

DEBIAN SERVER INSTALLATION HANDBOOK

Project team

- BARNOUIN Clément



Contents

Project team.....	0
Contents.....	0
Introduction.....	1
Prerequisites.....	1
Commands.....	2
Basic commands.....	2
System management commands.....	2
File management commands.....	2
Package management commands.....	3
Service management commands.....	3
Network commands.....	3
PostgreSQL commands.....	4
Apache Commands.....	4
PHP Commands.....	4
SSH key management commands.....	4
Shortcuts.....	4
Notations.....	4
Installation.....	5
Step 1: Setting up the Installation.....	5
1. Download the Debian 12 ISO image.....	5
2. Check image integrity.....	5
3.1. [University Condition] Create a new virtual machine using Qemu/KVM.....	5
3.2. [Real Condition] Create a new virtual machine using Qemu/KVM.....	5
Step 2: Install Debian 12.....	5
1. Boot from the ISO and start the installation.....	5
2. Post-installation commands.....	6
Step 3: Move Disk Image.....	6
1.1. [University Condition] Move the disk image to the server.....	6
1.2. [Real Condition] Move the disk image to the server.....	6
Step 4: Verify the Server Installation.....	7
1.1. [University Condition] Launch the virtual machine.....	7
1.2. [Real Condition] Launch the virtual machine.....	7
2. Check partition mounting.....	7

3. Check network configuration and ensure connectivity.....	7
Step 5: Install Apache.....	8
1. Install Apache.....	8
2. Verify Apache installation.....	8
3. Access Apache from the host.....	8
Step 6: Install PostgreSQL.....	9
1. Install PostgreSQL.....	9
2. Verify PostgreSQL installation.....	9
3. Test PostgreSQL.....	9
4. Configure PostgreSQL for accessibility from the host station.....	9
5. Query the database.....	10
6. Checking password hashes.....	10
Step 7: Install PHP.....	11
1. Install PHP.....	11
2. Test PHP installation.....	11
3. Access the PHP info page from the host.....	11
4. Access to another page.....	11
Step 8: Install PhpPgAdmin.....	12
1. Install PhpPgAdmin.....	12
2. Configure PhpPgAdmin.....	12
3. Access PhpPgAdmin from the host.....	12
Step 9: Security Analysis.....	13
1. Final storage space.....	13
2. Identify and fix potential security vulnerabilities.....	13
Optional Enhancements.....	13
1. SSH Access with Key Authentication.....	13
Method 1 – Accessing the root account via SSH.....	13
Method 2 – Accessing the SSH Server.....	14
2. Enable HTTPS on Apache:.....	14
Conclusion.....	14

Introduction

This guide will provide step-by-step instructions for installing a Debian 12 server with Apache, PostgreSQL, and PHP. This installation will be performed on a Qemu/KVM virtual machine and should be accessible and manageable from the host machine.

Prerequisites

1. Qemu/KVM installed on the host machine
2. Debian 12 ISO image (netinst)
3. Basic knowledge of Linux commands and system administration

Commands

Basic commands

cd [directory] : Changes the current directory.

Example : \$ cd /var/www/html

ls [options] [directory] : Lists the contents of a directory.

Example : \$ ls -l

cp [source] [destination] : Copying files or directories.

Example : \$ cp file.txt /home/user/

mv [source] [destination] : Moves or renames files or directories.

Example : \$ mv fichier.txt /home/user/new_name.txt

rm [options] [file/directory] : Delete files or directories.

Example : \$ rm file.txt

System management commands

su - root: Login as root from user.

logout: Disconnect.

shutdown -h now: Shuts down the system immediately.

reboot: Restarts the system.

poweroff: Turns off the system.

File management commands

nano [file]: Command line text editor.

Example : \$ nano /etc/apache2/apache2.conf

cat [file]: Displays the contents of a file.

Example : \$ cat /etc/passwd

find [directory] -name [filename]: Finds files in a directory.

Example : \$ find / -name "file.txt"

grep [pattern] [file]: Searches for a pattern in a file.

Example : \$ grep "root" /etc/passwd

Package management commands

apt-get update: Updates the list of available packages.

Example : # apt-get update apache2

apt-get upgrade: Updates installed packages.

Example : # apt-get upgrade apache2

apt-get install [package]: Installs a package.

Example : # apt-get install apache2

apt-get remove [package]: Removes a package.

