How to install a Qemu/KVM virtual machine ?

Author

Chaboud Adèline

Summary

1 - Debian system installation

Welcome to this installation guide. In this guide, we will learn how to install Debian version 12.x for x86 64-bit processors using a “netinst” ISO image.

Firstly, we need to download the ISO file. To do this, go to the website https://cdimage.debian.org/cdimage/release/current/amd64/iso-cd/. On this website, you can download the ISO file and view the ISO image. After downloading the ISO file, we need to verify the ISO image. For this, using a terminal, navigate to the directory where you downloaded your ISO file and execute the command ‘sha512sum FILE_NAME’.

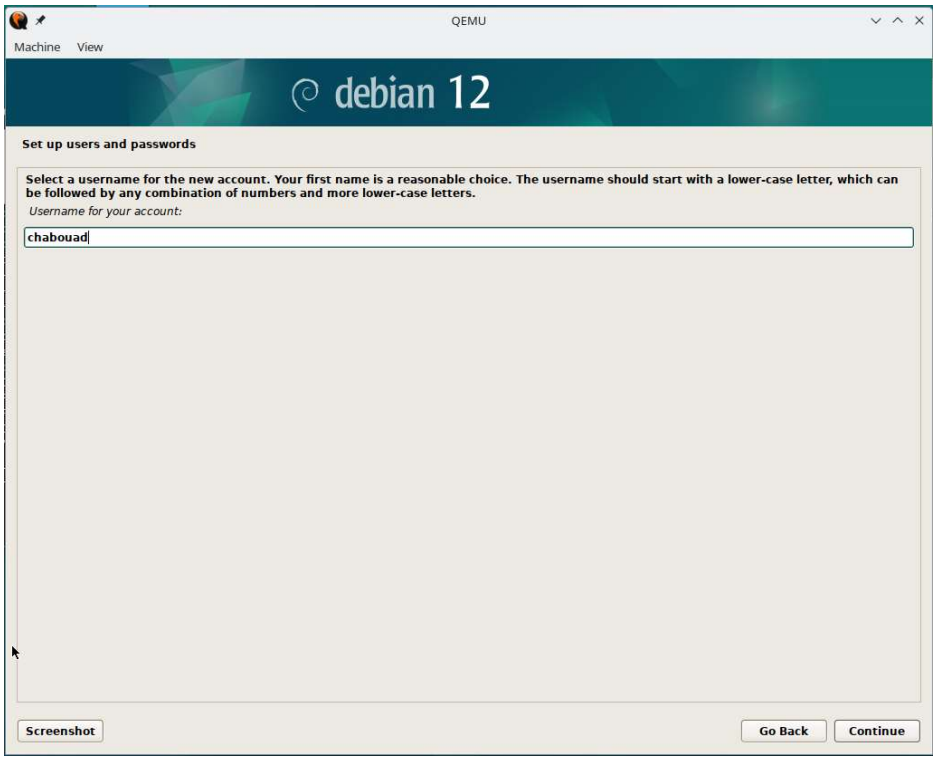
Now, we can start the installation of the Debian system. Use the command ‘S2.03-lance-installation’ to start the virtual machine with the ISO image.
To know the settings used to start Qemu/KVM, we need to read the script ‘S2.03-commun’. To read this document, you can use the command ‘nano’.

```
# Commande lancement Qemu
lance_qemu="qemu-system-x86_64 -machine q35 -cpu host -m 4G -enable-kvm -device VGA,xres=1024,yres=7
68 -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"
~
```

