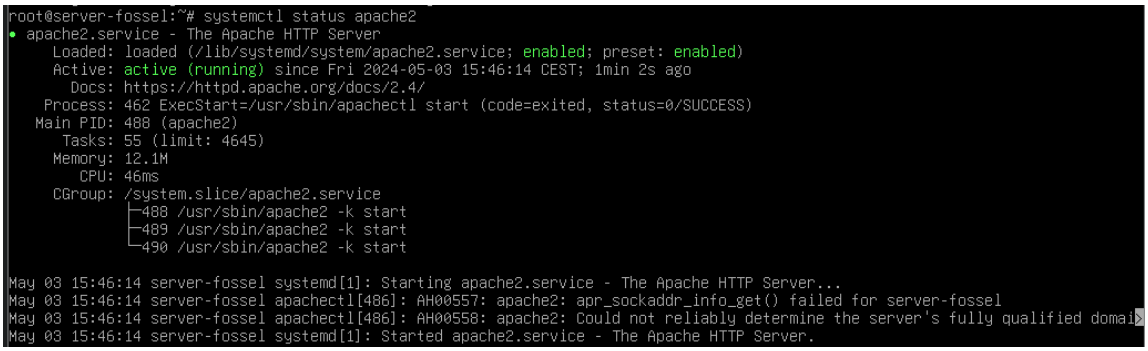
Installation Apache

To install Apache, execute the following commands to update the packages and install Apache:

```
# apt update
# apt install apache2
```

Verify that Apache is started:

```
# systemctl start apache2
```



If Apache is not running, start the service :

```
# systemctl start apache2
```

Vérification de l'installation

Execute :

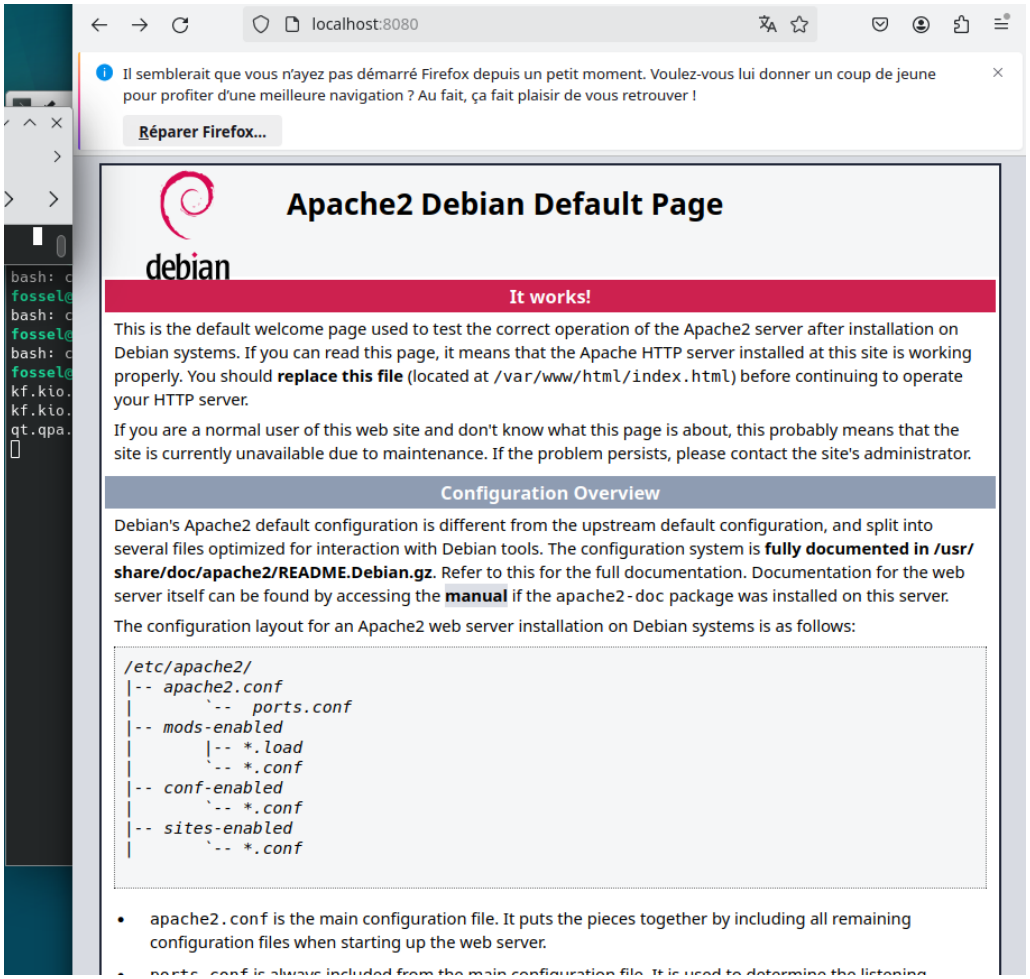
```
$ telnet localhost80
```

```
$ telnet localhost 80
Trying ::1...
Connected to localhost.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
[...]
```