Example : # apt-get remove apache2

apt-get clean: Cleans installation packages.

Service management commands

systemctl status [service]: Displays the status of a service.

Example : # systemctl status apache2

systemctl start [service]: Starts a service.

Example : # systemctl start apache2

systemctl stop [service]: Stops a service.

Example : # systemctl stop apache2

systemctl restart [service]: Restarts a service.

Example : # systemctl restart apache2

systemctl enable [service]: Enables a service to start at system startup.

Example : # systemctl enable apache2

systemctl disable [service]: Disables a service so that it does not start at system startup.

Example : # systemctl disable apache2

Network commands

ip addr: Displays IP address and network configuration information.

Example : \$ ip addr

ssh [user]@[ip_address] -p [port]: Connects to a remote machine via SSH.

Example : \$ ssh user@192.168.1.10 -p 22

scp [source] [user]@[ip_address]:[destination]: Copies files to a remote machine.

Example : \$ scp fichier.txt user@192.168.1.10:/home/user/

PostgreSQL commands

su - postgres: Changes the current user to PostgreSQL user.

psql -l: List PostgreSQL databases.

psql -d [database_name]: Connects to a PostgreSQL database.

psql -h: Connects to a PostgreSQL database on a remote host by specifying the user.

Example : \$ psql -h localhost -U postgres mydatabase

Apache Commands

apachectl configtest: Tests Apache configuration for errors.

systemctl reload apache2: Reloads the Apache configuration without restarting the service.

Example : # systemctl reload apache2

PHP Commands

php -v: Shows the installed PHP version.

Example : \$ php -v

SSH key management commands

ssh-keygen: Generates a new SSH key pair.

Example : \$ ssh-keygen

ssh-add [private_key_path]: Adds a private key to the SSH agent.

Example : \$ ssh-add ~/.ssh/id_rsa

ssh user@ip_address: SSH connection.

Shortcuts

CTRL + C: Stop a process in a terminal.

CTRL + L: Clean terminal.

CTRL + W: Search in nano software.

CTRL + O: Save in nano software.

CTRL + X: Exit nano software.

Notations

Commands marked with a “\$” at the beginning can be executed in a user's command terminal.

Commands with “#” are only executed in a root terminal. The commands highlighted in **blue** are to be performed on the host machine and in **gray**, on the virtual machine.

Installation

Step 1: Setting up the Installation

1. Download the Debian 12 ISO image

The ISO image can be downloaded from Debian's official site :

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

Install Debian version 12.x "bookworm" pour processeurs x86 64 bits avec l'image ISO de type "netinst".

2. Check image integrity

Visually compare the two prints by performing the command:

```
$ sha512 FILE_NAME
```

3.1. [University Condition] Create a new virtual machine using Qemu/KVM

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

3.2. [Real Condition] Create a new virtual machine using Qemu/KVM

Only if a Qemu/KVM machine is not yet initialized.

```
# qemu-system-x86_64 -hda debian12.img -cdrom debian-12-netinst.iso -boot d -m 1024
```

Step 2: Install Debian 12

1. Boot from the ISO and start the installation

- Language : English
- Location : Other/Europe/France
- Locales : United States, en_US.UTF-8
- Keyboard : French
- Hostname : `server-YOUR_UGA_LOGIN`
- Root Password : `root`
- User Account : Full Name: `Your Name`, Username: `YOUR_UGA_LOGIN`, Password: `etu`
- Partition Disks : Guided - use entire disk, All files in one partition
- Software Selection : Uncheck 'Debian desktop', check 'ssh server'
- Install GRUB : Yes
- Device for boot loader : /dev/sda

