Guide to installing a Debian 12 web server on QEMU

Powered by Apache2, PostgreSQL and PHP functional and searchable

METAYER Kylian

31 Mai 2024

Contents

Introduction **2**
- Guide presentation . . . 2
- Tools and their uses . . . 2

Preparing for installation **3**
- Checking debian 12 iso . . . 3

Installing debian **4**
- Starting QEMU . . . 4
- System installation . . . 5

Good practice **7**
- Important command . . . 7
- Note . . . 7

Server check **8**
- Priority check . . . 8
- SSH port and access . . . 8

Installing Apache2 **9**
- Install . . . 9
- Configure . . . 10

Installing PostgreSQL **11**
- Install . . . 11
- Configure . . . 11
- Test interogation . . . 12

Installing PHP **14**
- Install . . . 14

Installing PhpPgAdmin **15**
- Install . . . 15
- Configure . . . 15
- Test interogation . . . 16

Installation check **17**
- Check . . . 17

Security **19**
- Safety improvement strategy . . . 19

Introduction

Guide presentation

This guide is a simple and complete guide on how to install a functional and searchable web server, with the linux operating system under the debian 12 distribution. For this installation, 3 devices will need to be installed and configured. Everything will be done on a QEMU virtual machine

Tools and their uses

We'll need to install three pieces of software to run the server:

- Apache2 a free HTTP software for serving content on the web.
- PostgreSQL our object-oriented relational database management system, used to manage data on a single server, unlike other systems which distribute their function over several nodes. A limit will be imposed on our server in terms of capacity, as it will have to calculate the data, and control frames in the event of overload. The only solution is to take a single, more powerful server without being able to dispatch workloads to several servers.
- PHP and PhpPgAdmin, PHP is a free programming language used to produce functional and dynamic web pages. It can be installed on any type of web server, and is a must-have since it is used in most web sites to interact with databases.
  Then PhpPgAdmin

Preparing for installation

Checking debian 12 iso

We will check the integrity of the iso before launching the installation to avoid any problems in the future.

Go to this link → https://cdimage.debian.org/cdimage/release/current/amd64/iso-cd/
Scroll to the bottom of the page and click on the “SHA512SUMS” link.
Go to /usr/local/images-ISO/ to retrieve the iso provided by the IUT: debian-12.5.0-amd64-netinst.iso then retrieve its fingerprint with the command :
- $ sha512sum File-Name
Finally, compare the fingerprints of the command and those of the site. If they’re not the same, ask the teacher for help.

Installing debian

Starting QEMU

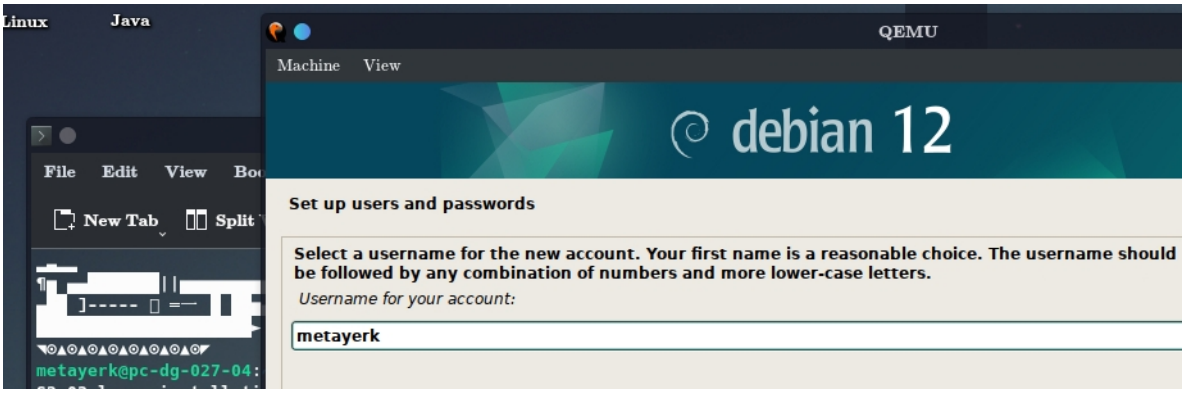
From your shell, issue the command $ S2.03-lance-installation This command should launch the QEMU virtual machine with this parameter:

- -machine q35 : *launching the machine name q35.*
- -cpu host : *Selection the CPU model, here the processor with all supported host feature.*
- -m 4G : *RAM memory size allocate the guest startup, here* 4 *GigaOctet.*
- -enable-kvm : *Enable KVM full virtualization support. This option is only available if KVM support is enabled when compiling.*
- -device VGA : *Video device of the host machine, here VGA.*
- xres=1024,yres=768 : *Resolution mode to* $1024X768$.
- -display gtk : *Select type of display to use, for see the display avaible types -display help, here is gtk.*
- zoom-to-fit=off : *Expand video output to the window size, here off.*
- -drive $drive : *This includes creating a block driver node (the backend) as well as a guest device, and is mostly a shortcut for defining the corresponding -blockdev and -device options.*
- -device e1000,netdev=net0 : *Specify the device e1000, a network interface controllers and supports data rates from 10 to 100 Megabits. It also links to the network device net0*
- -netdev user : Configure user mode host network backend which requires no administrator privilege to run. Valid options are:
  - id=net0 : *Assign symbolic name for use in monitor commands, here 0.*
  - hostfwd=tcp::2222-:22 : *Redirect incoming TCP or UDP connections to the host port hostport to the guest IP address guestaddr on guest port guestport, here port host 2222 → port VM 22.*
  - hostfwd=tcp::4443-:443 : *here port host 4443 → port VM 22.*
  - hostfwd=tcp::8080-:80 : *here port host 8080 → port VM 80.*
  - hostfwd=tcp::5432-:5432 : *here port host 5432 → port VM 5432.*
- The command execute with option → $ 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

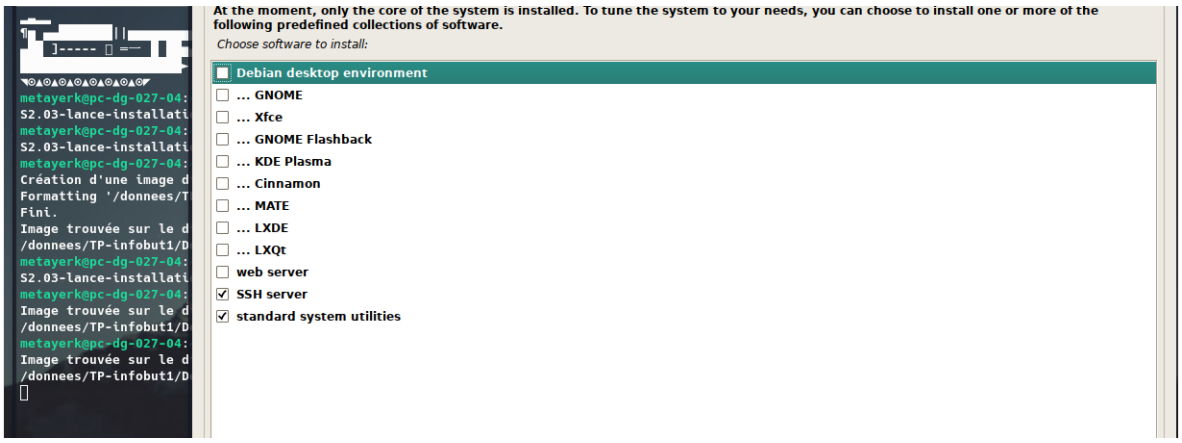
System installation

After executing the command, a QEMU screen will appear on your desktop and the debian 12 installer will be launched.
Here is a list of the main installation steps (if nothing is specified, leave by default):

- Langage : English
- Location : other/Europe/France
- Locales : United States, en_US.UTF-8
- Keyboard : French
- Hostname : serveur-"LOGIN_UGA"

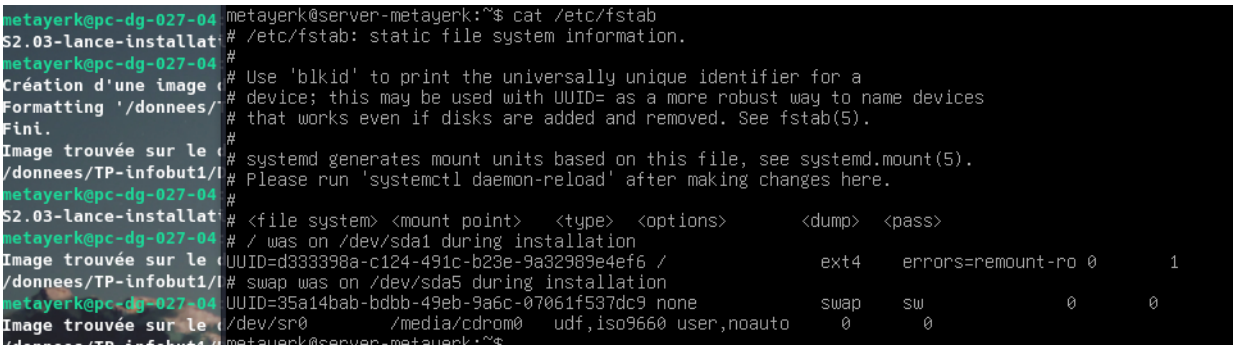


- Root Password : registre a strong password (is the superuser password the most important)
- USER Name : use you login UGA
- USER Password : use a personnal password (is you user account of the server)
- Partition disks : Guided - use entire disk
- Partition disks : All files in one partition
- Partition disks : yes
- Software Selection : check that "Debian desktop" is unchecked and that "ssh server" is checked