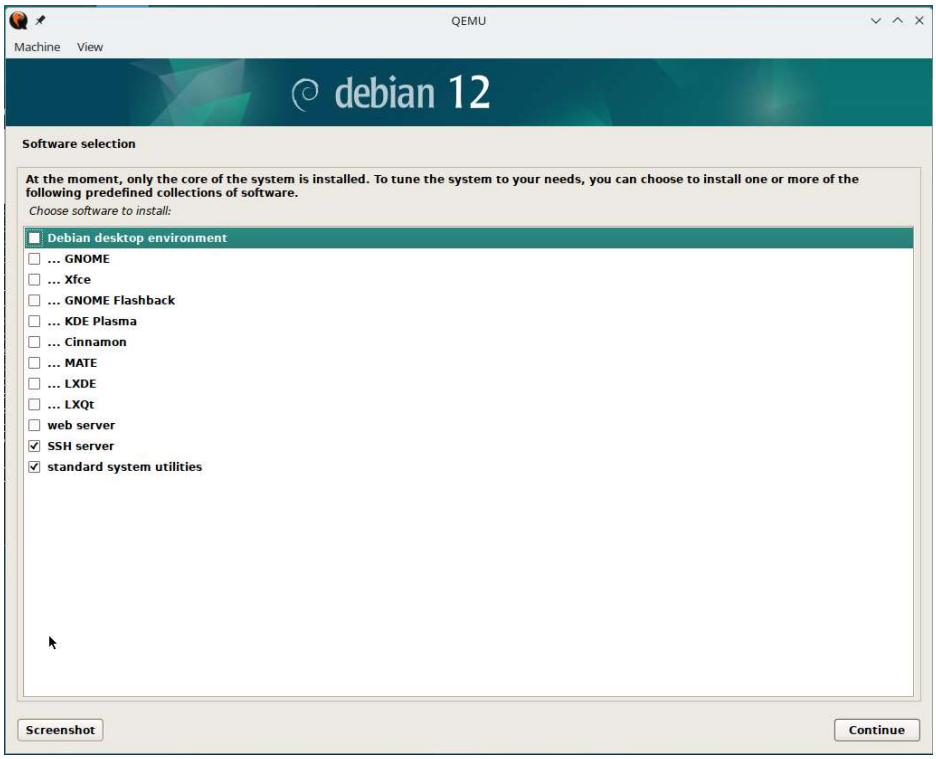
‘qemu-system-x86_64’ is the launch command for an x86_64 system.
‘-machine q35’ is a setting that specifies the virtual machine's type; here, it's a q35 type.
‘-cpu host’ is a setting that specifies which CPU QEMU must use; here, it uses the same CPU as the host.
‘-m 4G’ is the storage size for the RAM of the virtual machine; for this installation, it's 4 GB.
‘-enable-kvm’ is a setting that activates hardware acceleration.
‘-device VGA,xres=1024,yres=768’ is a setting that configures a graphical peripheral with a resolution of 1024x768 pixels.
‘-display gtk,zoom-to-fit=off’ is a setting that specifies the backend system used for the graphical display; here, it's GTK. This command also specifies that the zoom option is deactivated.
‘-drive $drive’ is a setting that specifies the hard disk or the disk image that the virtual machine has to use; ‘$drive’ is probably a variable that contains the path to the disk image.
‘-device e1000,netdev=net0’ adds a network peripheral of the e1000 type. The part after the comma combines this network peripheral with a virtual network, here it's ‘net0’.
‘-netdev user,id=net0’ is a setting where the command configures the user's network with the ‘net0’ ID.
‘hostfwd=tcp::2222-:22’ This part of the command redirects port 2222 of the host to port 22 of the virtual machine; this port is used for SSH.
‘hostfwd=tcp::4443-:443’ With this part, port 4443 of the host is redirected to port 443 of the virtual machine; this port is used for HTTPS.
‘hostfwd=tcp::8080-:80’ With this part, we redirect port 8080 of the host to port 80 of the virtual machine; this port is used for HTTP.
‘hostfwd=tcp::5432-:5432’ This last part of the command is used to redirect port 5432 of the host to port 5432 of the virtual machine; this port is used for PostgreSQL.