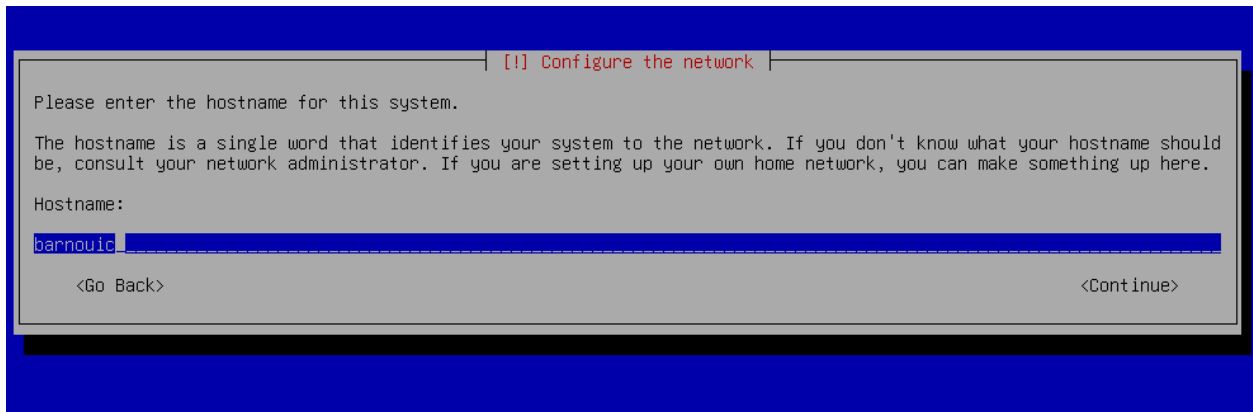


Figure 1. Entering the user login name.

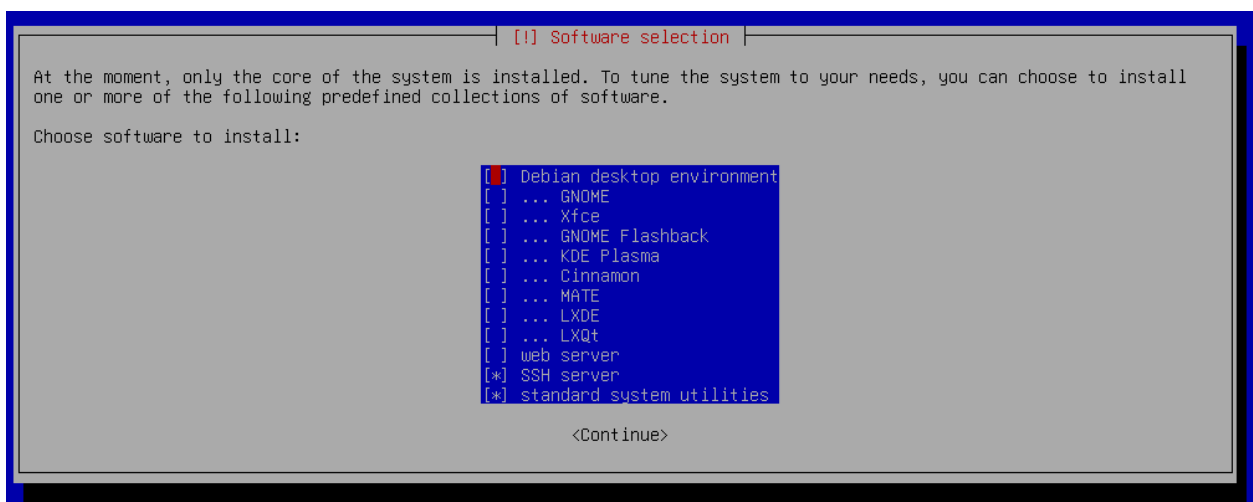


Figure 2. Software selection during installation.

2. Post-installation commands

After the first reboot, log in as root and run :

```
# poweroff
```

Step 3: Move Disk Image

1.1. [University Condition] Move the disk image to the server

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

1.2. [Real Condition] Move the disk image to the server

```
# scp debian12.img user@server:/path/to/destination
```

Step 4: Verify the Server Installation

1.1. [University Condition] Launch the virtual machine

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

1.2. [Real Condition] Launch the virtual machine

```
# qemu-system-x86_64 -hda debian12.img -m 1024
```

2. Check partition mounting

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

```
oot@barnouic:~# 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=c18fa667-374b-4c73-8e22-7c78327fdf7d / ext4 errors=remount-ro 0 1
swap was on /dev/sda5 during installation
UUID=fa75edd6-44a0-44e9-92d5-b75eb900f7b9 none swap sw 0 0
dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
oot@barnouic:~# _
```

Figure 3. Result of the `cat /etc/fstab/` command to verify the installation of the installation.

3. Check network configuration and ensure connectivity

Take note of the IPv4, IPv6 and MAC addresses.

```
$ ip addr
```

Check the absence of the Xorg server.