- Install GRUB : yes (is you bootloader)
- Device for boot loader : /dev/sda

Finally, after installation, run the command $ cat /etc/fstab in your virtual machine and you should get this result:



Good practice

Important command

After installation of your machine, different commands will be useful for the correct use of your machine.

$ su - : *commande for change user to superuser (root)*

$ apt search Name_Software : *to search for software in a database*

$ apt install Nam_Software : *can request the superuser permission for install, if is the case verify your software are not a virus*

$ apt remove Nam_Software : *can request the superuser permission for removing a software*

$ apt list Nam_Software : *list you software in stockage*

$ duf : *for verify you stockage system*

$ systemctl poweroff : *for stopping you virtual machine, you need the superuse permission*

$ systemctl restart Name_Software : *for restart you software*

$ systemctl {enable/disable} Name_Software : *for start or stop you software*

$ exit : *for return on you user*

Note

For save you system is very important to stop the machine with the systemctl command. For the security of system follow the guide and only install the software gived. If you install other software, verify isn't a virus. ClamAV is a good software for scan you system.

Server check

Priority check

Note the Ethernet and Ip characteristics of your machine with the command :
→ $ ip addr

Then try to communicate between the virtual machine and the host machine by issuing this command on the host machine
→ $ ssh [Nom_User]@[IP_Vitual_Machine] -p 2222

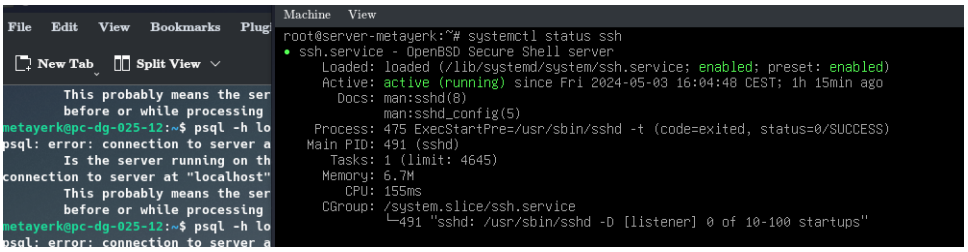
Finally, check that the machine is not equipped with an Xorg server with the command :
→ $ dpkg -l | grep xorg

As Xorg is a windowing system, we prefer to remove it for performance reasons.

SSH port and access

Network service	VM port	Linux station port	Example of use from Linux station
SSH	22	2222	$ ssh toto@localhost -p 2222
HTTP	80	8080	URL: http://localhost:8080/
HTPPS	443	4443	URL: http://localhost:4443/
PostgresSQL	5432	5432	$ psql -h localhost -U postgres postgres

Please note that the virtual machine must be started without a user being logged in for this link to work.
For verify the ssh status system enter the commande
→ $ systemctl status ssh



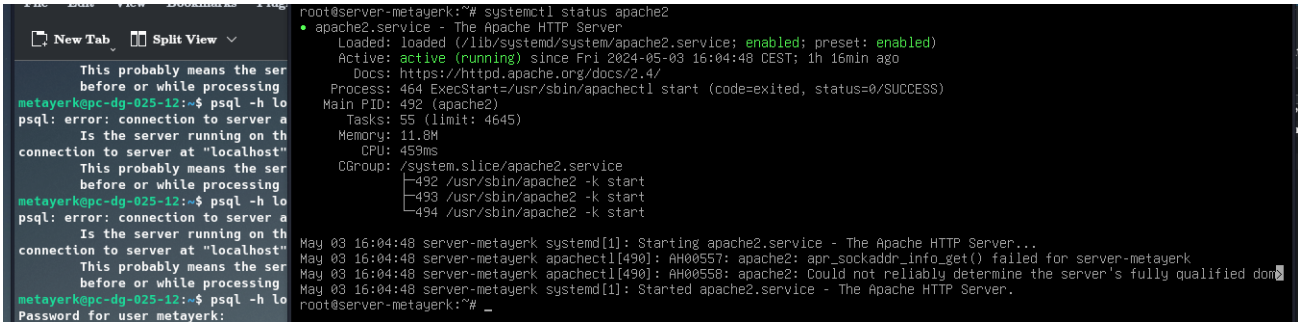
Installing Apache2

Install

To install the Apache2 software, we'll use apt.
To begin with, you'll need to have superuser rights. To do this, review the Good practice section, then install the software with the command :
→ $ apt install apache2

