INSTALLATION GUIDE

How to install a Debian 12 server equipped with Apache, PostgreSQL and PHP

Léa Garaix

BUT Informatique – 1st year

IUT2 – UGA

Sommaire

Introduction

This guide provides a step-by-step walkthrough on how to install Debian 12 on a QEMU/KVM virtual machine, along with an Apache server, PostgreSQL, PHP and PhpPgAdmin. Note that this setup will not include a graphical interface. It will be managed entirely via command lines.
The entire installation process should take around 4 hours if you take the time to understand what is happening.

Chapter 0: What Are Those

Apache

Apache HTTP Server is a web server. Web servers handle requests from clients such as web browsers, then deliver the requested content to users. Apache is open-source and can run on various operating systems.

- Apache's website: https://httpd.apache.org/

Debian 12

Debian is a free operating system. It is a Linux distribution composed mainly of free and open-source software. The version 12, named "Bookworm", is the latest stable version as of May 2024. Debian's package management system, APT (Advanced Package Tool) simplifies the process of installing, updating, and managing software packages on Debian systems.

- Debian's website: https://www.debian.org/
- Debian's Wikipedia page: https://en.wikipedia.org/wiki/Debian

PHP

PHP is an open-source script language used for creating dynamic web pages. It embed within HTML. It adds functionalities such as collecting form data, generating dynamic content and interacting with databases.

- PHP's website: https://www.php.net/manual/en/intro-whatis.php
- PHP's Wikipedia Page: https://en.wikipedia.org/wiki/PHP

PhpPgAdmin

PhpPgAdmin is a web application written in PHP that simplifies the administration of PostgreSQL databases. It provides a graphical interface for managing databases, running SQL queries and viewing tables.

- PhpPgAdmin GitHub Repository: https://github.com/phppgadmin/phppgadmin

PostgreSQL

PostgreSQL is an open-source relational database management system. It uses the SQL language. It runs on many operating systems.

- PostgreSQL's website: https://www.postgresql.org/about/

QEMU/KVM

We will install Debian on a virtual machine emulated by QEMU. QEMU is an open-source machine emulator and virtualization tool. It allows users to run virtual machines on their host systems.

The KVM (Kernel-based Virtual Machine) is a virtualization module for the Linux kernel. It provides hardware virtualization support and enhances the performance of virtual machines.

- QEMU's Website: https://www.qemu.org/
- QEMU's Wiki main page: https://wiki.qemu.org/Main_Page
- QEMU's Wiki KVM page: https://wiki.qemu.org/Features/KVM

VM (Virtual Machine)

A software emulation of a physical computer that runs an operating system and applications.

"$" and "#"

In terminal commands, you will see symbols "$" or "#" at the beginning. "$" is for commands that can be entered by a basic user. "#" is for commands used by the root account. These symbols are already displayed in the terminal and cannot be deleted, so you don't need to type them.

Chapter 1: Installing Debian 12

1) Downloading the ISO image and checking its integrity

We will install Debian 12 (“Bookworm”), version 12.x, for x86 64-bit processors using the "netinst" **ISO image** type.

The ISO image and the files needed to verify its integrity can be found here:
https://cdimage.debian.org/cdimage/release/current/amd64/iso-cd/

> **ISO image**
>
> It is a file containing the exact copy of all the data stored on an optical disc.

In the context of our project, the image has been pre-downloaded to save time and disk space.

- Open the folder where the ISO image is stored:
  `cd /usr/local/images-ISO/`
- Verify the integrity of the ISO image:
  - Go to the download page above (where you could download the ISO image).
  - Open the file “SHA512SUMS”.
    → It contains the SHA-521 checksums.
  - In the terminal, enter this command:
    `$ sha512sum NAME-OF-THE-ISO-IMAGE-FILE`
    → It computes the SHA-512 checksum of your ISO image.
  - Compare the two SHA-512 strings: they should be identical.
    → If the checksums do not mach, delete the ISO image and download it again. But before, ensure you are using the correct “SHA**512**SUMS” file.

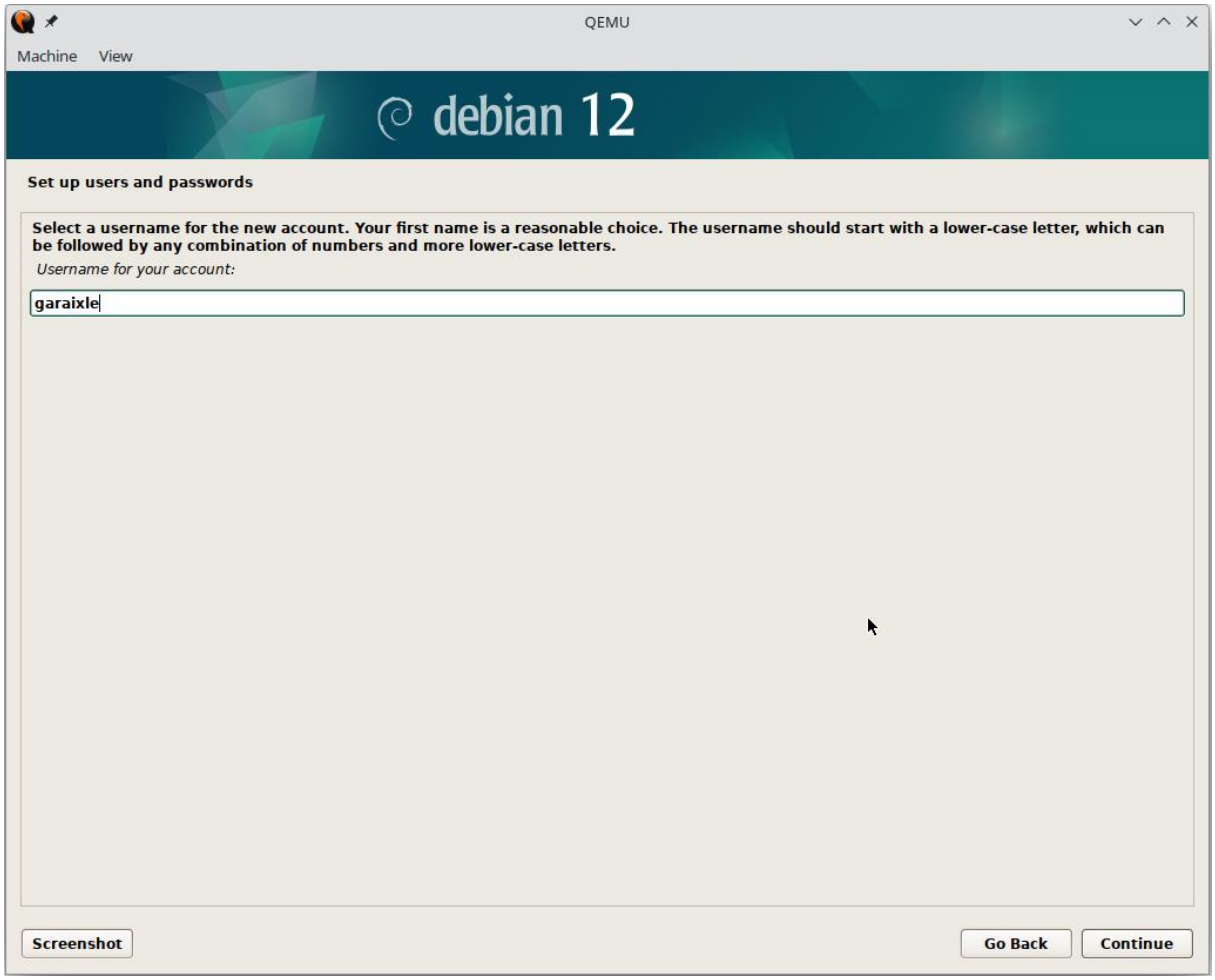
> **SHA-512**
>
> “Secure Hash Algorithm 512”.
>
> It converts text of any length into a fixe-sized string (512 bits, so 64 bytes).

2) Installing the Debian system

To launch the virtual machine, we use QEMU/KVM.

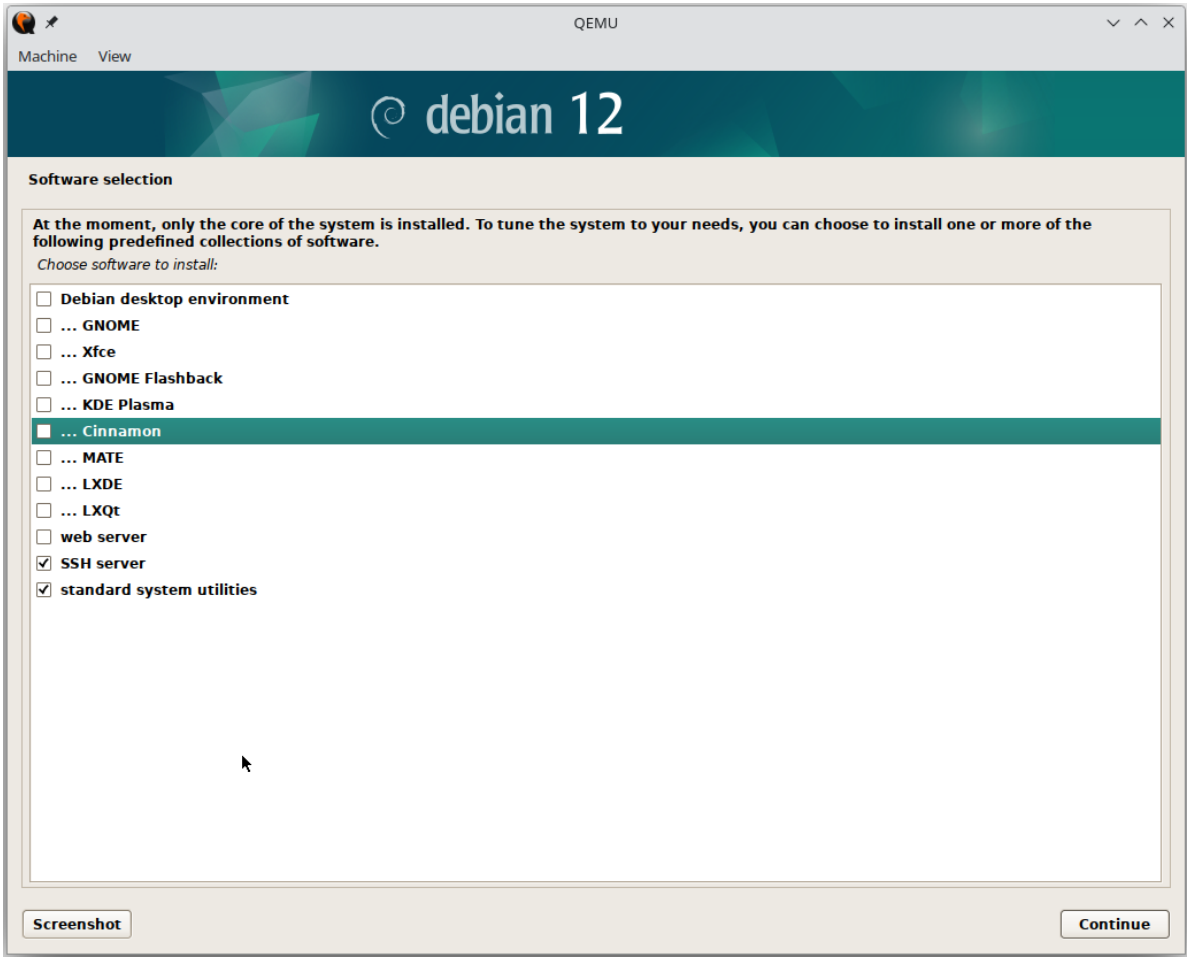
- Ensure that no other users are connected to the station:
  `$ who`
  → If there are other active connections, you may encounter issues during the installation process.
- Launch the virtual machine:
  `$ S2.03-lance-installation`
  → This command executes a script to initiate the installation process. *For more information, look in annexe!*
- Choose the option “Graphical install” to utilize a graphical interface for the installation process.