Afterwards, enter "HEAD / HTTP/1.0" followed by two line breaks.

Finally, on your host machine, visit the page http://localhost:8080 and verify that you are redirected to the Apache server page of the virtual machine, as follows:



Installation of PostgresSQL

Installation du programme

To begin, obtain a root shell by executing

```
sudo -i
```

Perform security updates:

```
# apt update
# apt upgrade
# apt clean
```

Install PostgreSQL using the command :

```
# apt install postgresql
```

Configuring the server to accept external connections

Switch to the postgres account:

```
# su – postgres
```

Connect to the PostgreSQL server:

```
psql
```

To allow external connections, edit the configuration file:

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

Uncomment the line :

```
listen_adresses = '*'
```

Next, edit the authentification rules file :

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

Add the following rule to allow connections authenticated by a hashed password :

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

Afterwards, switch back to rool (logout) and restard your server

```
# service postgresql restart
```

Créations to test your server

Once connected to your database (using the psql command) :

Create a user with your login name :

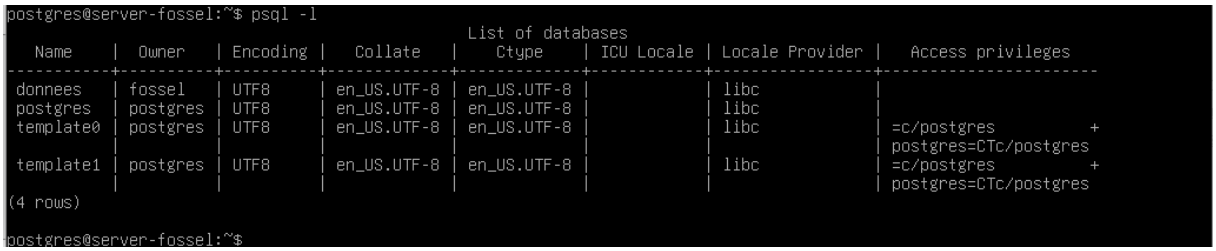
```
CREATE USER <votre_login> WITH PASSWORD 'password' ;
```

Next, create a database where your user is the ower :

```
CREATE DATABASE donnees OWNER <votre_login>;
```