Now, it's time to configure your virtual machine. Here are the principal stages:
Firstly, leave the virtual machine in English. Change the location; for this, go to other/Europe/France. Leave the locales with United States, en_US.UTF-8. Change the keyboard to

French. For the Hostname, use this format: server-“YOUR_UGA_LOGIN”. After this, you need to choose a root password; a recommended password is “root”. For the User Account – Full Name, enter your complete name. For your User Name, enter your UGA's login, like this:

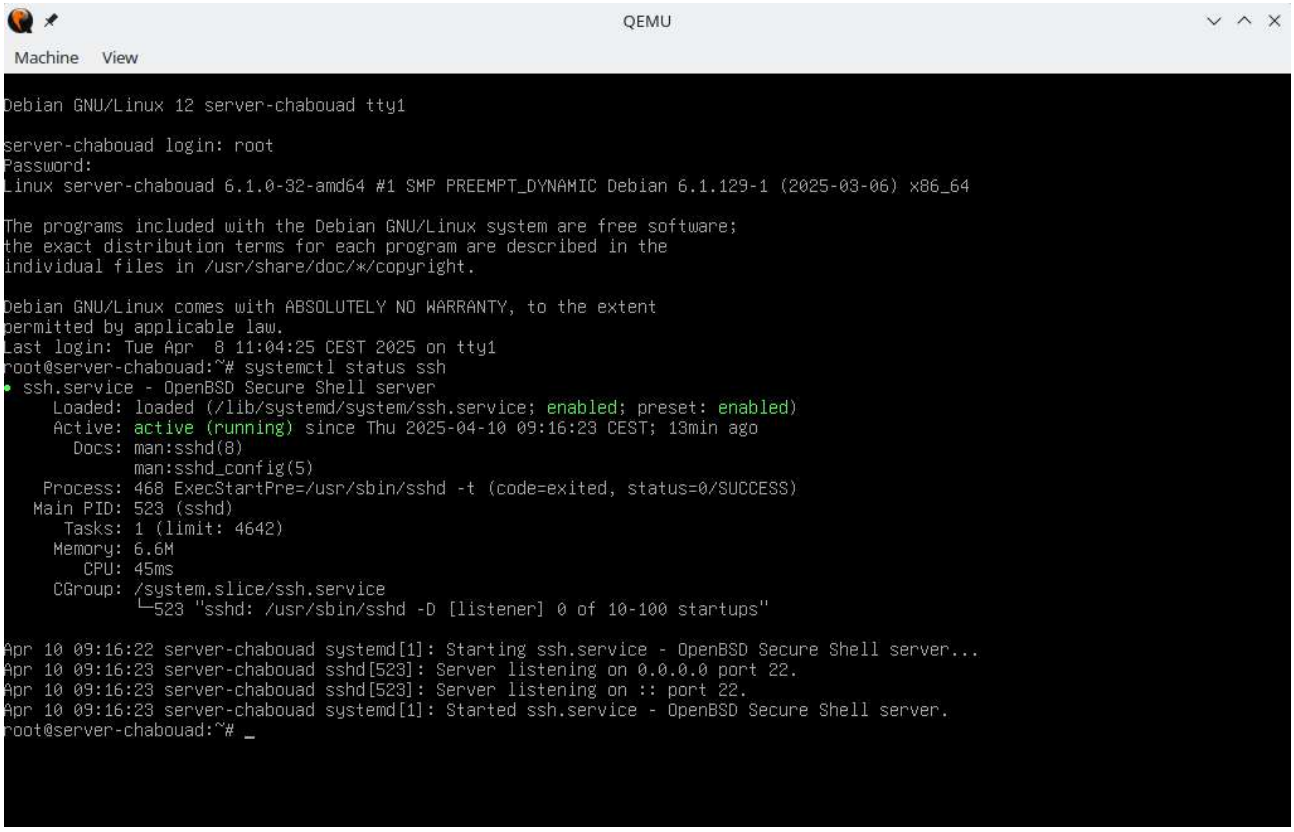


Now, choose a password for your user’s account; you can use the password “etu”. Regarding the disk partitioning, leave “Guided - use entire disk” and “All files in one partition”, but change the radio button from “No” to “Yes”. In the Software Selection, verify that “Debian desktop environment” is not checked and “SSH server” is checked. Leave the button “Yes” checked for “Install GRUB boot loader”, like this:



Finally, for the Device for boot loader, choose ‘/dev/sda’.
Now, the installation is finished, and the virtual machine will restart. For the next step, we need to turn off the virtual machine. To do this, log into the root account using the root password that you defined during the configuration. When you are logged into the root account, execute the command “systemctl poweroff” to cleanly your virtual machine.
The next step is to move your disk image from the local disk of your Linux station to the erebus4 server. This step will allow you to use your virtual machine more easily the next time. Before the transfer, verify that your virtual machine is turned off.
After, in your Linux station’s terminal, execute the command ‘S2.03-déplace-image-disque-sur-erebus4’ to move your virtual machine’s image to the erebus4 server. Now, we have finished the installation, and we need to test if the virtual machine is functional. To do this, turn on the virtual machine with the command ‘S2.03-lance-machine-virtuelle’.

When you are logged into your account, using the user name and the password that you defined during the configuration, you can see the Ethernet and IP configuration of your virtual machine with the command ‘ip addr’. We need to verify if we can communicate with the outside world. To do this use the ‘ping’ command and try to communicate with your Linux station.
We also need to verify if the Xorg server is not present. To do this use the command ‘dpkg -l | grep xorg’. The Xorg server is the server that produce a graphical interface. Here, we want to install a virtual machine with only a command-line, which is the reason why we don’t need to have the Xorg server.
Now, we verify if SSH access is functional. To verify if SSH is activated in your virtual machine, execute the command ‘systemctl status SSH’. We need to obtain a reply like this:



After this, we will make a connection from the Linux station to the virtual machine. In the terminal, execute the command ‘ssh LOGIN_NAME@localhost -p 2222’. Now, go to the root account with the command ‘su -’. Here, try to install the text editor ‘micro’ with the command ‘apt install micro’

Now, we will execute the command ‘cat /etc/fstab’. This command show the contents of the file ‘/etc/fstab’, which is an important configuration file for Unix-like operating systems. This file includes information about file system that need to be automatically mounted at system startup. Here is an example:

2 - Install and configure applications

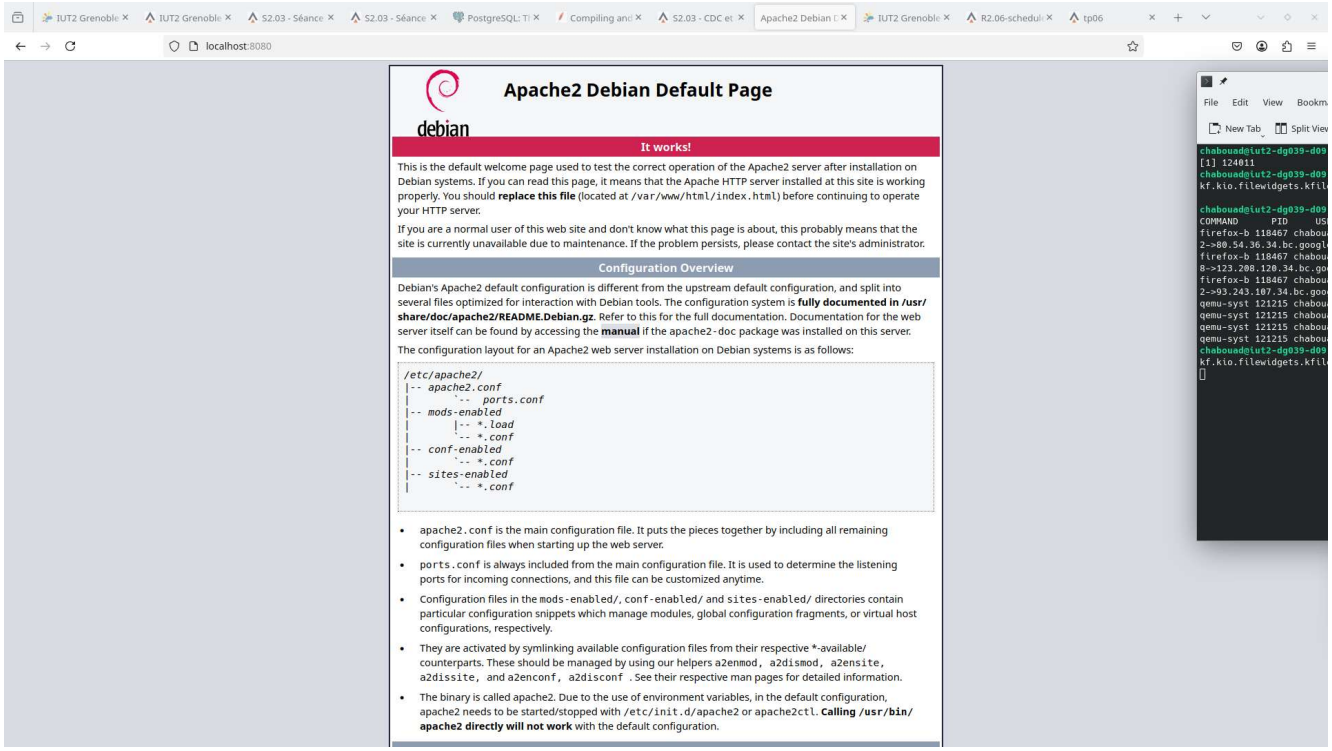
◦ Apache installation

In this part, we will install the Apache application. With Apache, we can create an HTTP server on your virtual machine.

Firstly, we need to be connected on the root account of your virtual machine. To install applications, we use ‘apt’. To install Apache, execute the command ‘apt install apache2’ in the root account of your virtual machine. Now, that you have Apache installed in your virtual machine, to use it, we need to start the application. To do this, execute the command ‘service apache2 start’. After this, verify if your Apache server is running with the command ‘systemctl status apache2’. You should obtain this result:

If you don’t have this result, start Apache with this command: ‘systemctl start apache2’.

Test a connection to the Apache server in your virtual machine. We will use the ‘telnet’ command because your virtual machine cannot display a graphical interface. Firstly, execute the command ‘telnet localhost 80’ in your user account. With this command we establish a connection to the server. Now, enter this request: ‘HEAD / HTTP/1.0’ and make two line breaks. If the installation is successful, the server will answer ‘HTTP/1.1 200 OK’ to your request.
During the installation, we saw that the port 8080 of your Linux station is redirected to the port 80 of your virtual machine for the HTTP connections. In the web browser of your Linux station search for http://localhost:8080. This web address will take you to the default page of your Apache server. It should look like this:



Now your Apache server is ready.