- Follow the instructions below to complete the configuration. If not specified, maintain the default options:
    - Language: English
    - Location: other/Europe/France
    - Locales: United States, en_US.UTF-8
    - Keyboard: French
    - Hostname: server-loginUGA (example : server-garaixle)
    - Domain name: *empty*
    - Root password: choose an easy password, such as 'root' (no security problem)
    - Full name for the new user: your full name
    - Username: login UGA (example: garaixle)



1. Debian installer, login input (Screenshot 1)

    - Password: easy too, such as 'etu'
    - Partition disks: Guided - use entire disk
    - Partition disks: All files in one partition
    - Partition disks: (Write the changes to disks?) Yes
    - Software Selection: check that "Debian desktop environment" isn’t ticked and that "ssh server" is ticked



2. Debian installer, software selection (Screenshot 2)

- Install GRUB: Yes
- Device for boot loader: /dev/sda

Once the installation is complete, the **virtual machine (VM)** restarts. You will be prompted with a terminal to connect to the server.

Before proceeding to the next step, shut down the VM:

- Connect to the server with the **root** account:
  - Login: root
  - Password: root (if you followed the advice)
- Enter the command :
  `# poweroff`

The VM should be shut down.

Root

The user « root » is privileged user having rights to administer the whole system.

3) Moving the ISO image

We need to move the ISO image to the server `erebus4`. This allows you to access the virtual machine from any other station on the network. You can also copy the image to a USB key if needed.

- Ensure the virtual machine is off.
- Move the ISO image to `erebus4`:
  `$ S2.03-déplace-image-disque-sur-erebus4`

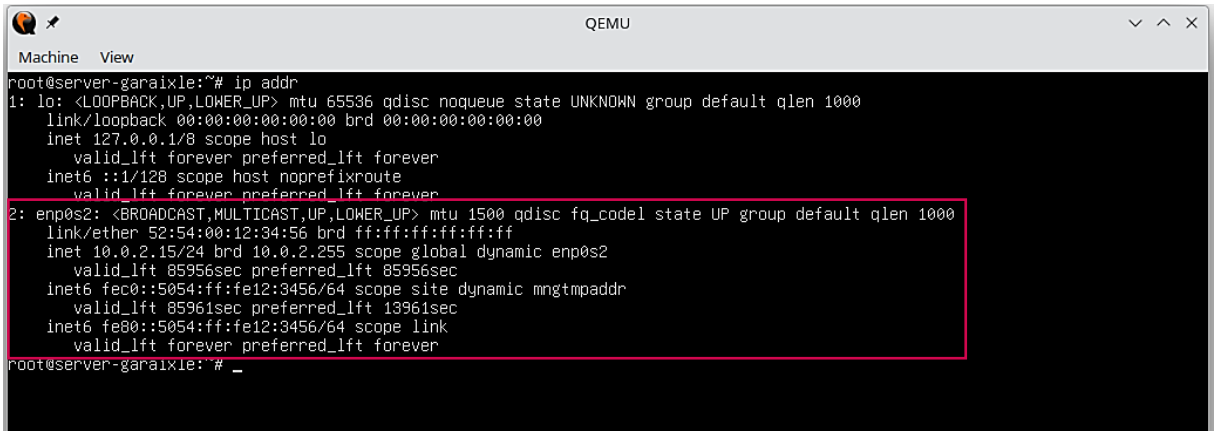
Once this step is complete, you can launch the virtual machine from any other station using the command:

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

4) Checking that everything is alright

Find the Ethernet and IP characteristics of your virtual machine.

- Launch the virtual machine. Reminder:
  `$ S2.03-lance-machine-virtuelle`
- Connect with the root account.
- Display the Ethernet and IP characteristics of the VM:
  `# ip addr`



3. Ethernet and IP characteristics

The command output shows various network interfaces. We are interested in the section for the Ethernet interface, here named "`enp0s2`".
The IPv4 address (and its CIDR notation) is `10.0.2.15/24`.
The IPv6 address is `fec0::5054:ff:fe12:3456/64`.

- Verify that the **X.org** server is not installed on the VM:
  `# dpkg -l | grep xorg`
  → If there is no output, it means the X.org server is not installed, which is what we want.

> **X.org**
>
> X.org provides the fundamental graphical environment for Unix-like operating systems. It enables graphical user interfaces for applications and user interaction. Since we want the virtual machine to operate using command lines only, we don't need an X server.

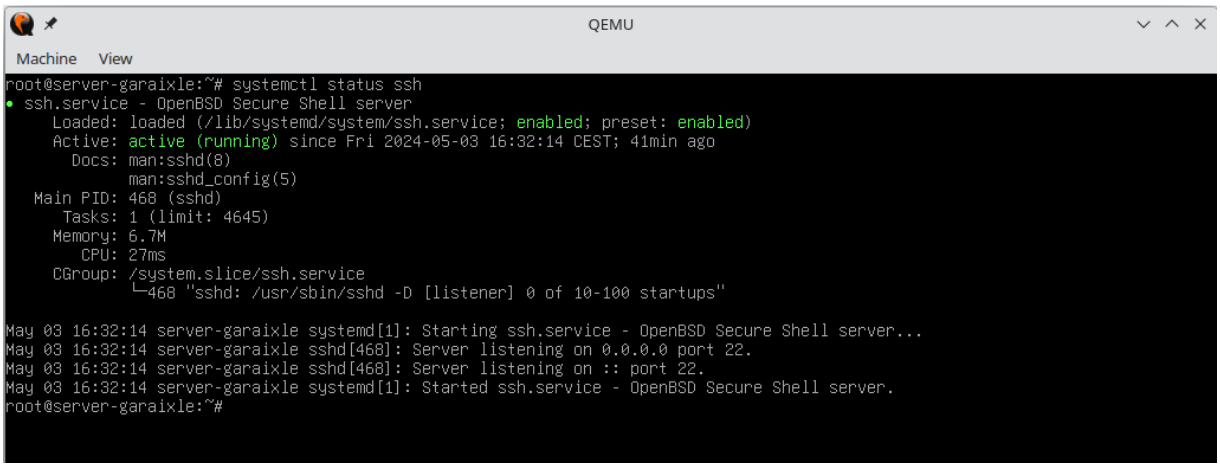
5) Accessing the virtual machine with SSH

To allow you to access the servers running on your VM from clients on your Linux station, port forwarding has been set up by the teaching team:

Network service	VM port	Linux station port	Example of use from a Linux station
SSH	22	2222	`$ ssh loginUGA@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`

Check the status of SSH:

- On your VM, with the root account, enter the following command:
  `# systemctl status ssh`
  → You should see a status message indicating that the SSH service is active and running, as in the image below.



4. Status command result for SSH (Screenshot 4.1)

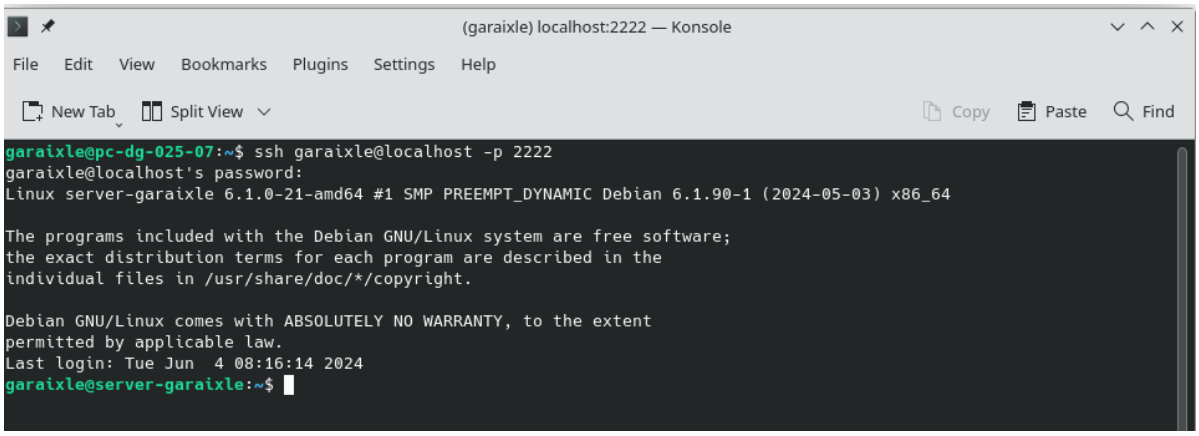
Connecting to you VM with SSH

To connect with SSH to your virtual machine, use the simple user account created during the OS configuration (login UGA and password “etu”).

