

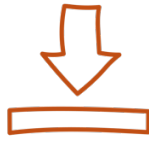
Installation Guide Debian 12 Server with Apache, PostgreSQL and PHP

Author
Lucas FOSSE



Table des matières

Debian 12 Server Installation	3
Preparing the Installation	4
Debian System Installation.....	4
Installation	4
Moving the Disk Image.....	5
Verifying the Debian Server	5
Check /etc/fstab	6
Installation Apache, PostgreSQL et PHP	<i>Erreur ! Signet non défini.</i>
Installation Apache	8
Vérification de l'installation	8
Installation PostgreSQL	9
Installation du programme.....	9
Configuration du serveur pour accepter des connexions extérieures	Erreur ! Signet non défini.
Créations pour tester votre serveur	Erreur ! Signet non défini.
Vérification que les mots de passe sont bien hachés	Erreur ! Signet non défini.
Installation PHP	12
Test de l'installation PHP	12
Installation PhpPgAdmin	13
Configuration	13
Security Analysis.....	15
Espace disque	Erreur ! Signet non défini.
Security recommendations	15
Conclusion	<i>Erreur ! Signet non défini.</i>



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

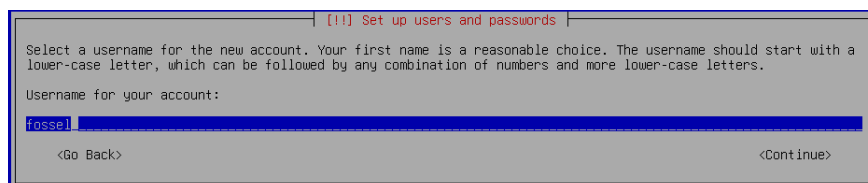
To start the virtual machine installation, run the command

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

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

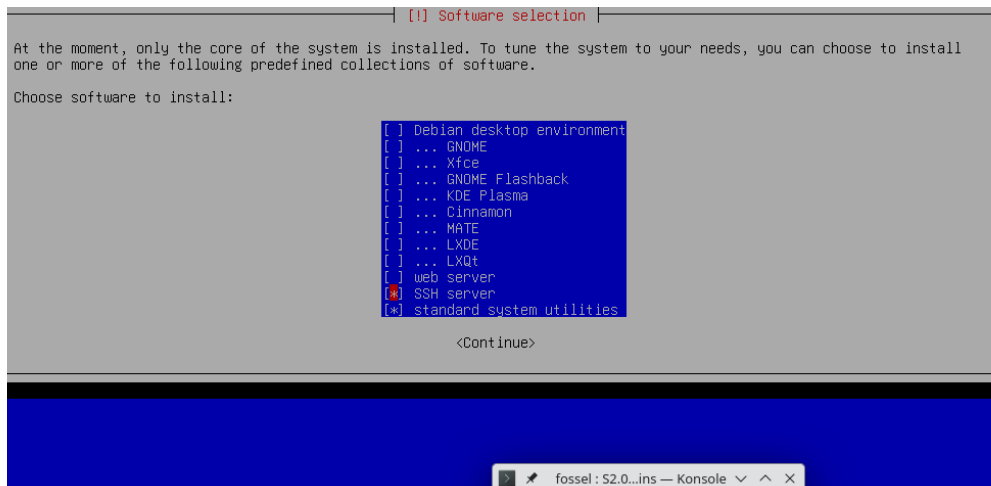
- **Location** : other/Europe/Franfe
- **Locales** : United States, en_US.UTF-8
- **Keyboard** : French
- **Hostname** : server-<your_uga_login>
- **Root Password** : root
- **User account** : Full name
- **User Name** : <your_uga_login>

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
```

```
root@server-fossil:~# 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=728f5794-94b1-4cc9-abf5-c44552aa47c1 / ext4 errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=938e7e53-82b7-4fac-913d-4b24acdab042 none swap sw 0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
root@server-fossil:~#
```

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

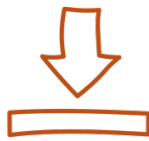
To verify that the Xorg server is not installed, run the command:

```
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
```

```
root@server-fossil:~# 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 15:46:14 CEST; 1min 2s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 462 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 488 (apache2)
    Tasks: 55 (limit: 4645)
   Memory: 12.1M
      CPU: 46ms
   CGroup: /system.slice/apache2.service
           └─488 /usr/sbin/apache2 -k start
             └─489 /usr/sbin/apache2 -k start
               └─490 /usr/sbin/apache2 -k start

May 03 15:46:14 server-fossil systemd[1]: Starting apache2.service - The Apache HTTP Server...
May 03 15:46:14 server-fossil apachectl[486]: AH00557: apache2: apr_sockaddr_info_get() failed for server-fossil
May 03 15:46:14 server-fossil apachectl[486]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, please
May 03 15:46:14 server-fossil systemd[1]: Started apache2.service - The Apache HTTP Server.
```

If Apache is not running, start the service :