◦ PostgreSQL installation

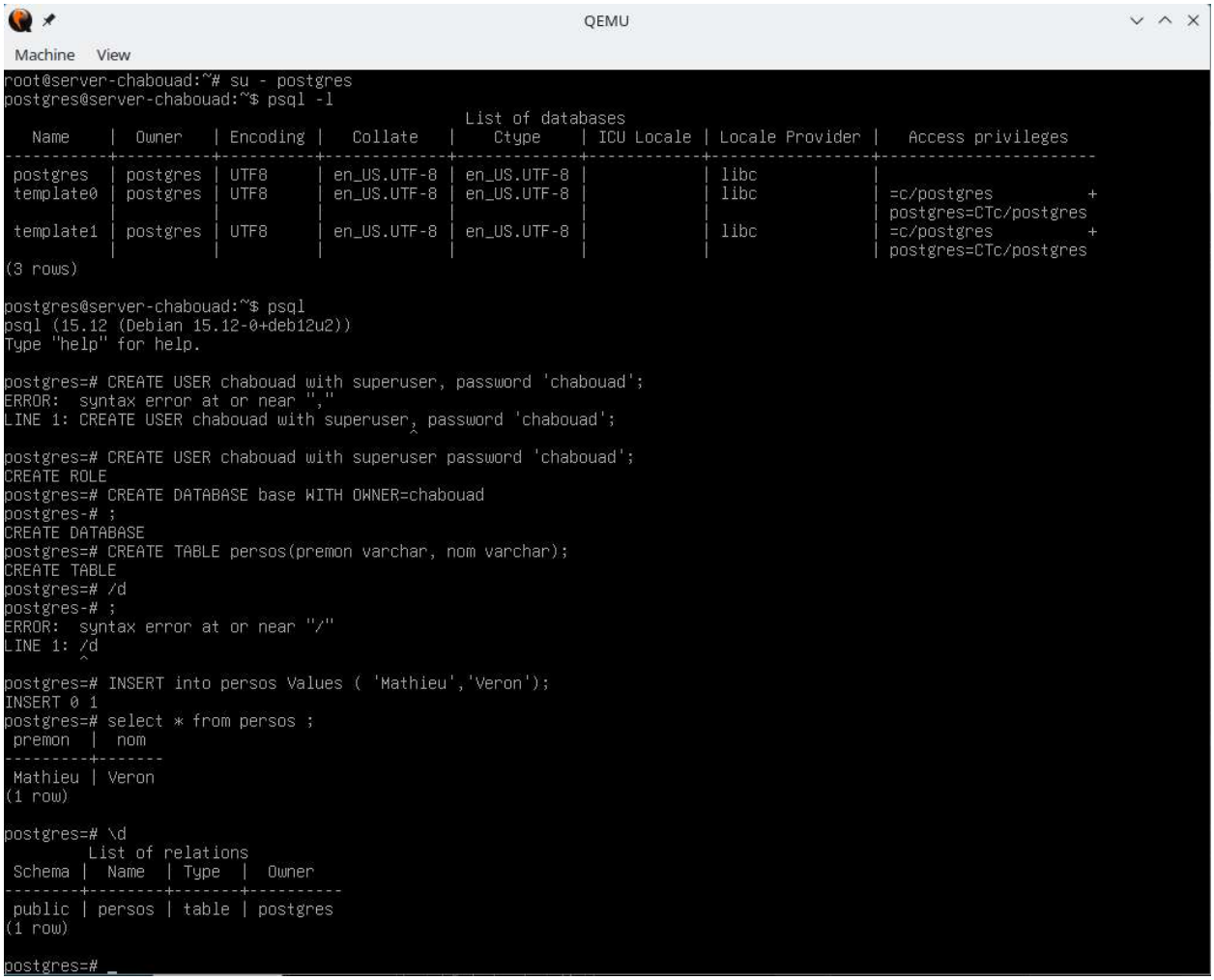
To install PostgreSQL, log in to the root account. Once logged in, execute the command ‘apt install postgresql’. Now, check if the installation was successful using the status command of systemctl. You should see this answer:

```
root@server-chabouad:~# systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; preset: enabled)
     Active: active (exited) since Fri 2025-03-28 14:14:30 CET; 4min 48s ago
   Main PID: 2768 (code=exited, status=0/SUCCESS)
        CPU: 506us

Mar 28 14:14:30 server-chabouad systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
Mar 28 14:14:30 server-chabouad systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
root@server-chabouad:~#
```