- Ensure the virtual machine is running.
- On your Linux station, enter the following command, replacing “loginUGA” with your own login:
  `$ ssh loginUGA@localhost -p 2222`

→ The first time you connect, you may need to authorize the connection by typing `yes`.

Your terminal should now display a different header. It indicates that you are connected to the V:



5. SSH connection

Once connected with your user account, switch to the root account by entering:

- `$ su -`
- Enter the password of the root account when prompted.

6) Installing Debian packages

Debian uses the APT (Advanced Package Tool) installer, which makes upgrading and managing **packages** easier by handling dependencies and automating updates and cleanups.

Packages

Archives containing the files, information and procedures needed to install a software on an operating system.

- Connect to the virtual machine with your root account.
- Find package names with the command:
  `# dpkg -list | grep 'keyword'`
  → The keyword can be a part of the name or a word that describes the functionality of the software you are looking for.
- Install a package:
  `# apt install name-of-the-package`
- Clean up useless files:
  `# apt clean`

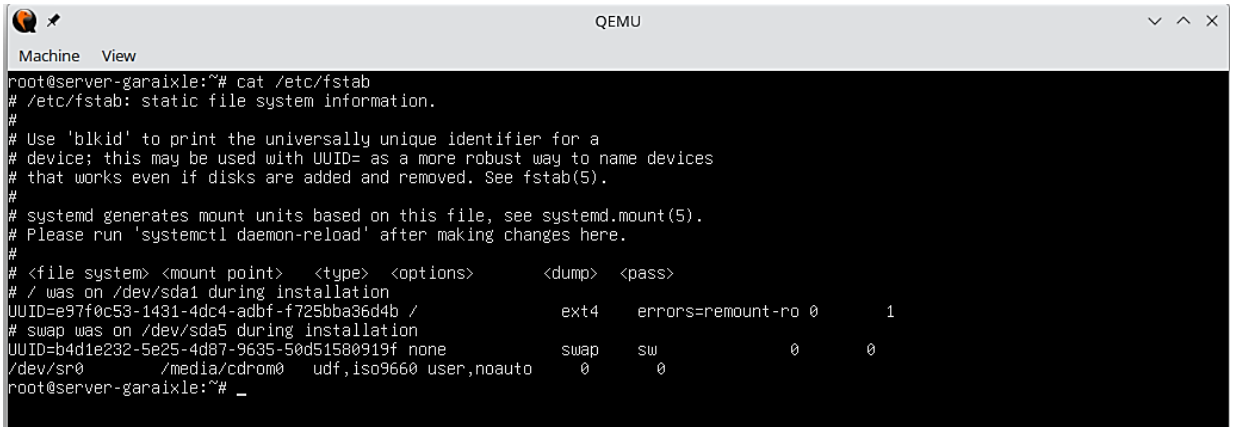
Those are some packages you could install:

- command-not-found:
  `# apt install command-not-found`
  → This software suggests packages to install when you enter an unknown command.
- micro:
  `# apt install micro`
  → Micro is a text editor. Once installed, you can launch it by entering “micro” in the shell.

The `fstab` file (located in `/etc/fstab`) is the configuration file that contains information about mounting file systems. It lists all available disks and partitions and indicates where to mount them in the Linux filesystem hierarchy.

It is useful to examine the content of this file to make sure configurations are correct.

- In the virtual machine, enter the command:
  `# cat /etc/fstab`



6. The fstab file (Screenshot 3)

Chapter 2: Installing Apache2

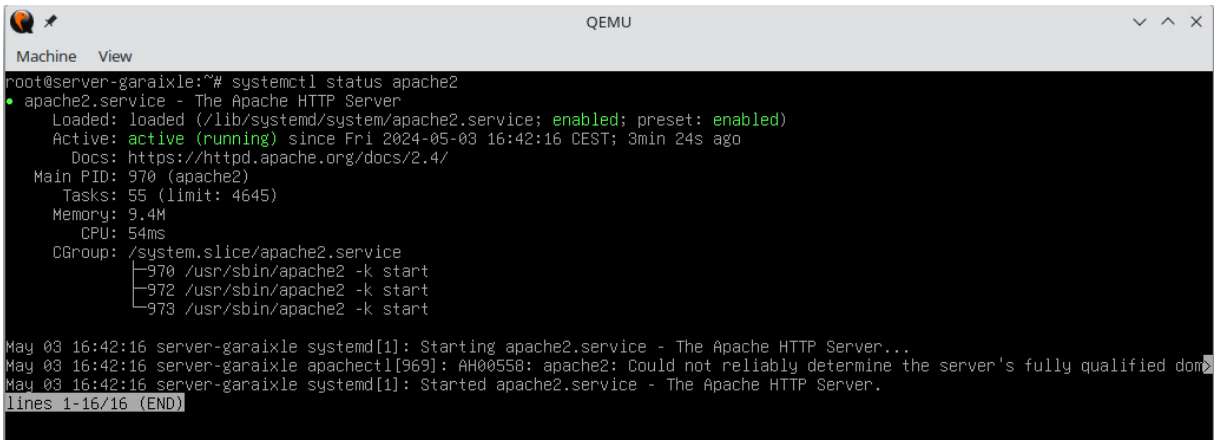
1) Installing Apache2

For more information, you can read the Apache2 documentation:
https://httpd.apache.org/docs/2.4/en/install.html

- Open a root shell on your VM.
- Install Apache2:
  `# apt install apache2`
- Start Apache2:
  `# service apache2 start`

Check if it is installed and started:

- Check the status:
  `# systemctl status apache2`
  → You should see an output indicating that Apache2 is active and running, as on the image below. If you do not see this, it means Apache2 isn't running. Start it with:
  `# systemctl start apache2`



7. Status command result for Apache2 (Screenshot 4.2)

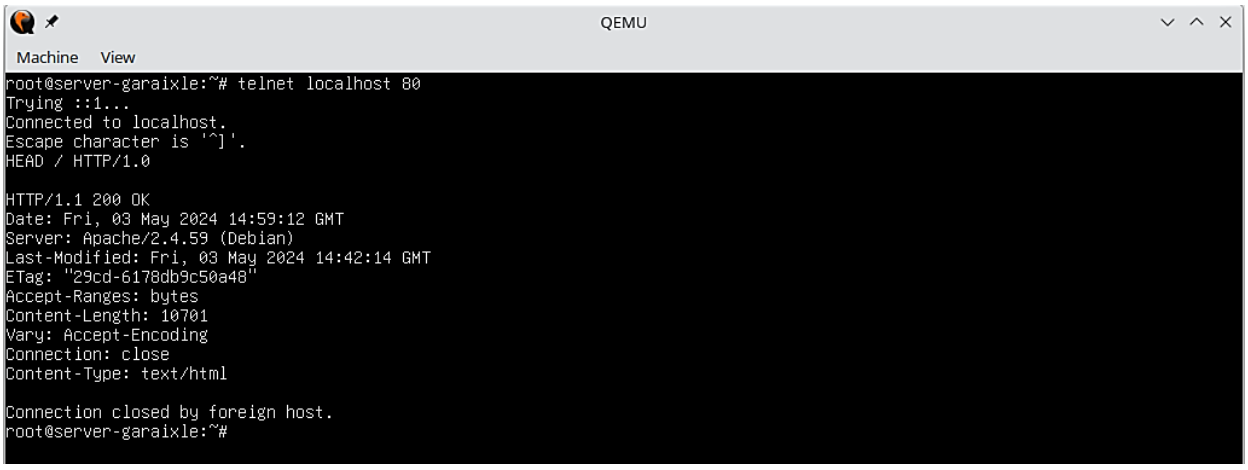
2) Connecting to the Apache server

The server we use has no graphical interface, so it can't display HTML pages directly. To connect to the Apache server, we use **Telnet**.

- Open the connection:
  `# telnet localhost 80`
  → The 80 port is the HTTP port of the virtual machine.
- Enter the following command:
  `HEAD / HTTP/1.0`
- Press the « Enter » key **twice**.

Telnet

"Teletype Network Protocol". This protocol allows text-based interactions between distant computers over a network.

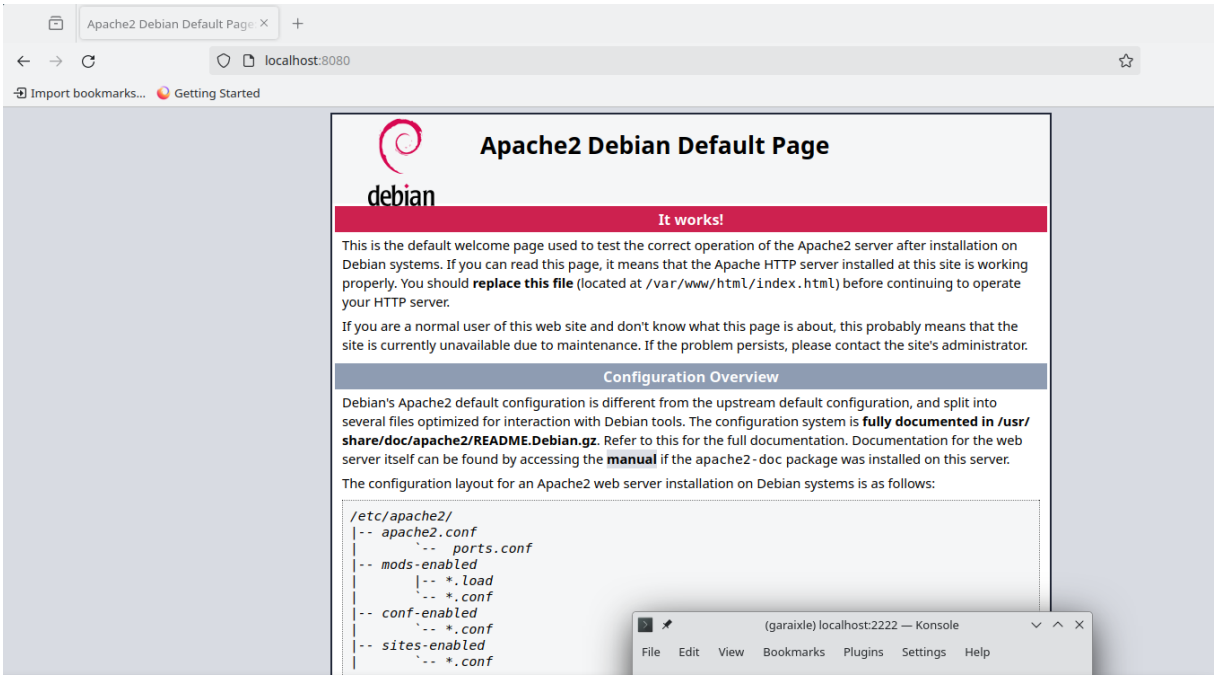


8. Telnet connection

Although we can't display a web page directly on the virtual machine, we can do it from the host machine. The script we executed to launch the virtual machine includes port forwarding from the host machine to the VM. In this case, port 8080 on the host machine is redirected to port 80 on the VM.