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

Port forwarding table:

Network service	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

Step 5: Install Apache

1. Install Apache

```
# apt-get install apache2
```

2. Verify Apache installation

```
# systemctl status apache2
```

```
09barnouic:~# systemctl status apache2
apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-04-09 17:04:14 CEST; 1min 38s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 988 (apache2)
    Tasks: 55 (limit: 4645)
   Memory: 9.3M
      CPU: 60ms
   CGroup: /system.slice/apache2.service
           └─988 /usr/sbin/apache2 -k start
             990 /usr/sbin/apache2 -k start
             991 /usr/sbin/apache2 -k start

Apr 09 17:04:14 barnouic systemd[1]: Starting apache2.service - The Apache HTTP Server...
Apr 09 17:04:14 barnouic systemd[1]: Started apache2.service - The Apache HTTP Server.
09barnouic:~# _
```

Figure 4. Verifying that Apache2 is working.

If Apache is not started :

```
# systemctl start apache2
```

3. Access Apache from the host

```
$ telnet localhost 80
```

Write “HEAD / HTTP/1.0 then two newlines. The server should respond “HTTP/1.1 200 OK”

Open a web browser and navigate to `http://localhost:8080` sur la machine hôte.

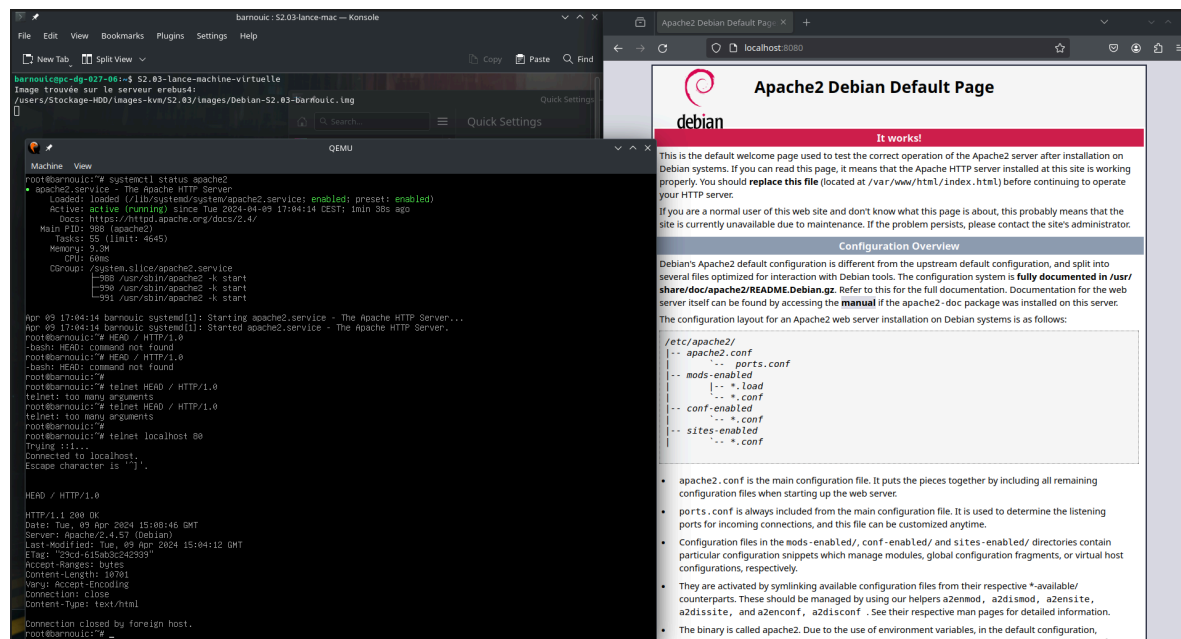


Figure 5. Web page on the host machine, hosted on the virtual machine using Apache2.