Finally, launch the software with the commands
→ $ systemctl enable apache2
→ $ systemctl start apache2

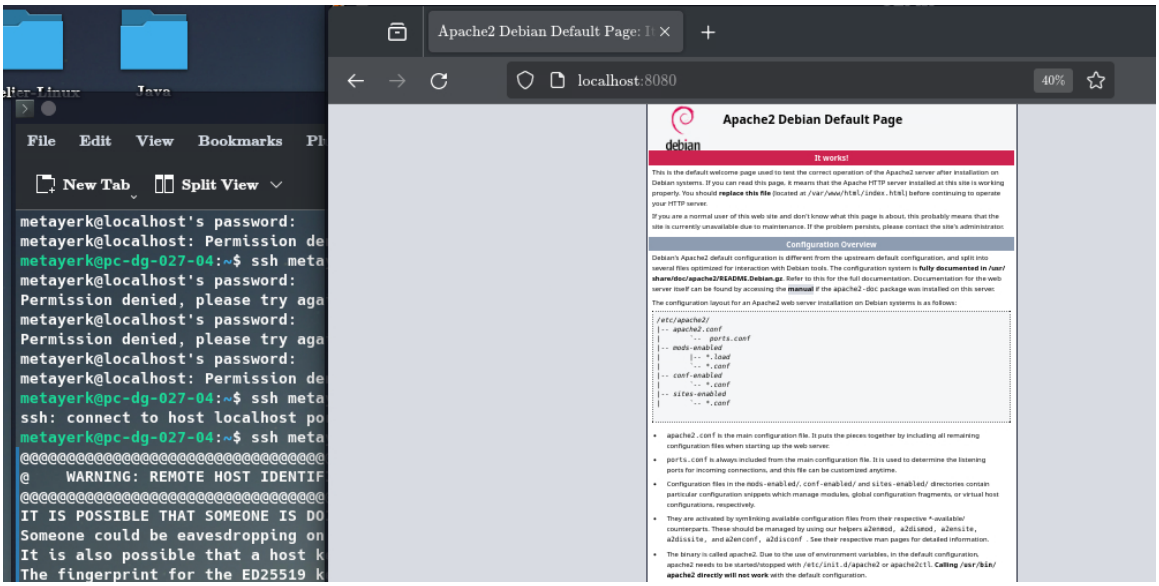
Finally, check that Apache2 is running correctly with the command :
→ $ systemctl status apache2
You should get this result



If not, try restarting apache2 or try again.

Configure

We're going to try to communicate with the server from the host machine. To do this, you can issue a telnet command:
$ telnet localhost 80 *Then write back* HEAD / HTTP/1.0 *This should return* HTTP/1.1 200 OK *Otherwise, restart apache 2.*
Now, on the host machine, try to access the default apache server page with the URL `http://localhost:8080`.



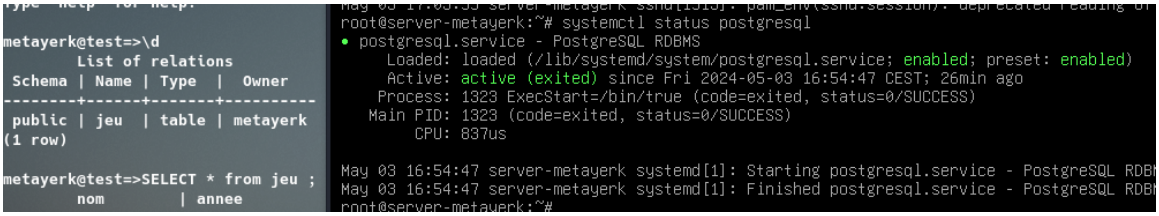
Installing PostgreSQL

Install

To install the PostgreSQL software, we'll use apt.
To begin with, you'll need to have superuser rights. To do this, review the Good practice section, then install the software with the command :
→ $ apt install postgresql

Then check that postgreSQL is running correctly with the command :
→ $ systemctl status postgresql

You should get this result



You can also try to connect to the postgres user to check the installation
→ $ su - postgres

Configure

Now you should be able to connect to the postgres user with the command :
→ $ su - postgres
And finally access the database
→ $ psql -l

Finally, create a user with database creation rights in your name: → $ CREATE USER Name_User WITH password 'User_Password';
→ $ ALTER USER metayerk WITH createdb;
metayerk is my name but replace it with yours.