To see the HTTP request results:

- Open a browser on your host machine.
- Enter the URL:
  http://localhost:8080
  → This will display the default Apache2 web page, confirming that the Apache server is running and accessible.



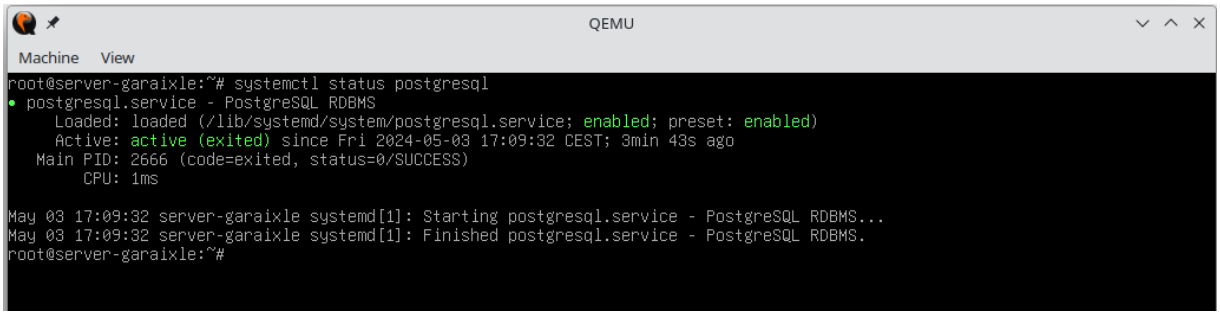
9. Default Apache2 web page (Screenshot 5)

Chapter 3: Installing PostgreSQL

For more information, you can read the PostgreSQL documentation: https://www.postgresql.org/

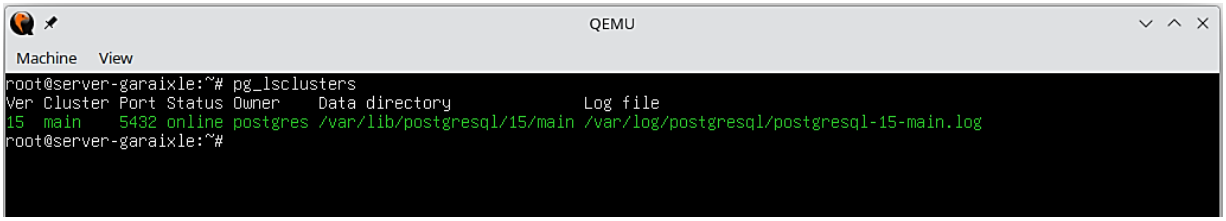
1) Installing PostgreSQL

- Connect to the VM with the root account.
- Install PostgreSQL:
  `# apt install postgresql`
- Check if it is installed and started:
  `# systemctl status postgresql`
  → You should see an output indicating that PostgreSQL is active and running, as on the image below.



10. Status command result for PostgreSQL (Screenshot 4.3)

- Verify the installation:
  `# pg_lsclusters`
  → You should see the details of the PostgreSQL clusters:



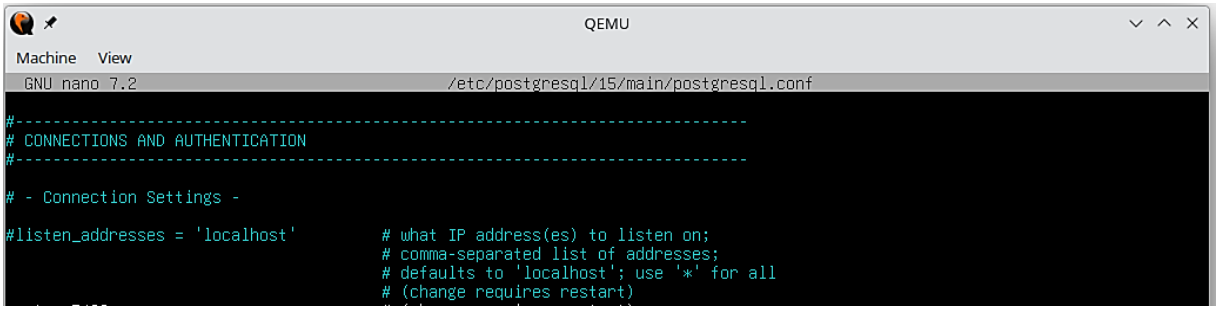
11. PostgreSQL clusters details

2) Configuring PostgreSQL

We want to be able to connect to the server remotely with the users we will create.

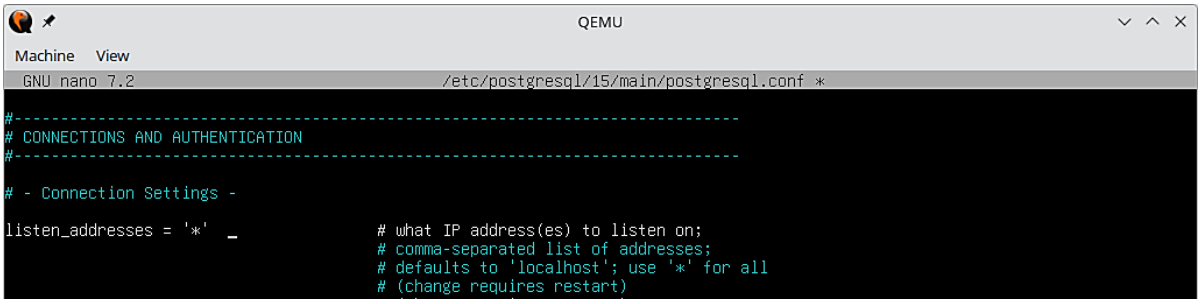
- Connect you on the VM with your root account.
- Launch the PostgreSQL server:
  `# su - postgres`
- Open the configuration file:
  `$ nano /etc/postgresql/15/main/postgresql.conf`
- Edit the configuration:

- Find the section "`CONNECTIONS AND AUTHENTICATION`"
- Find the line "`listen_adresses`"



12. "Listen_adresses" before

- Remove the # and replace "`localhost`" with "`*`".
  → The final line should look like this:
  `listen_adresses = '*'`



13. "Listen_addresses" after

- Save the file with *`Ctrl + S`* and exit with *`Ctrl + X`*.

Now, we will define an authentication rule for connection requests from non-local IP addresses. We will only allow connections authenticated by a password hashed stocked with a strong hash function.

- Define an authentication rule:
  - Open another configuration file:
    `$ nano /etc/postgresql/15/main/pg_hba.conf`
  - Find the line:
    `#IPv4 remote connections`
  - Add this rule below it:
    `host all all 0.0.0.0/0 scram-sha-256`
  - Save the file when exiting.
- Restart PostgreSQL to apply changes:
  `# service postgresql restart`
- You can now **connect with SSH to your PostgreSQL server**. On your Linux station:
  `$ psql -h localhost postgres`

14. SSH connection to the PostgreSQL server

3) Creating users and a database

Practice some SQL by creating a database and users and assigning rights.

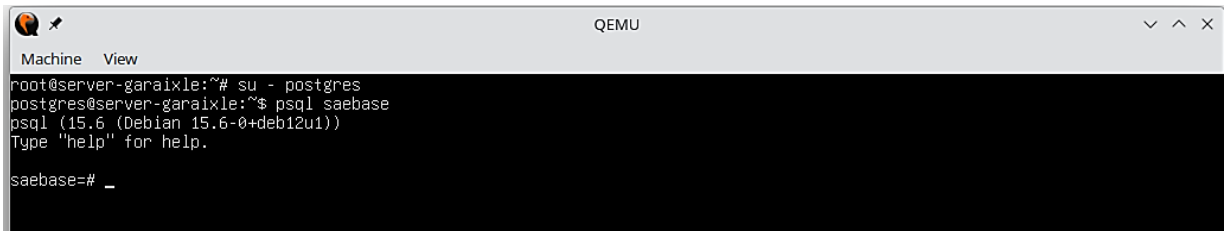
- Connect to the PostgreSQL server:
  `# su - postgres`
- Connect to the PostgreSQL database:
  `$ psql`

First, we will create a new user named after our UGA login.

- Create a new user:
  `CREATE USER garaixle WITH createdb createrole;`
  → If you need to delete the user:
  `DROP USER garaixle;`
- Assign a password to your user:
  `\password garaixle`
  → Do not forget it!

Now, we will create a database owned by your user.