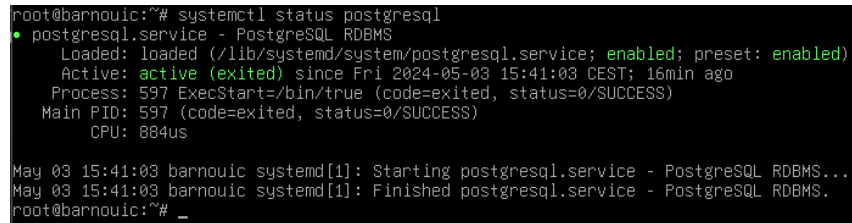
Step 6: Install PostgreSQL

1. Install PostgreSQL

```
# apt-get install postgresql
```

2. Verify PostgreSQL installation

```
# systemctl status postgresql
```



```
root@barnouic:~# systemctl status postgresql
• postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; preset: enabled)
   Active: active (exited) since Fri 2024-05-03 15:41:03 CEST; 16min ago
   Process: 597 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 597 (code=exited, status=0/SUCCESS)
   CPU: 884us

May 03 15:41:03 barnouic systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
May 03 15:41:03 barnouic systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
root@barnouic:~# _
```

Figure 6. Verifying the operation of PostgreSQL.

3. Test PostgreSQL

Connect to PostgreSQL with login postgres and list current databases :

```
# su - postgres
```

```
$ psql -l
```

Create a new user with your UGA login as the name, then create a new database whose owner is your user with a new table in the database containing several lines.

```
$ createuser barnouic
```

```
CREATE USER barnouic WITH CREATEROLE PASSWORD 'etu';
```

```
ALTER USER barnouic WITH CREATEROLE PASSWORD 'etu';
```

```
DROP USER barnouic;
```

```
GRANT INSERT, UPDATE, SELECT ON TABLE identifiant_etu TO barnouic;
```

```
$ createdb universite -O barnouic
```

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

```
INSERT INTO identifiant_etu (nom, age) VALUES ('Clement', 19), ('Loucas', 18);
```

4. Configure PostgreSQL for accessibility from the host station

- Edit `/etc/postgresql/15/main/postgresql.conf` :

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

Edit to : listen_addresses = '*'

- Edit `/etc/postgresql/15/main/pg_hba.conf` :

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

Edit to : host all all 0.0.0.0/0 scram-sha-256

```
# sytemctl restart postgresql
```

5. Query the database

```
$ psql -d universite
```

```
SELECT * FROM identifiant_etu;
```

```
universite=> SELECT * FROM identifiant_etu;
 id |  nom  | age
-----+-----+----
  1 | Clement | 19
  2 | Loucas | 18
(2 rows)

universite=>
```

Figure 7. Querying the university database from the virtual machine.

```
$ psql -h posgres-info universite -U barnouic
```

```
SELECT * FROM identifiant_etu;
```

```
barnouic@universite=> SELECT * FROM identifiant_etu;
 id |  nom  | age
-----+-----+----
  1 | Clement | 19
  2 | Loucas | 18
(2 rows)

barnouic@universite=>
```

Figure 8. Querying the university database from the host machine.

```
$ psql -l
```

```
barnouic@barnouic:~$ psql -l
```

Name	Owner	Encoding	Collate	Ctype	ICU Locale	Locale Provider	Access privileges
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 +
template1	postgres	UTF8	en_US.UTF-8	en_US.UTF-8		libc	=c/postgres +
universite	barnouic	UTF8	en_US.UTF-8	en_US.UTF-8		libc	postgres=Ctc/postgres

(4 rows)

Figure 9. List of PostgreSQL database owners from the virtual machine.

6. Checking password hashes

```
# su - postgres
```

```
SELECT * FROM pg_shadow;
```

```
postgres=# SELECT * FROM pg_shadow;
username | usesysid | usecreatedb | usesuper | use repl | usebypassrls |          | valuntil | useconfig
-----+-----+-----+-----+-----+-----+-----+-----+-----
postgres |      10 | t           | t         | t         | t           |          |          |
barnouic |   16384 | f           | f         | f         | f           | SCRAM-SHA-256$4096:5tpyJM90JVt+08KmCxdJbQ==$cURNu2ipGO+cs0vJLRtz6Bzs3L7Suo2jaCjV4cFJx4k=:db1QL40Gs8TEMrRIn10BC7mVBvAx9bvcmCMVs1aLOA= |
(2 rows)

(END)
```

Figure 10. Querying the pg_shadow table from the virtual machine.

Step 7: Install PHP

1. Install PHP

```
# apt-get install php
```

2. Test PHP installation

- Create a `info.php` file in `/var/www/html/` :

```
# nano /var/www/html/info.php
```

Write down :

```
<?php
phpinfo();

phpinfo(INFO_MODULES);

?>
```

3. Access the PHP info page from the host

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

4. Access to another page

```
# /sbin/blkid
```

```
# scp
```

barnouic@pc-dg-027-10:/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`.