Finally, we're going to modify a file so that the database can be accessed on the host machine, that is the one linked to the server. To do this, we'll modify a configuration file.
→ $ nano /etc/postgresql/13/main/postgresql.conf

After modifying the line starting with "listen_address" remove from comment mode (#) then modify 'localhost' by '*'
Now removed from comment mode the line 'password_encryption = scram-sha-256'. and if needed, if another entry is written than 'scram-sha-256' then replace it.

Finally modified the config file pg_hba.conf so that it's possible to connect from a machine other than the server.
→ $ nano /etc/postgresql/13/main/pg_hba.conf *Then add the line under the IPv4 comment.*
host all all 0.0.0.0/0 scram-sha-256
If you notice any 'md5' replace them with 'scram-sha-256'.

Restart the service → $ systemctl restart postgresql
Finally, connect to psql from your account with → $ psql Then run the command
→ $ SELECT * FROM pg_shadow;

You'll notice your password encrypted.

Test interogation

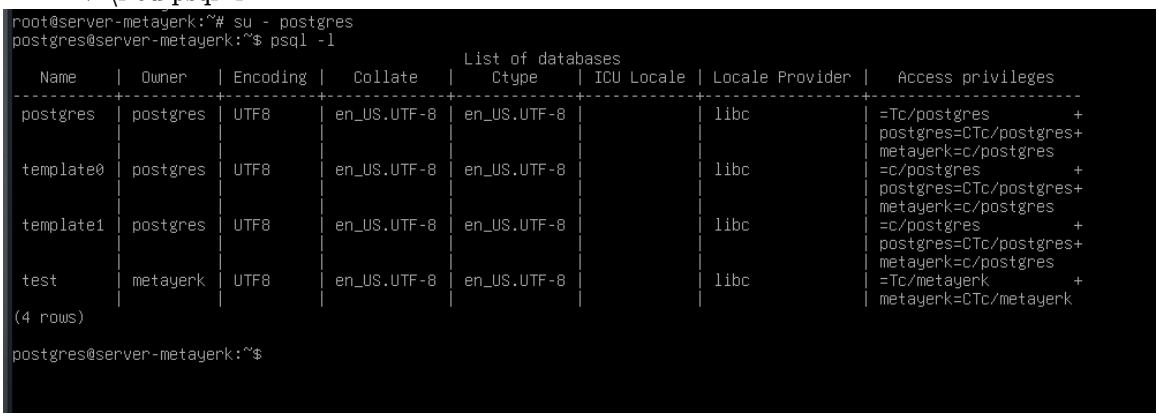
Try connecting to your account from a shell on the host machine and the virtual machine.

→ $ psql -h localhost -U Name_User → *Machine host*

→ $ psql -U Name_User → *Virtual machine*

For different test create a database from your user account:
→ $ CREATE DATABASE Name_Base;

Check the creation of the database with the command
→ \l ou psql -l



Now you can connect to the database with → \c Name_Base Name_User

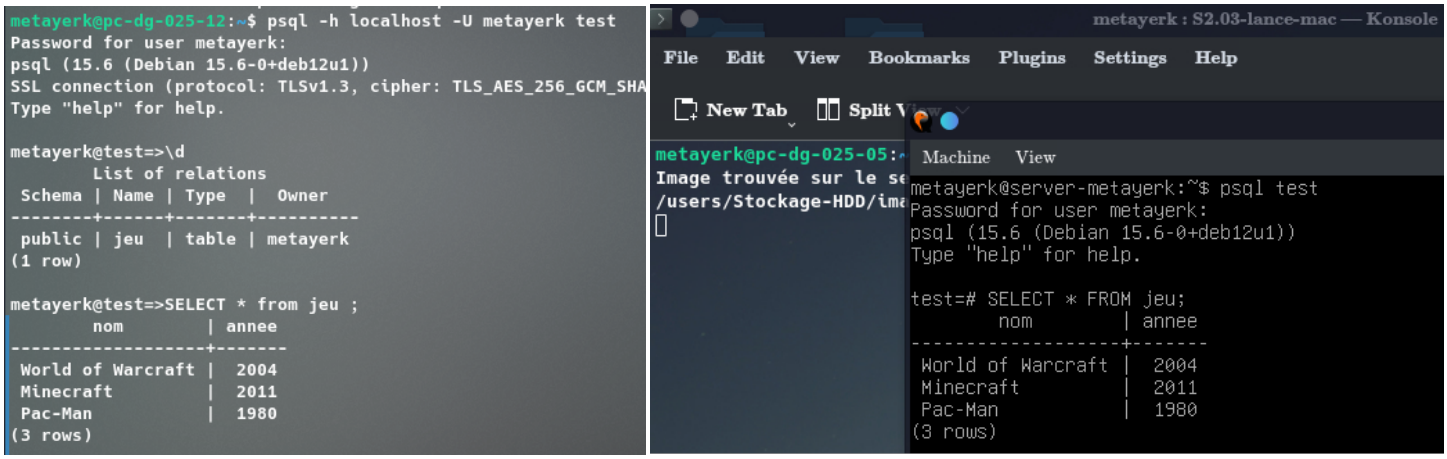
And create a new table in example :
CREATE TABLE jeu(
nom varchar primary key,
annee int);