- Create the database (here, its name is “saebase”):
  `CREATE DATABASE saebase WITH OWNER=garaixle;`
- Connect to the new database:
  - Exit the default “postgres” database:
    `\q`
  - Connect to the new database:
    `$ psql saebase`



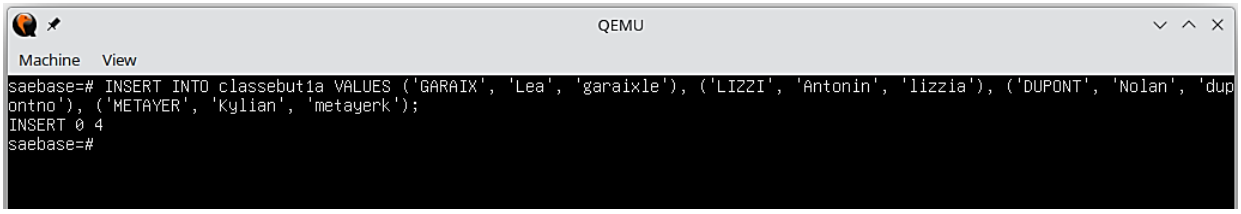
15. Connection to the "saebase" database

- Create a table in the database:

```
CREATE TABLE classebut1a (
          nom     varchar,
          prenom  varchar,
          login   varchar(8)
);
```

- Insert data into the table:

```
INSERT INTO classebut1a VALUES ('GARAIX', 'Lea', 'garaixle'),
('LIZZI', 'Antonin', 'lizzia'), ('DUPONT', 'Nolan',
'dupontno'), ('METAYER', 'Kylian', 'metayerk');
```



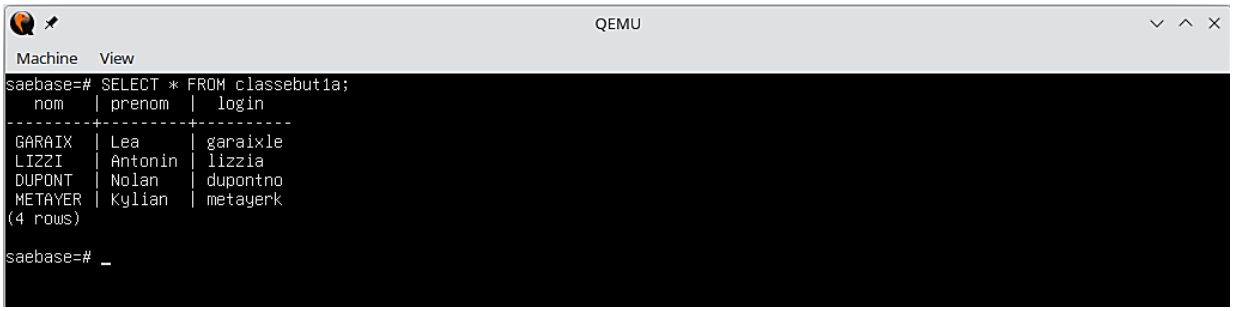
16. Data insertion

→ If you want to delete data:

```
DELETE FROM classebut1a WHERE nom='LIZZI' OR nom='METAYER' OR
nom='GARAIX' OR nom='DUPONT';
```

- View the data:

```
SELECT * FROM classebut1a;
```



17. Interrogation from the virtual machine (Screenshot 6)

Useful commands:

- See the list of users and their roles:

```
\du
```

- See the list of tables in the database:

```
\d
```

- Exit the database:

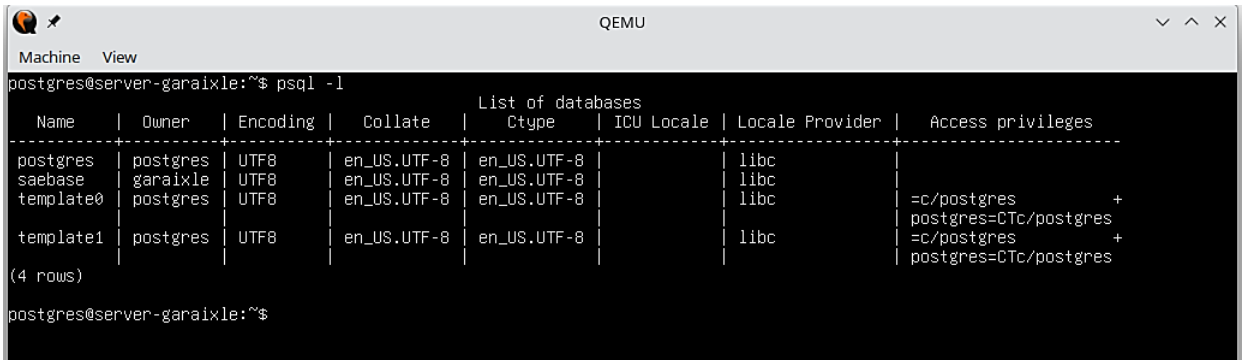
```
\q
```

4) Exploration

If you want to know all the databases that your server hosts:

- Connect to the server (not in a database).

- Request the list of databases:
  `$ psql -l`
  → In the image below, you can see the database “saebase”: “garaixle” is listed as the owner.

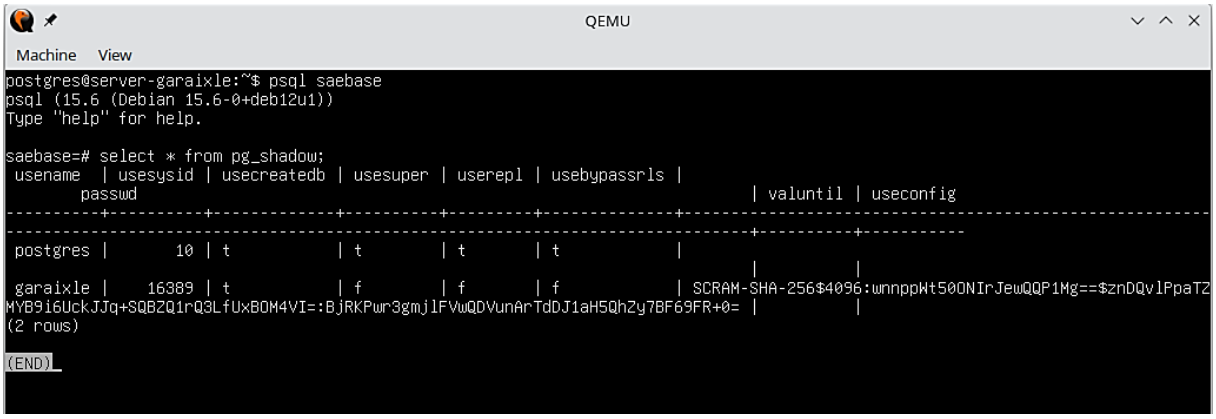


18. Databases list (Screenshot 8)

Understanding `pg_shadow`

> **pg_shadow**
>
> It is a system view in PostgreSQL that contains information about database user accounts. It stores details such as the username, encrypted password, whether the account is a superuser or not… It allows administrators to manage accounts and their privileges.

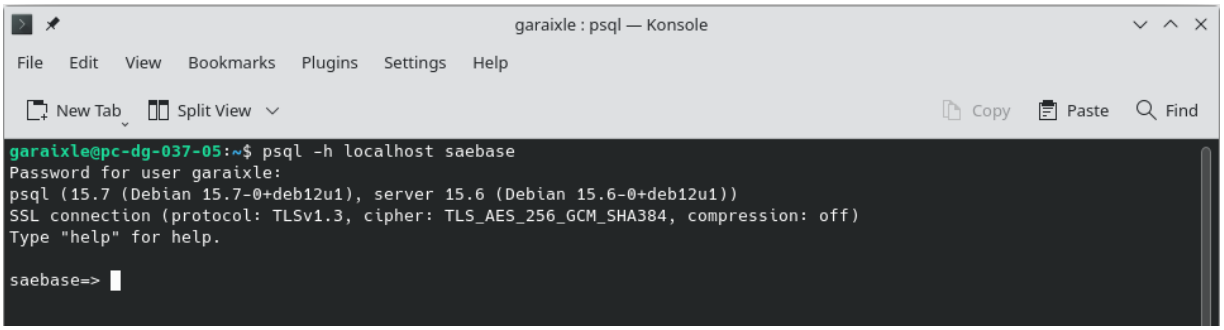
- Connect to a database.
- View `pg_shadow`:
  `SELECT * FROM pg_shadow;`



19. View on pg_shadow (Screenshot 9)

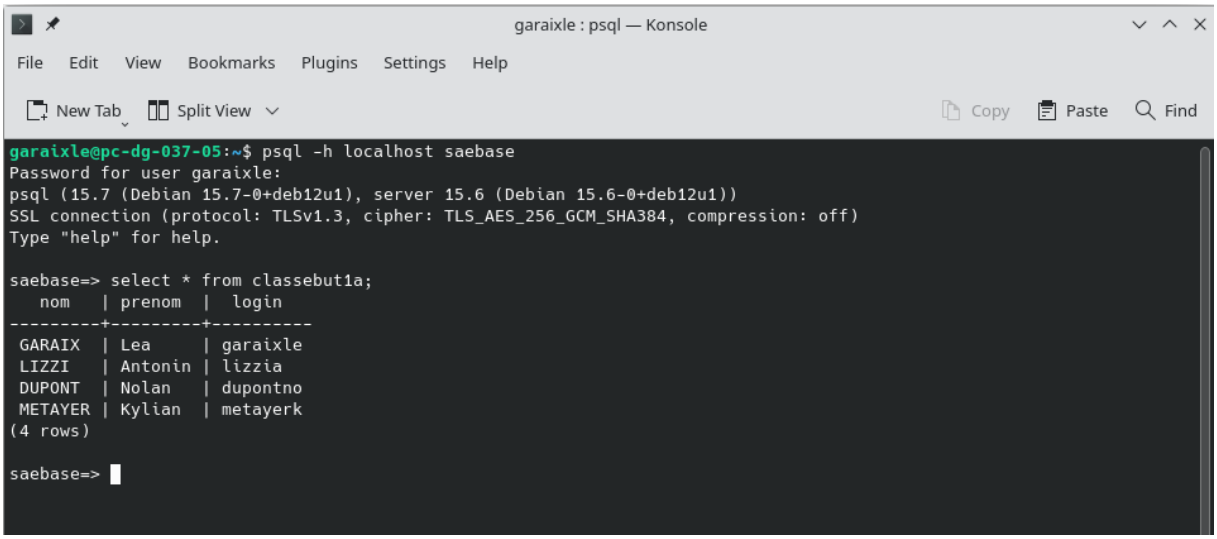
Connect via SSH to your database

- Connect:
  `$ psql -h localhost saebase`



20. SSH connection to the database "saebase"

Then, you can execute your SQL queries:



21. Interrogation from the Linux station (Screenshot 7)

Chapter 4: Installing PHP

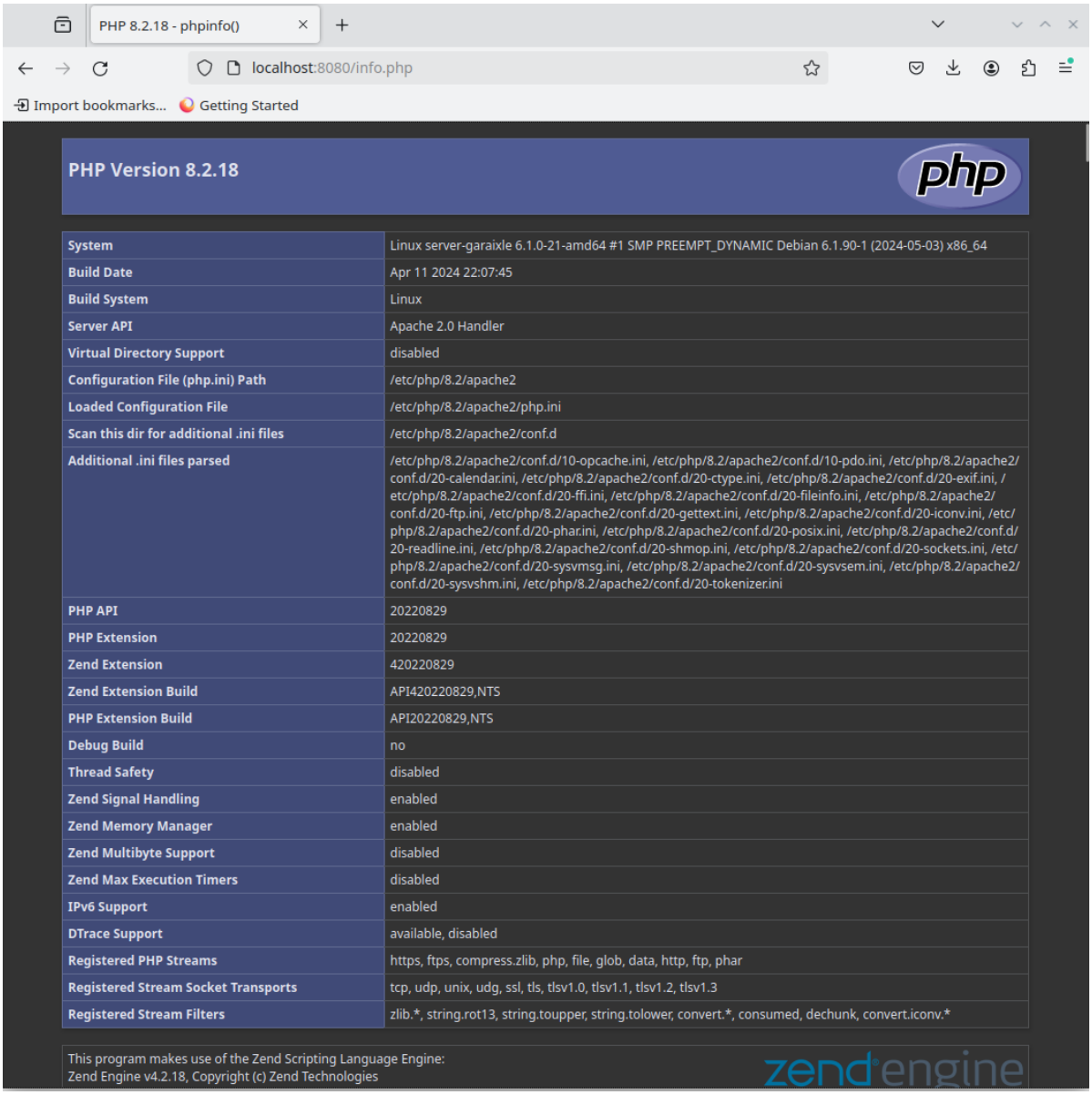
For more information, you can read the PHP documentation:
https://www.php.net/manual/en/install.unix.php
https://www.php.net/manual/en/install.unix.debian.php

1) Installation

- Connect to the VM with your root account.
- Start the installation:
  ```
  # apt install php-common libapache2-mod-php php-cli
  ```
- Apache must be restarted:
  ```
  # service apache2 restart
  ```

2) Testing the installation

- Start Apache2:
  ```
  # service apache2 start
  ```
- Navigate to the HTML folder:
  ```
  # cd /var/www/html/
  ```
- Create a PHP file named “info.php”:
  ```
  # nano info.php
  ```
- Add the following contents:
  ```
  <?php
  phpinfo();
  phpinfo(INFO_MODULES);
  ?>
  ```
- Access this URL from the host machine:
  http://localhost:8080/info.php
  → This will display a page with the principal characteristics of your PHP installation.



22. Characteristics of the PHP installation

Chapter 5: Installing PhpPgAdmin

1) Installation

To find a package in apt:

- `#apt list | grep 'keyword'`
  → You can try using the keyword '`phppgadmin`'

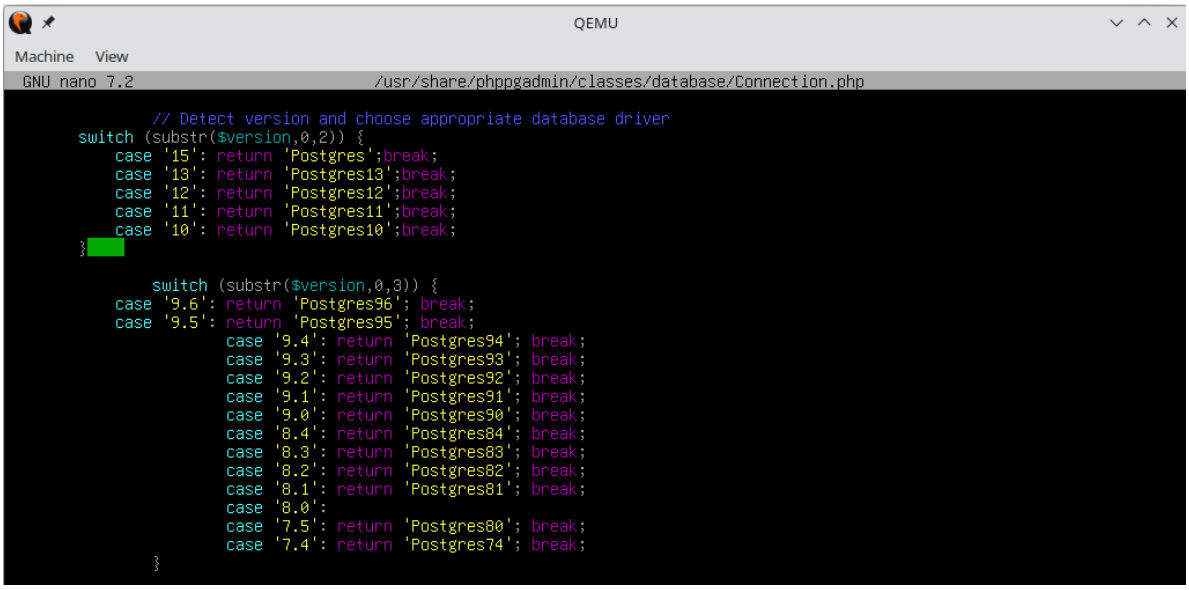
To install PhpPgAdmin:

- `# apt install phppgadmin`

2) Configuring the web interface

To allow access to the PhpPgAdmin web interface:

- Open the connection.php file:
  `# nano /usr/share/phppgadmin/classes/database/Connection.php`
- Find the line:
  `case '14': return 'Postgres';break;`
- Modify "14" to "15":
  `case '15': return 'Postgres';break;`

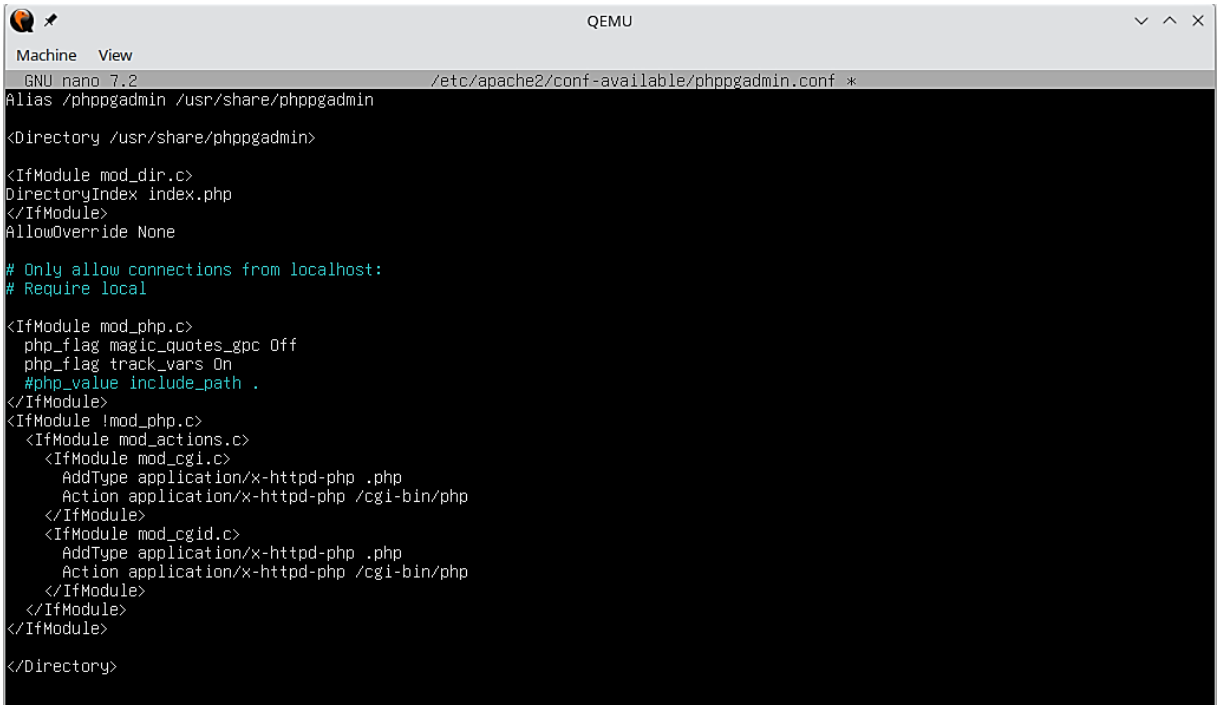


23. After the modification of the Connection.php file

You can now use the interface to visualize and edit your tables. Before accessing it through your browser, a few adjustments are needed:

- Open the phppgadmin.conf file :
  `# nano /etc/apache2/conf-available/phppgadmin.conf`

- Comment out the line "`Require local`" by adding a "`#`" at the beginning of the line.