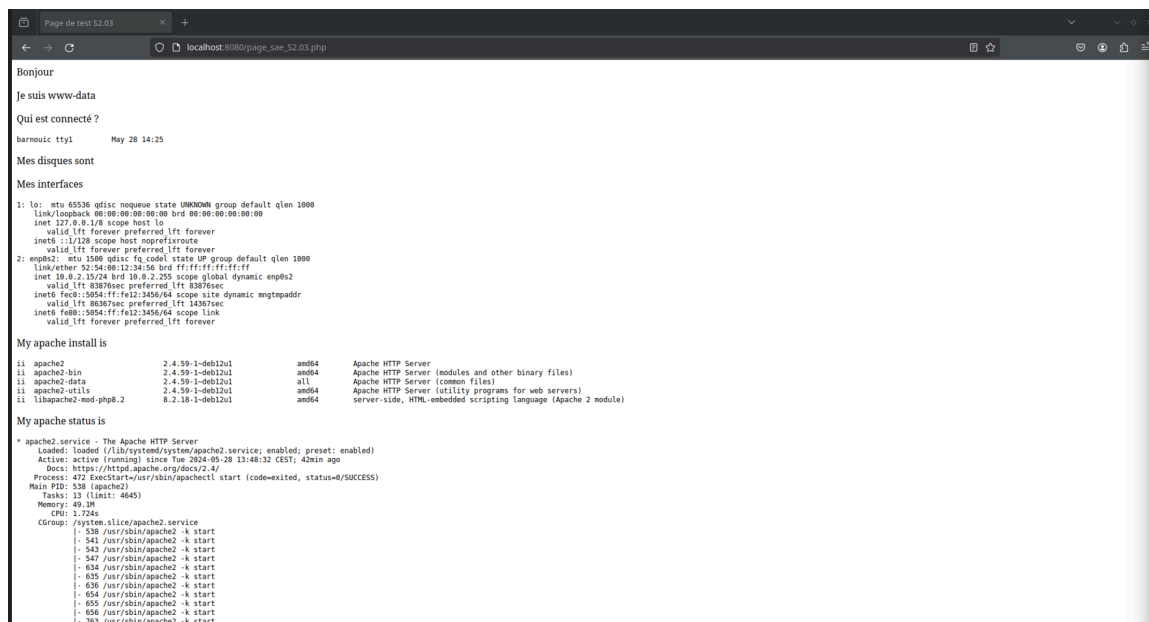


Figure 10. Navigation on the page_sae_S2.03.php from the host machine using apache2.

Step 8: Install PhpPgAdmin

1. Install PhpPgAdmin

```
# apt-get install phppgadmin
```

2. Configure PhpPgAdmin

- Edit `Connection.php` to ensure compatibility :

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

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

Edit :

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

To :

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

- Edit `phppgadmin.conf` to allow connection :

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

Write down :