```
# systemctl start apache2
```

Vérification de l'installation

Execute :

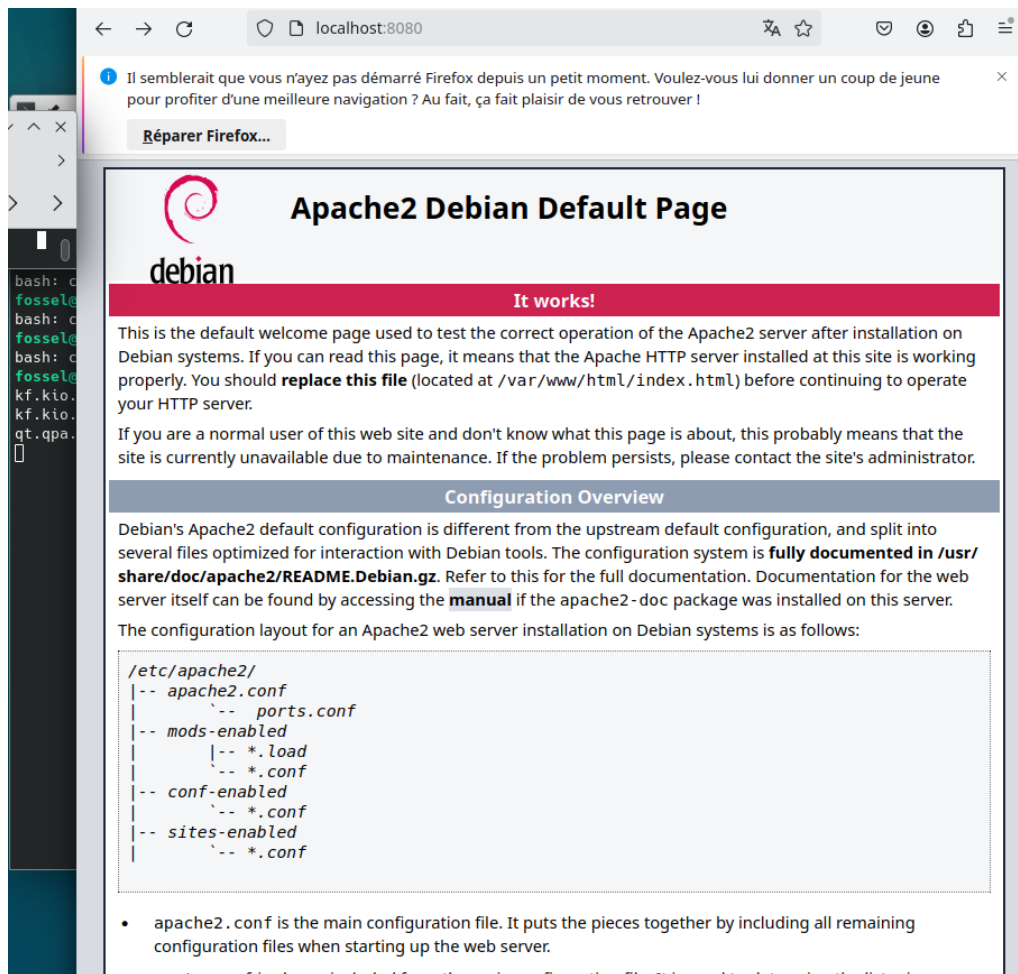
```
$ telnet localhost80
```

```
$ 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 PostgreSQL

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_addresses = '*'
```

Next, edit the authentication 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 root (logout) and restart 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 owner :

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

Check the existence of this database with the command

```
psql -l
```

```
postgres@server-fossel:~$ psql -l
                                List of databases
  Name  | Owner  | Encoding | Collate | Ctype  | ICU Locale | Locale Provider | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----
 donnees | fossel | 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-fossel:~$
```

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 :

The screenshot shows a QEMU virtual machine window with a terminal. The terminal displays the PostgreSQL configuration file content, followed by commands to restart the service, switch to the postgres user, and run a query. The query result shows three rows of data.

```
# Database administrative login by Unix domain socket  
local all postgres peer  
  
# TYPE DATABASE USER ADDRESS METHOD  
  
# "local" is for Unix domain socket connections only  
local all all peer  
# IPv4 local connections:  
host all all 127.0.0.1/32 scram-sha-256  
# IPv6 local connections:  
host all all ::1/128 scram-sha-256  
# Allow replication connections from localhost, by a user with the  
# replication privilege.  
local replication all peer  
host replication all 127.0.0.1/32 scram-sha-256  
host replication all ::1/128 scram-sha-256  
  
#IPv4 remote connections  
host all all 0.0.0.0/0 scram-sha-256  
  
postgres@server-fossil:~$ service postgresql restart  
./bash: service: command not found  
postgres@server-fossil:~$ logout  
root@server-fossil:~$ service postgresql restart  
root@server-fossil:~$  
root@server-fossil:~$ psql  
psql: error: connection to server on socket "/var/run/postgresql/.s.PGSQL.5432" failed: FATAL: role "root" does not exist  
root@server-fossil:~$ psql  
psql: error: connection to server on socket "/var/run/postgresql/.s.PGSQL.5432" failed: FATAL: role "root" does not exist  
root@server-fossil:~$ su - postgres  
postgres@server-fossil:~$ psql  
psql (15.6 (Debian 15.6-0+deb12u1))  
Type "help" for help.  
  
postgres=# \c donnees  
You are now connected to database "donnees" as user "postgres".  
postgres=# select * from data ;  
numero  
-----  
1  
2  
3  
(3 rows)
```

Connect to your database from the host machine

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

Query your table from your host machine :

The screenshot shows a host machine terminal window titled 'fossil : psql - Konsole'. It displays the command to connect to the database and the query result showing three rows of data.