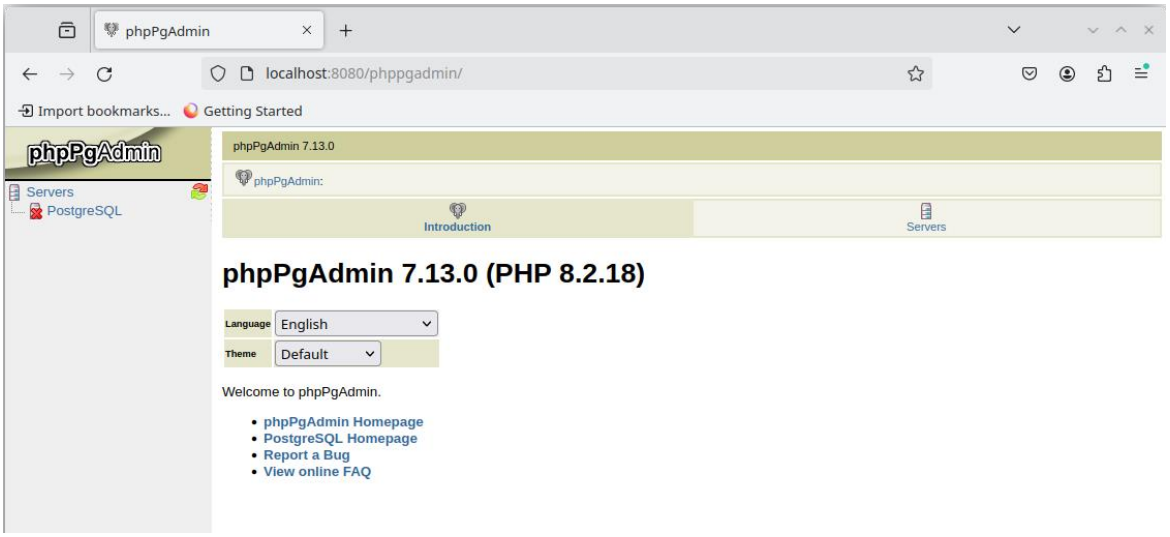


24. After the modification of the phppgadmin.conf file

- Restart Apache2:
  `# systemctl restart apache2`

Now, you can connect to the web interface!

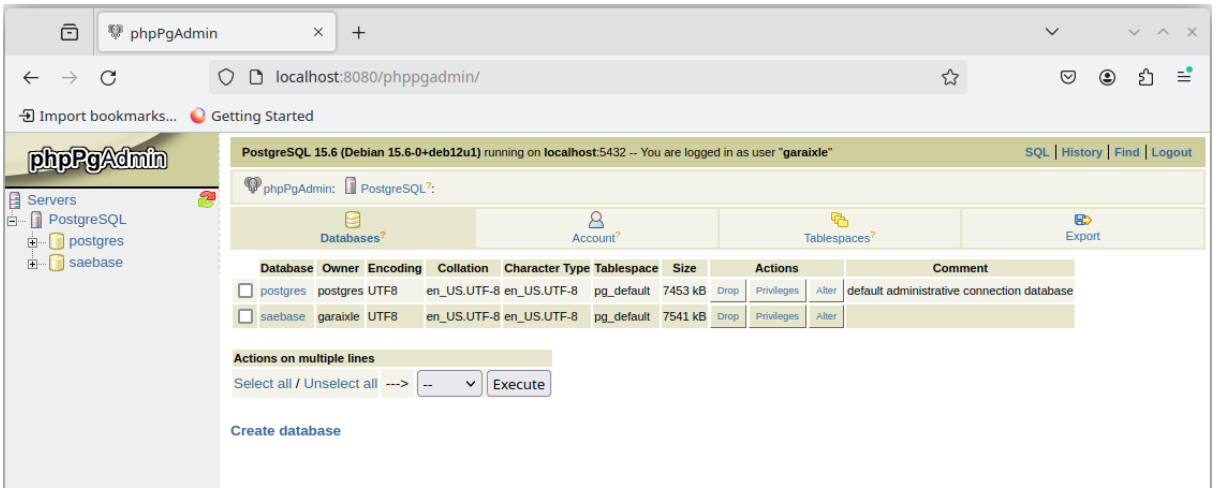
- In your browser, enter the URL:
  `localhost:8080/phppgadmin/`
- Click on "PostgreSQL" below "Servers" on the left side.



25. Web interface of PhpPgAdmin

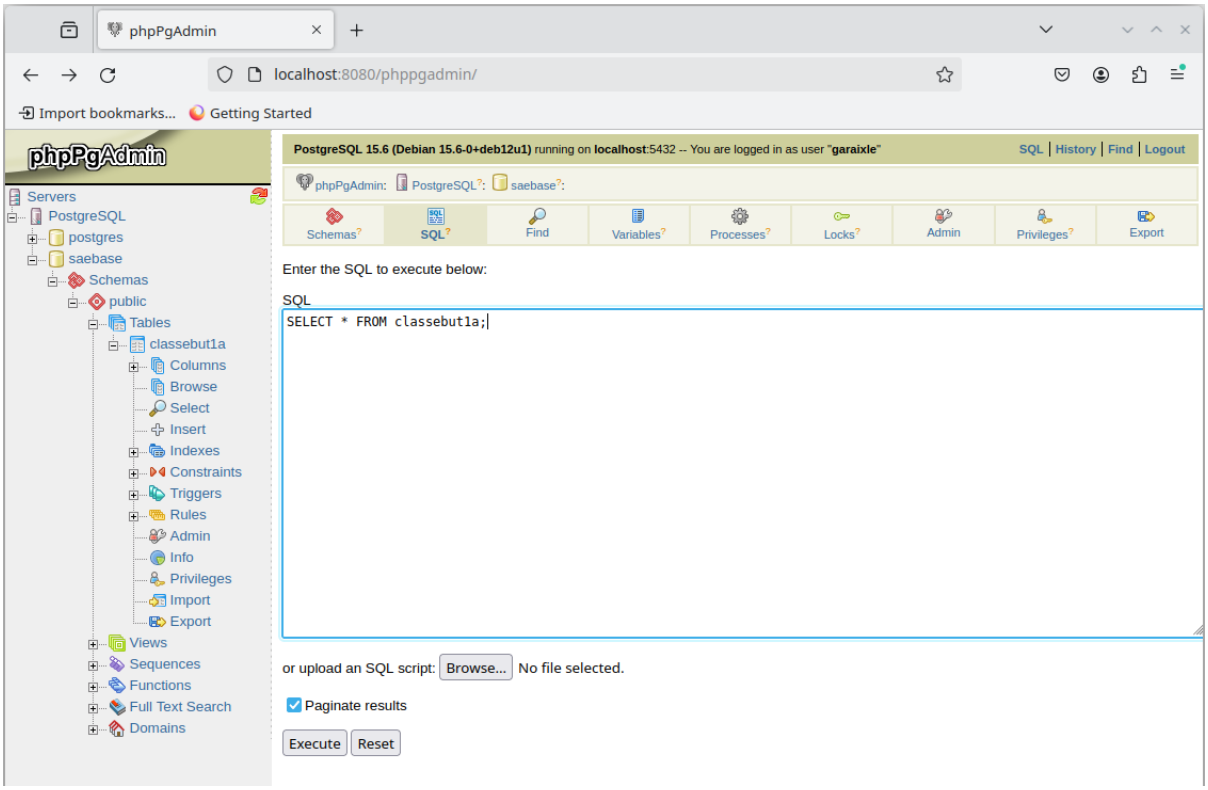
- Connect with your user named after your UGA login.

Now, you have access to the databases.



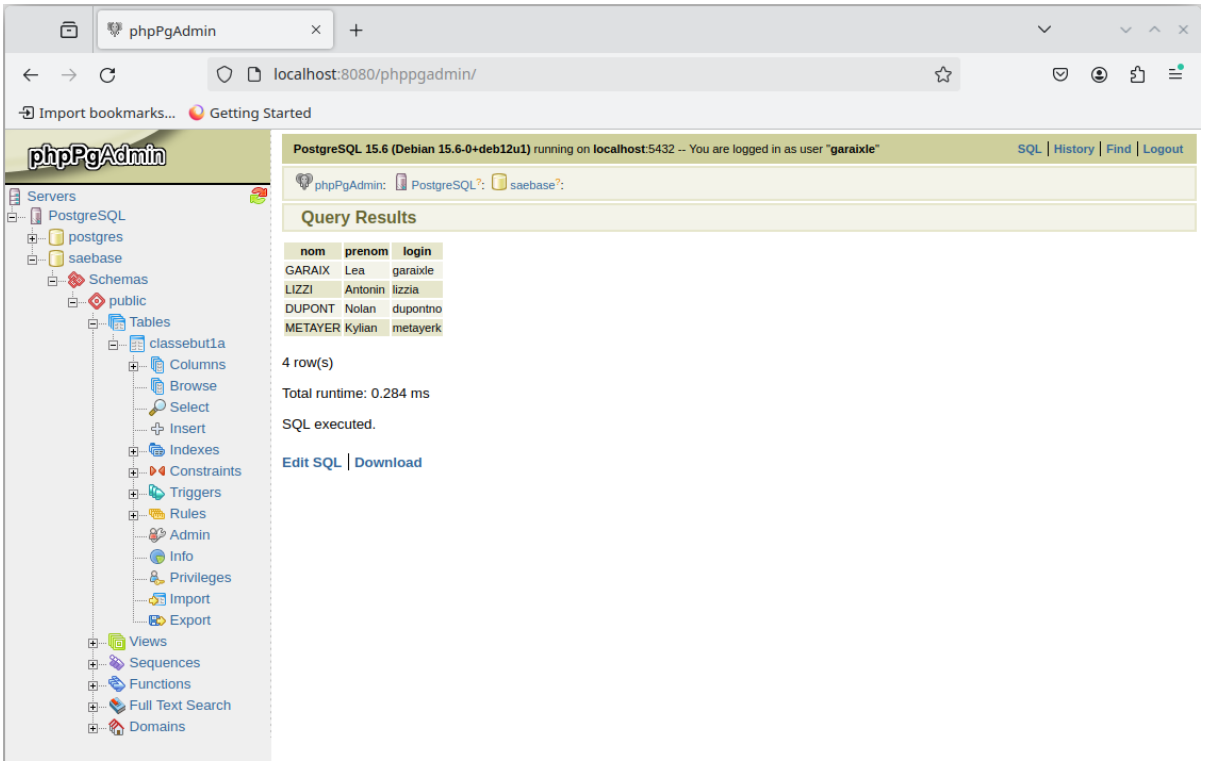
26. Databases on the PostgreSQL server

You can write queries and modify the databases. For example, click on the “saebase” database, then click on “SQL” in the menu at the top. You can write your query in the frame.