Require all granted

3. Access PhpPgAdmin from the host

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

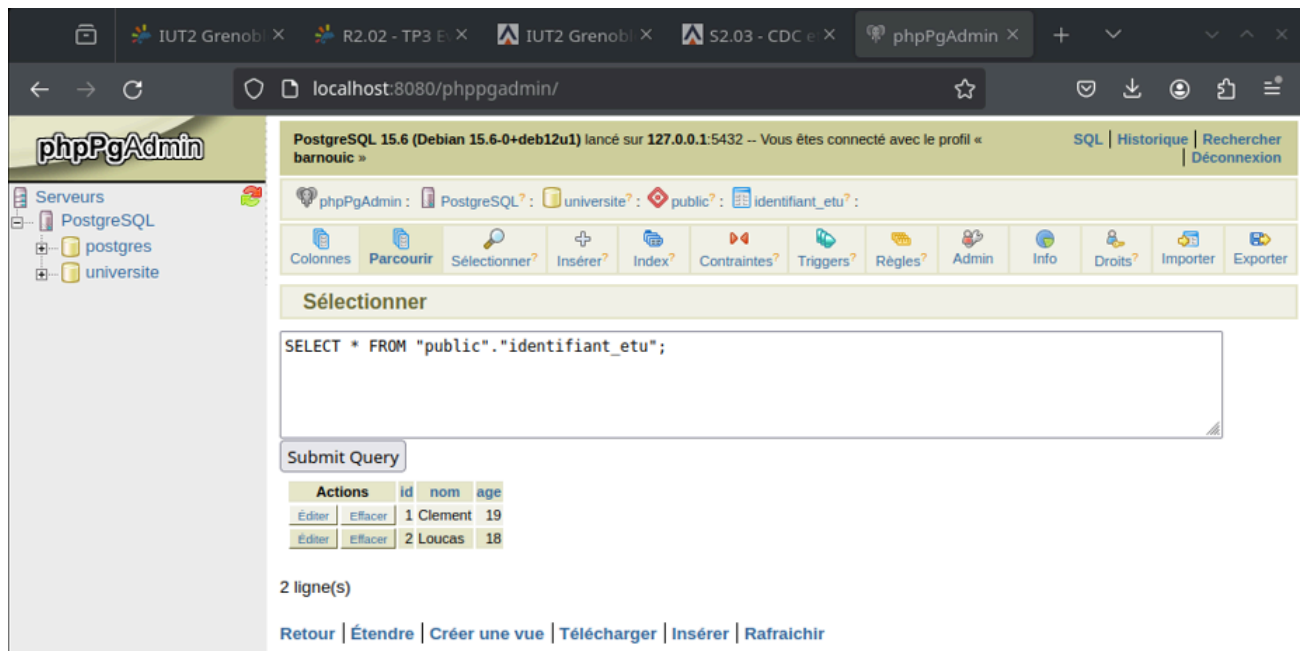


Figure 11. Querying the university database from PhpPgAdmin on the host machine.

Step 9: Security Analysis

1. Final storage space

```
root@barnouic:~# 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  2.0G  803M  72% /
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
root@barnouic:~# _
```

Figure 12. Storage space after installations.

2. Identify and fix potential security vulnerabilities

- Regularly update system packages :

```
# apt-get update && apt-get upgrade
```

- Install Logchecker and apply specific rules if you want or leave as default

```
# apt-get install logcheck
```

```
# nano /etc/logcheck/logcheck.conf
```

- Configure firewalls and access controls appropriately.

Optional Enhancements

1. SSH Access with Key Authentication

```
# systemctl status ssh
```

```
root@barnouic:~# systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-04-09 17:00:12 CEST; 5min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 468 (sshd)
    Tasks: 1 (limit: 4645)
   Memory: 6.7M
      CPU: 26ms
   CGroup: /system.slice/ssh.service
           └─468 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Apr 09 17:00:12 barnouic systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Apr 09 17:00:12 barnouic sshd[468]: Server listening on 0.0.0.0 port 22.
Apr 09 17:00:12 barnouic sshd[468]: Server listening on :: port 22.
Apr 09 17:00:12 barnouic systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
root@barnouic:~#
```

Figure 13. Verifying SSH operation.

Method 1 – Accessing the root account via SSH

```
$ ssh-keygen
```

```
$ ssh-copy-id root@10.0.2.15
```

```
# nano /etc/ssh/sshd_config
```

Decomment / Edit to :

```
PermitRootLogin prohibit-password
```

```
PubkeyAuthentication yes
```

```
# systemctl restart sshd
```

Method 2 – Accessing the SSH Server

- Generate SSH keys and configure the server to accept key-based authentication.

```
$ ssh-keygen
```

```
ssh-copy-id barnouic@10.0.2.15
```

```
# nano /etc/ssh/sshd_config
```

Decomment :

```
PubkeyAuthentication yes
```

```
AuthorizedKeysFile .ssh/authorized_keys
```

```
# systemctl restart sshd
```

2. Enable HTTPS on Apache:

- Generate a self-signed certificate using OpenSSL and configure Apache to use it.

```
# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
```

```
/etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

- Répondez aux questions interactives pour créer votre certificat.

```
# nano /etc/apache2/sites-available/default-ssl.conf
```

Make sure the `SSLCertificateFile` and `SSLCertificateKeyFile` directives point to the correct paths of the certificate and private key files.

```
# a2enmod ssl
```

```
# systemctl restart apache2
```

- Make sure port 443 (HTTPS) is open in your firewall.

Conclusion

This guide provides the essential steps to install and configure a Debian 12 server with Apache, PostgreSQL, and PHP. By following these steps, you should have a functional server environment accessible from your host machine.