Check the existence of this database with the command

```
psql -l
```



Connect to your database :

```
\c data
```

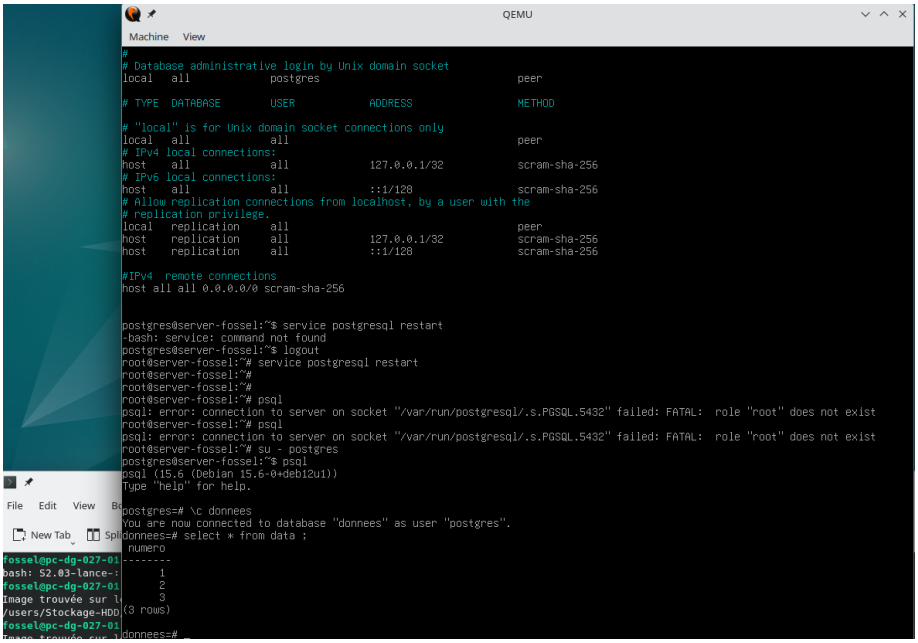
Create a table in this database :

```
CREATE TABLE data (numero int);
```

Insert some data into the table :

```
INSERT INTO data VALUES (‘1’);
INSERT INTO data VALUES (‘2’);
INSERT INTO data VALUES (‘3’);
```

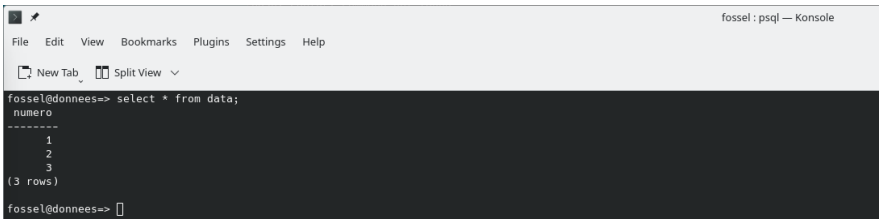
Query your table from your virtual machine :



Connect to your database from the host machine

```
$ psql -h localhost donnees -U fossel
```

Query your table from your host machine :



To verify that the passwords are properly hashed

To verify that the passwords are hashed correctly, from your virtual machine, use the command:

```
SELECT usename, passwd FROM pg_shadow ;
```

```
postgres=# select usename, passwd from pg_shadow;
 usename  |                                                            passwd
----------+-----------------------------------------------------------------------------------------------------------------
-----------------
 postgres |
 fossel   | SCRAM-SHA-256$4096:gUhnfdHEghEm2sK2TdwUng==$KSY8VJyly7/wg5AWlRoGQHfdsF5yyh01ErfklxUiePw=:SRB1B1CIT293kbX6Lcg+UrvH6Gl
k5VeRRL3pOYHv0/s=
(2 rows)
```

Installation of PHP

To install PHP on your virtual machine, in the root shell, enter the following command:

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

Test PHP Installation

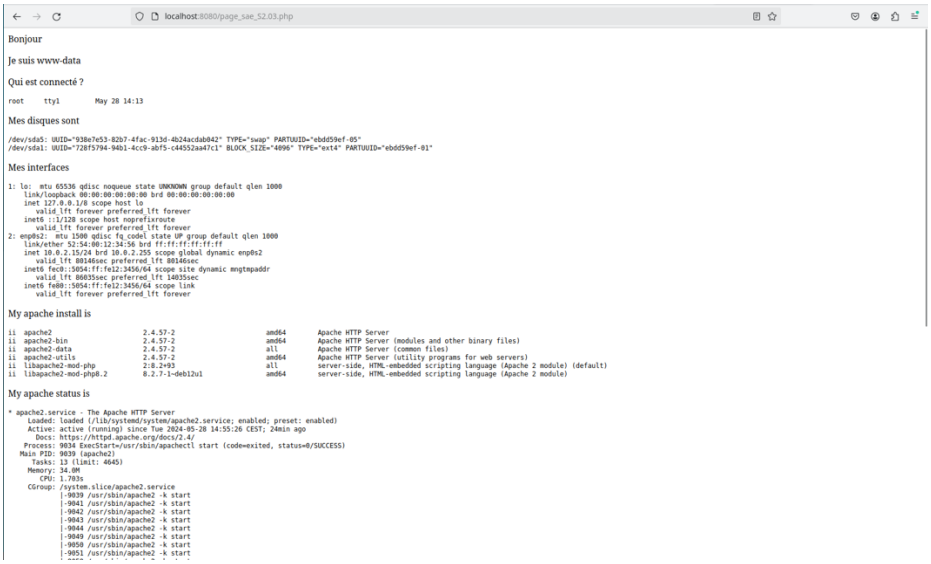
Create the file info.php :
nano /var/www/html/info.php
And place the following code in this file :