27. Interrogation from PhpPgAdmin (Screenshot 10)

And here are the results:



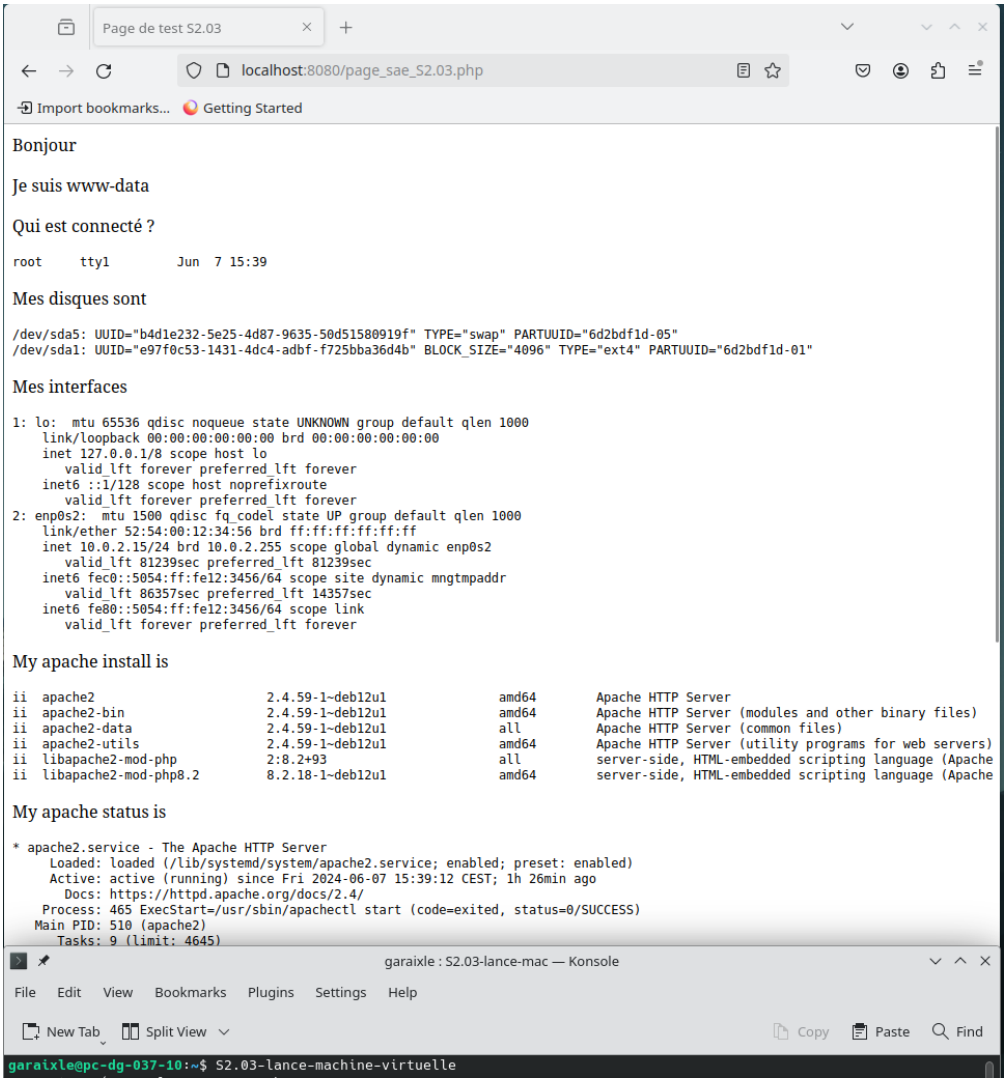
28. Results of the interrogation

Chapter 6: Finalizing the setup

1) PHP file for VM information

`/users/info/www/intranet/enseignements/S2.03/page_sae_S2.03.php` is a PHP file that gathers information about your VM installation. We will copy it to the folder `var/www/html`, and you will be able to see it through a browser on your host station.

- Copy the PHP file to your VM. On your VM, run the following command:
  `# scp -Crp garaixle@transit.iut2.univ-grenoble-alpes.fr:/users/info/www/intranet/enseignements/S2.03/page_sae_S2.03.php var/www/html`
- Confirm the connection: answer “yes” when it asks you if you want to continue connecting.
- Open the PHP file in a browser. On your host station, enter this URL:
  `localhost:8080/page_sae_S2.03.php`

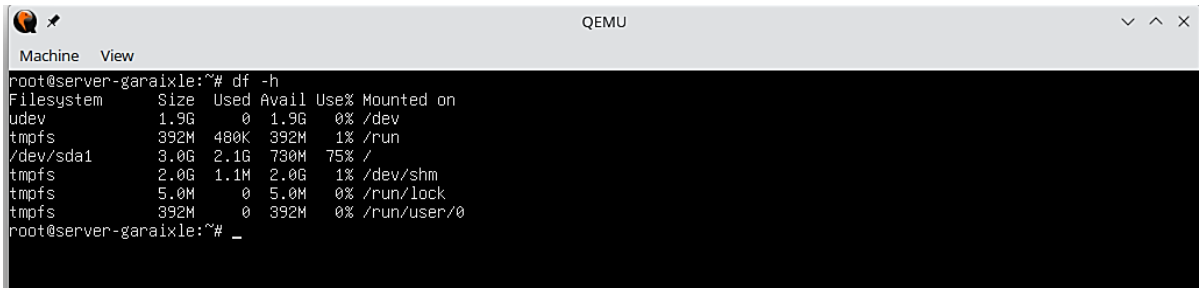


29. PHP file interrogation results (Screenshot 11)

2) Verifying disk usage

To ensure your virtual machine is running efficiently, it's important to check the used and available space on your system.

- Check disk space in your VM:
  `# df -h`
  → The -h ask for a human-readable format.



30. Disk space (Screenshot 12)

3) Enhancing security

One of the final steps in setting up your VM is to enhance its security.

Update your system

To secure the entirety of the Debian system, the first step is to **update and upgrade regularly.**

- `# apt update`
  → Updates the available packages list (not the packages themselves).
- `# apt upgrade`
  → Updates the system by installing and updating the packages.
- `# apt clean`
  → Deletes the APT cache.
- `# apt autopurge`
  → Automatically deletes the unused dependencies and their configuration files.

Secure SSH access

Root access and SSH access can be source of vulnerabilities in security. The root account can access your entire system. This wiki proposes a few solutions:
https://fr-wiki.ikoula.com/fr/S%C3%A9curiser_sa_machine_Debian

For example, you could change the default SSH port, disable the root login (and only use a standard user with sudo privileges) or use SSH keys for authentication.

Setup a firewall

It helps to control incoming and outgoing traffic.

4) Securing Apache, PHP and PostgreSQL

Apache, as a web server, has its own vulnerabilities. OpenClassrooms offers a comprehensive course on securing Apache, which you can access here: https://openclassrooms.com/fr/courses/1733551-gerez-votre-serveur-linux-et-ses-services/5236056-securisez-votre-serveur-web

For improving PHP security, you can refer to this resource that outlines various techniques: https://hidora.io/ressources/10-facons-dameliorer-la-securite-de-votre-php/

In PostgreSQL, implementing role-based access control ensures that users have only the necessary privileges for their tasks. This minimizes the risk of unauthorized access.

Annexe

The virtual machine launching script

To launch the virtual machine, we used the following command:
`$ S2.03-lance-installation`

This executes a script containing a lengthy command, which initiates QEMU/KVM with all the necessary parameters:

```
$ 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
format=raw,file=/donnees/TP-infobut1/Debian-S2.03-
garaixle.img,discard=unmap -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
/usr/local/images-ISO/debian-12.5.0-amd64-netinst.iso
```

Let's see step-by-step what they mean:

`qemu-system-x86_64`
This command starts the QEMU emulator for x86_64 architecture.

`-machine q35`
This option specifies the machine type. Here, it is q35.

`-cpu host`
It tells QEMU to emulate the CPU as closely as possible to the host CPU.

`-m 4G`
It sets the amount of RAM allocated to the VM to 4GB.

`-enable-kvm`
It enables Kernel-based Virtual Machine.

`-device VGA,xres=1024,yres=768`
It specifies the graphics device for the virtual machine, with a resolution of 1024x768.

`-display gtk,zoom-to-fit=off`
It specifies the display settings for the VM.

`-drive format=raw,file=/donnees/TP-infobut1/Debian-S2.03-garaixle.img,discard=unmap`
It defines a virtual disk drive. The disk image is in raw format, and the path to access it is specified in the "file=" section.

`-device e1000,netdev=net0`

It adds a network device emulating an Intel E1000 network adapter to the VM. It specifies that it is connected to a network device named “net0”.

`-netdev user,id=net0,hostfwd=tcp::2222-:22,hostfwd=tcp::4443-:443,hostfwd=tcp::8080-:80,hostfwd=tcp::5432-:5432`

It defines a user-mode network device (“net0”). The port forwardings are set up here.

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

It specifies the path to the ISO image file used as the virtual CD-ROM drive.

Sources

These URL will guide you to the websites I consulted while writing this manual. I also consulted the R2.06 lessons to install PostgreSQL.

https://www.debian.org/intro/about

https://en.wikipedia.org/wiki/Debian

https://wiki.qemu.org/Main_Page

https://wiki.qemu.org/Features/KVM

https://www.ionos.com/digitalguide/server/know-how/what-is-kvm/

https://httpd.apache.org/

https://www.techtarget.com/whatis/definition/Web-server

https://www.postgresql.org/about/

https://en.wikipedia.org/wiki/PostgreSQL

https://en.wikipedia.org/wiki/Optical_disc_image

https://fr.wikipedia.org/wiki/SHA-2

https://help.ubuntu.com/kubuntu/desktopguide/fr/root-and-sudo.html

https://doc.ubuntu-fr.org/xorg

https://fr.wikipedia.org/wiki/Paquet_(logiciel)

https://doc.ubuntu-fr.org/apt-cli

https://www.linuxtricks.fr/wiki/fstab-explications-sur-le-fichier-et-sa-structure

https://fr-wiki.ikoula.com/fr/S%C3%A9curiser_sa_machine_Debian