Insert content in your table :

- INSERT INTO jeu values ('World of Warcraft', 2004)
- INSERT INTO jeu values ('Minecraft', 2011)
- INSERT INTO jeu values ('Pac-man', 1980)

You can check the creation with the command \d

Finally, send a request to the host machine and the virtual machine as follows:
→ SELECT * FROM jeu;



Installing PHP

Install

To install the PHP software, we'll use apt.
To start with, you'll need to have superuser rights, so please review the Good practice section.
Install PHP with the command :
→ apt install php-common libapache2-mod-php php-cli

Restart apache2 server
→ systemctl restart apache2

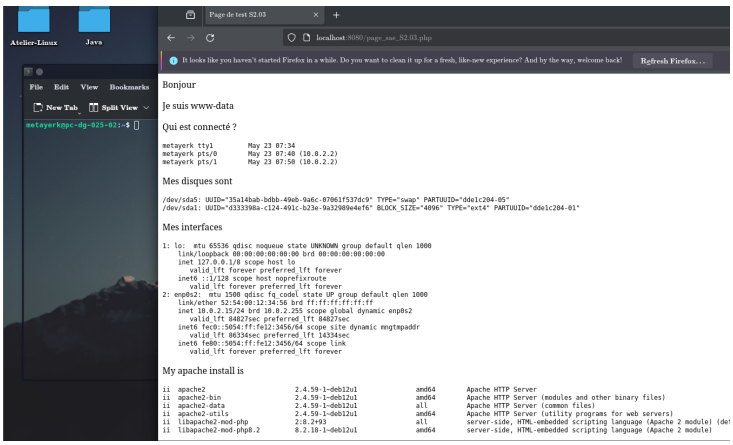
Then create an info.php file in the → /var/www/html directory. Then modify the same file
→ $ nano /var/www/html/info.php

Then write this →

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

This file will give you information on your apache server from the following web link http://localhost:8080/info.php
For security reasons, this file should not be accessible to anyone. For more information, see the Security chapter. Finally, run the following command
→ /sbin/blkid

Copy the page_sae_S2.03.php file given with this command :
→ scp Name_User@transit.iut2.univ-grenoble-alpes.fr:/users/info/www/intranet/enseignements/S2.03/page_sae_S2.03 /var/www/html/
scp (file transfer command)
Name_User@transit.iut2.univ-grenoble-alpes.fr: (the host's adresse)
/users/info/www/intranet/enseignements/S2.03/page_sae_S2.03.php (the destination file)
/var/www/html/ (the transfer destination) You can now open the file in a browser.



Installing PhpPgAdmin

Install

To install the PhpPgAdmin software, we'll use the apt software.
To begin, you'll need superuser rights, so review the Good practice section.
Install PHP with the command :
→ apt install phppgadmin

Restart the apache2 server with
→ $ systemctl restart apache2

Configure

Two files need to be modified for PhpPgAdmin to be accessible on a browser.
In the /etc/apache2/conf-available directory, modify the phppgadmin.conf file with the software
→ nano /etc/apache2/conf-available/phppgadmin.conf

Then comment out (#) the "Require local" line.
This will allow access to the software on the host machine or any machine linked to the Apache server.
Then we'll modify the Connection.php file in the /usr/share/phppgadmin/classes/database directory with the software
→ nano /usr/share/phppgadmin/classes/database/Connection.php