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

Open a web browser and navigate to http://localhost/8080.info.php

```
# /sbin/blkid
# scp user@pc-dg-xxx-yy :
/users/info/www/intranet/enseignements/S2.03/page_sae_S2.03.php /var/www/html
```

Open a web browser and navigate to http://localhost:8080/page_sae_S2.03.php



Installation of phpPgAdmin

To install phpPgAdmin, execute the following command :

```
# apt install phppgadmin
```

Configuration

Execute

```
# nano /usr/share/phppgadmin/classes/database/Connection.php
```

Remplace the line with :

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

To :

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

Edit ‘phppgadmin.conf’ to allow conection :

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

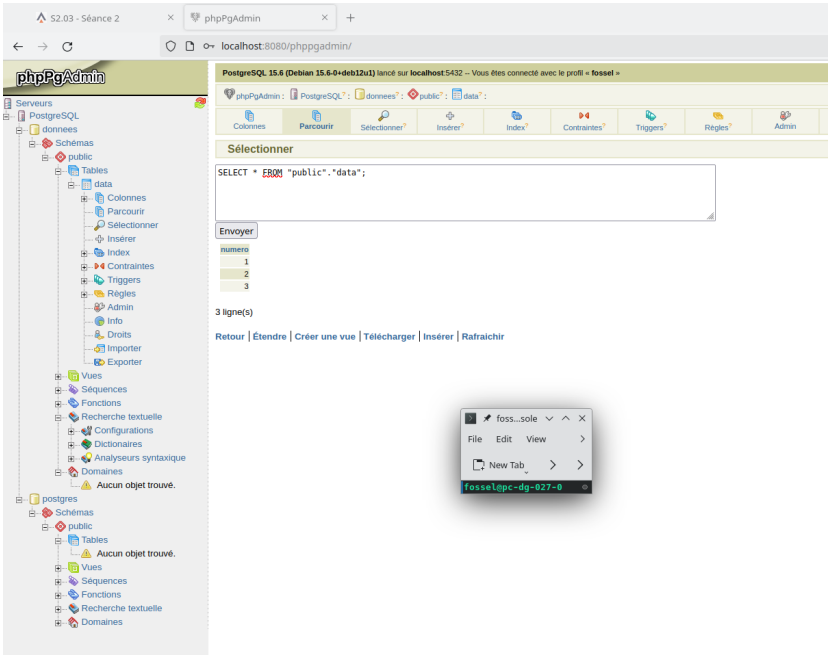
Replace :

```
Require all local
```

To :

```
Require all granted
```

Open a web browser and navigate to http://localhost:8080/phppgadmin

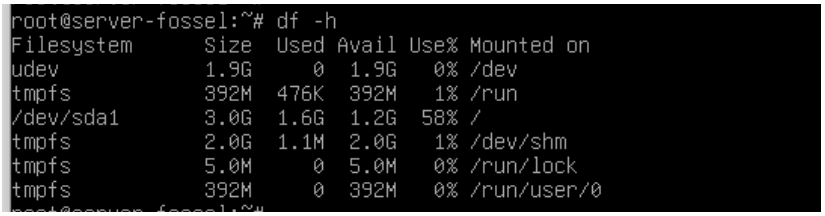


Security Analysis

Disk space

To check the remaining disk space, use the command :

```
$ df -h
```



Security recommendations

Regulary ypdate packages :

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

Conclusion

By following this guide, you will have a fully functional Debian 12 server equipped with Apache, PostgreSQL, and PHP. This setup will be accessible from the host machine, providing a robust environment for web development and database management.