```
fossil@donnees=> select * from data;  
numero  
-----  
1  
2  
3  
(3 rows)
```

To verify that the passwords are properly hashed

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

```
SELECT username, passwd FROM pg_shadow ;
```

```
postgres=# select username, passwd from pg_shadow;
username |                                passwd
-----+-----
postgres |
fossil   | SCRAM-SHA-256$4096:gUhnfdHEghEm2sK2TdwUng==$KSY8VJyly7/wg5AWlRoGQHfdsF5yyh01ErflxUiePw=:SRB1B10IT293kbX6Lcg+UrvH6G1
x5VeRRL3p0YHv0/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

```
localhost:8080/page_sae_S2.03.php
Bonjour
Je suis www-data
Qui est connecté ?
root tty1 May 28 14:13
Mes disques sont
/dev/sda5: UUID="938c7e53-82b7-4fac-9136-8024acdab8a2" TYPE="xfs" PARTUUID="ebdd59ef-05"
/dev/sda1: UUID="728f5794-94b1-4c9b-a0f5-c4552aa7c1c1" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="ebdd59ef-01"
Mes interfaces
1: lo: mtu 65536 uid 0 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: enp0s3: mtu 1500 uid 0 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 enp0s3
    valid_lft 86400sec preferred_lft 86400sec
inet6 fe80::1004:ff:fe22:3456/64 scope site dynamic mngtnp0s3
    valid_lft 86400sec preferred_lft 14400sec
inet6 fe80::1004:ff:fe22:3456/64 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 2.0.2+93 all server-side, HTML embedded scripting language (Apache 2 module) (default)
ii libapache2-mod-php8.2 8.2.7-1+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 Tue 2024-05-28 14:55:26 CEST; 24min ago
Docs: https://httpd.apache.org/docs/2.4/
Process: 9838 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
Main PID: 9839 (apache2)
Tasks: 13 (limit: 4640)
Memory: 34.0M
CPU: 1.78s
Group: /system.slice/apache2.service
-9839 /usr/sbin/apache2 -k start
-9841 /usr/sbin/apache2 -k start
-9842 /usr/sbin/apache2 -k start
-9843 /usr/sbin/apache2 -k start
-9844 /usr/sbin/apache2 -k start
-9849 /usr/sbin/apache2 -k start
-9850 /usr/sbin/apache2 -k start
-9851 /usr/sbin/apache2 -k start
```

Installation of phpPgAdmin

To install phpPgAdmin, execute the following command :

```
# apt install phppgadmin
```

Configuration

Execute

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

Replace the line with :

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

To :

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

Edit 'phppgadmin.conf' to allow conection :

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

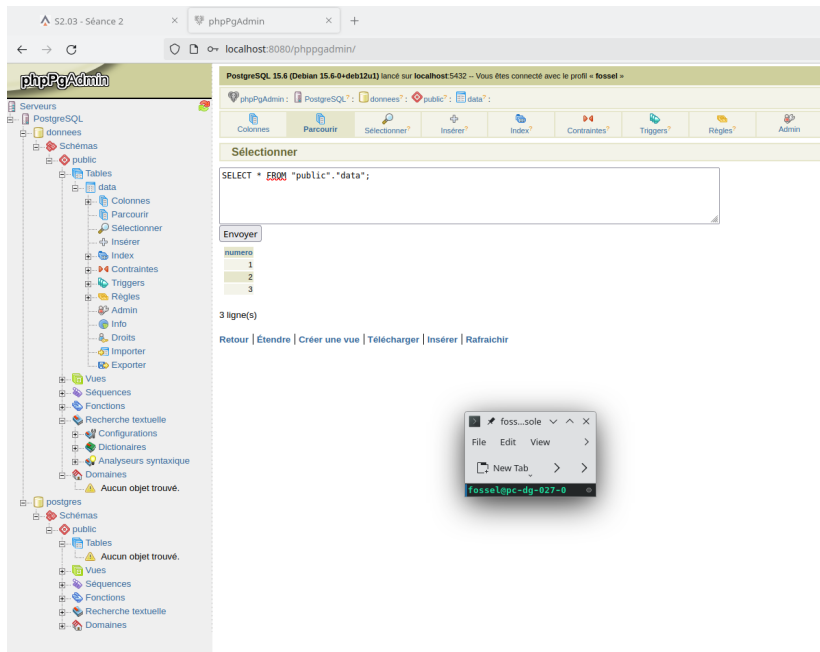
Replace :

```
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
```

```
root@server-fossil:~# 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  1.2G  58% /
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/0
root@server-fossil:~#
```

Security recommendations

Regularly update 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.