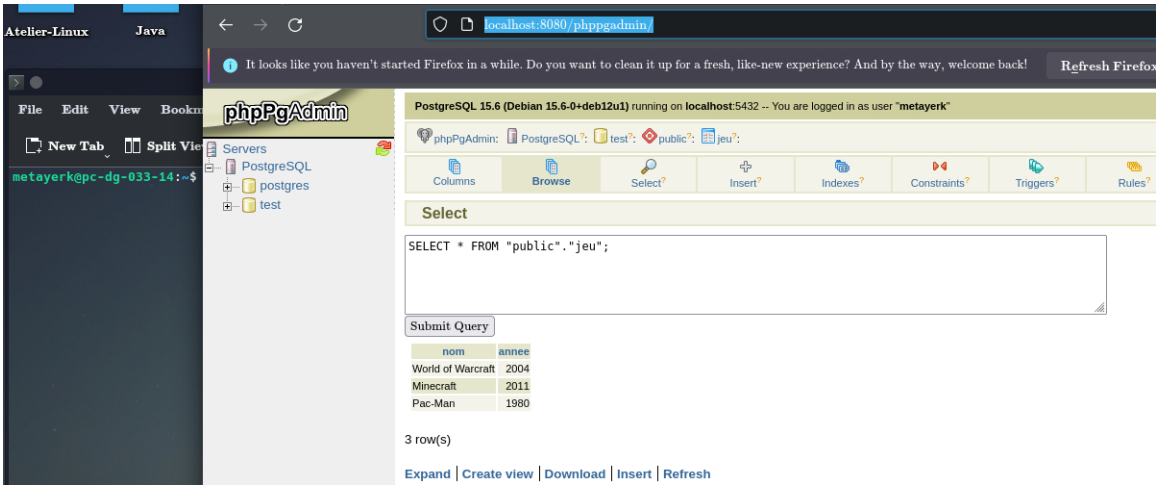
Then modify the line :
case '14': return 'Postgres';break; → case '15': return 'Postgres';break;

Now the software will be accessible from the host machine and machines linked to the server.

Test interogation

You can go to phppgadmin from the following link:
http://localhost:8080/phppgadmin
click on the server tab, enter your log-in details, click on your DATABASE and then on your public table for my game. Click on the Browse tab and enter a query.
Example → SELECT * FROM jeu;

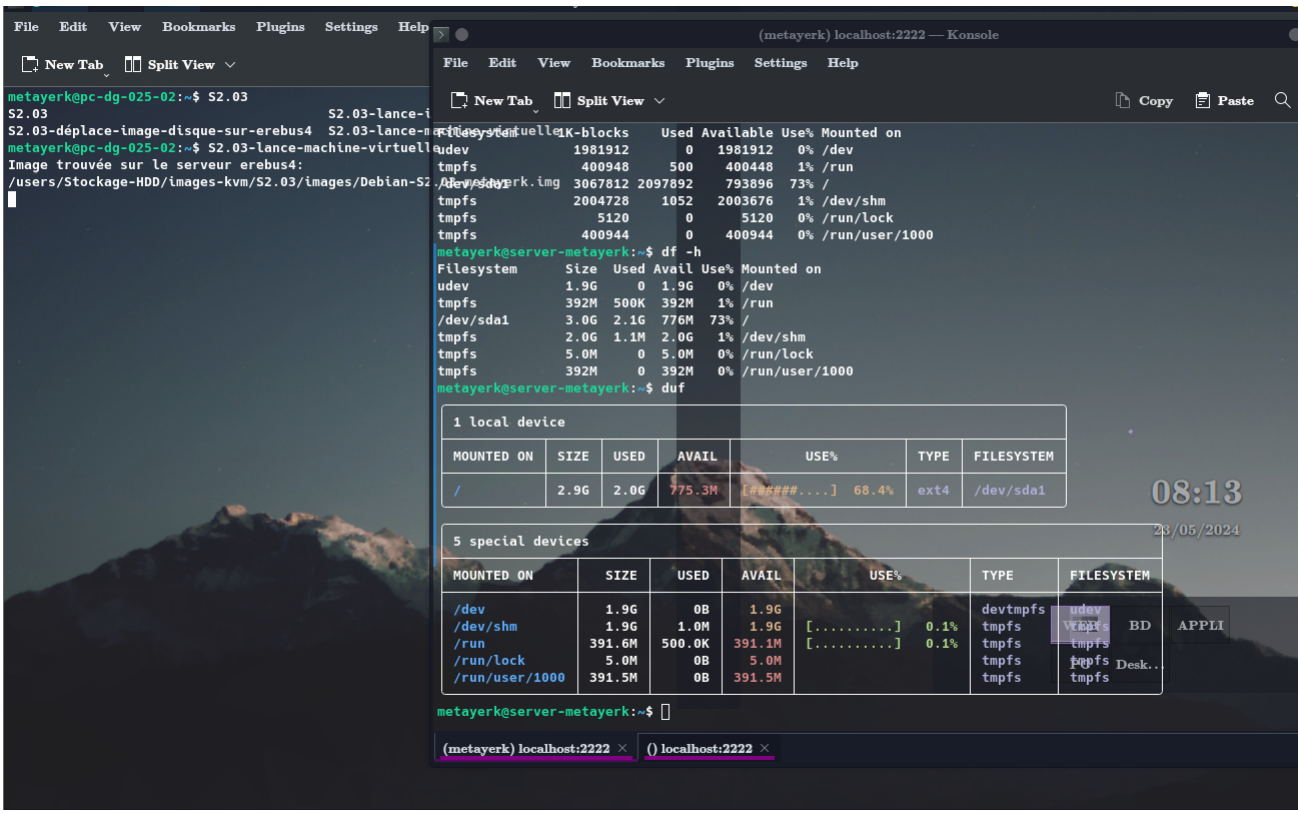
You should have a result displayed below the query.



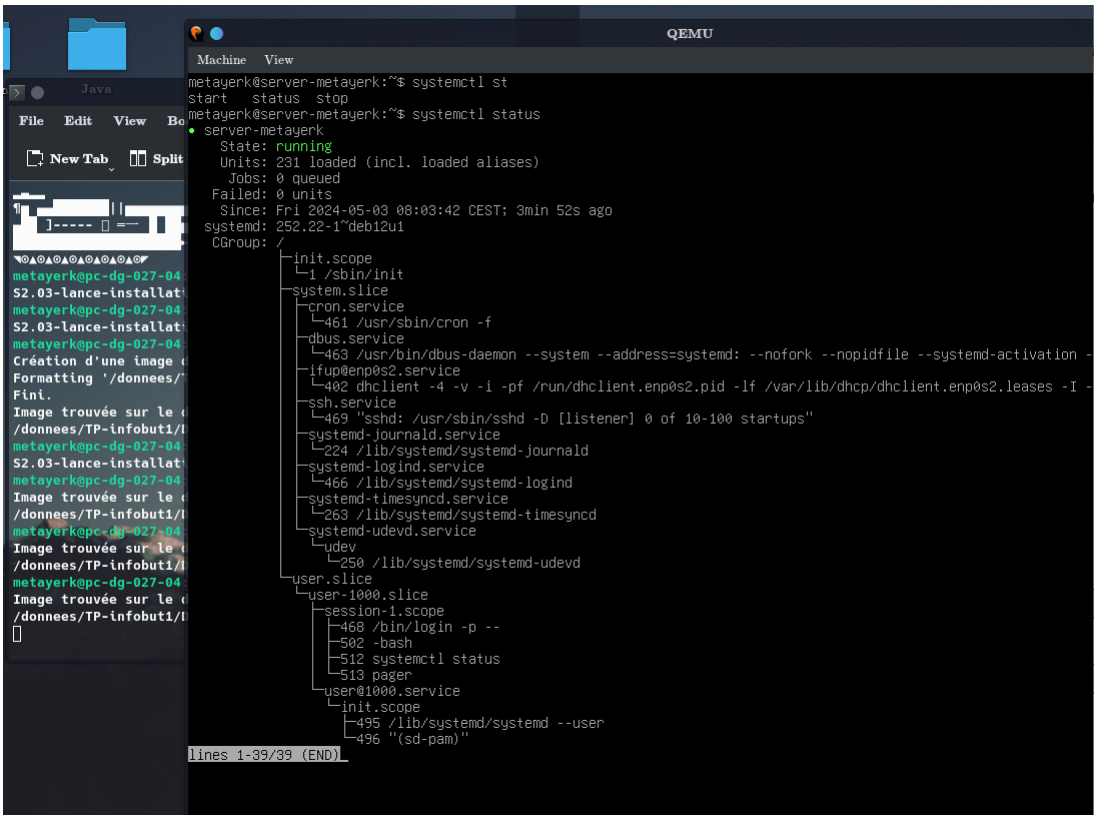
Installation check

Check

Congratulations, your installation is now complete and we can check the installation and status of your machine. To begin, we'll check the machine's storage with the duf software, if it's not present in the machine.
you can install it with a command :
→ $ apt install duf
After installation, run the command → $ duf (you should get this display).



Check that the local device is not overloaded, otherwise check that there are no bad installations that could overload the storage.
At the same time, you can check your machine's process tree with the command.
→ $ systemctl status



Security

Safety improvement strategy

- Restrict access to the info.php file, which gives access to machine information relating to the php installation and is therefore possibly critical information.
- Add Anti-Virus scanning software to check file integrity after each installation and/or update.
- Add Firewall software to control access ports and close them in the event of an attack, and to perform network maintenance on part of the server.
- Perform regular updates with :
  - apt update → check repository database and update
  - apt upgrade → updates software
- Set up an SSL certificate for HTTPS access, a combination of HTTP with an encryption layer
- Hide the Apache2 version so that attackers can't easily discover exploitable vulnerabilities.
- Create ssh keys for key-based access to the server, instead of shorter and often less secure passwords.
- Easier access to the access.log file to check connections on the server