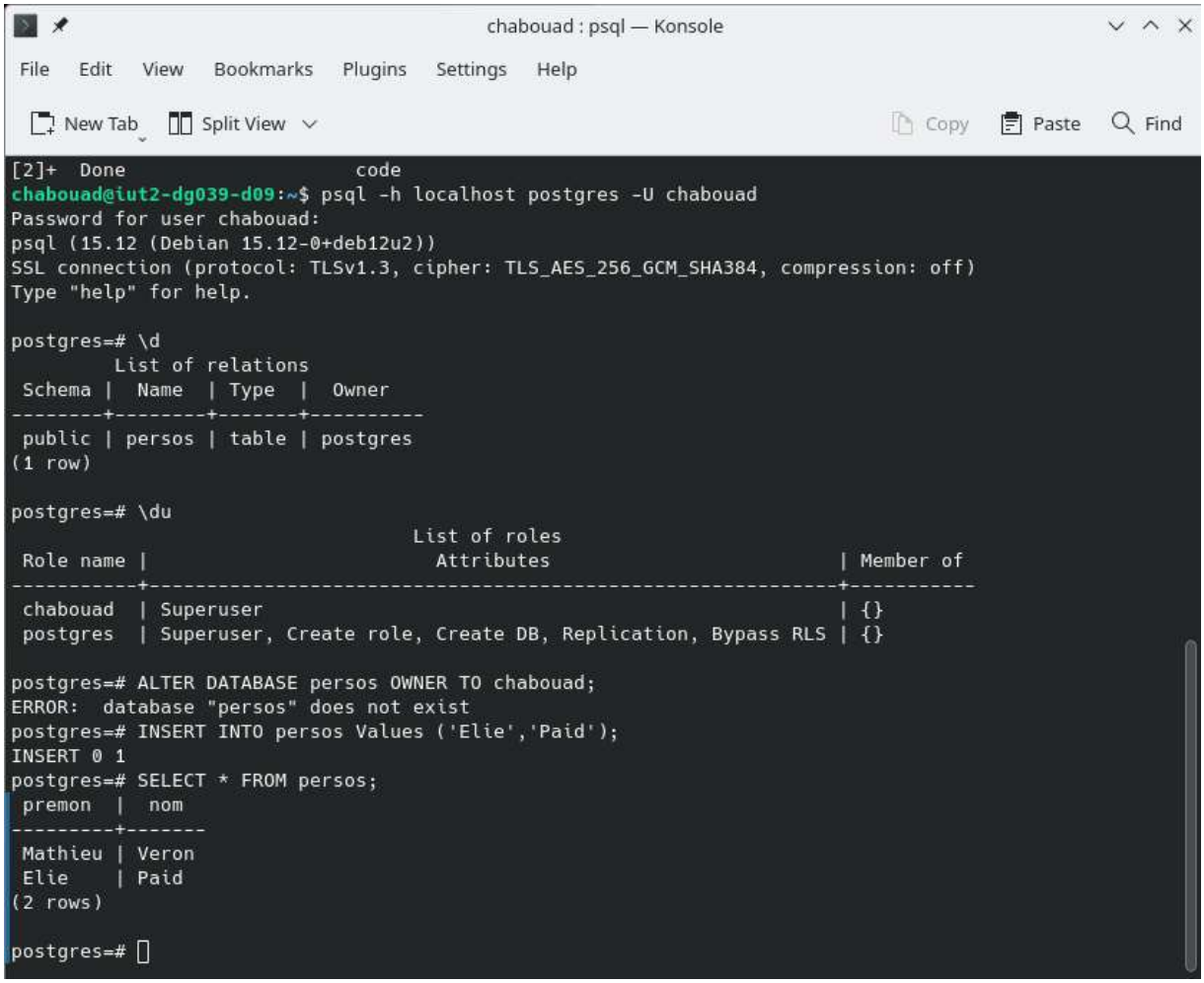
To verify if the installation is ready, try to connect you from your virtual machine using the postgres login. From the root shell, execute the command ‘su – postgres’ to connect to the Unix account

postgres. To know the default database use the command ‘psql -l’. Now we can log in PostgreSQL with the command ‘psql’. In PostgreSQL we can create new users. Create a new user with your UGA’s login by using the command ‘CREATE USER *login* WITH superuser password ’temporary_password’;’. We will also create a new database and a new table. To create a new database execute the command ‘CREATE DATABASE *base_name* WITH OWNER=*login*;’ and to create a new table execute the command ‘CREATE TABLE *table_name* (*column type, column type,* …);’. Now, we can insert some data in your table with the command ‘INSERT INTO *table_name* VALUES (*‘data’, ‘data’*,...);’. Here is an example:

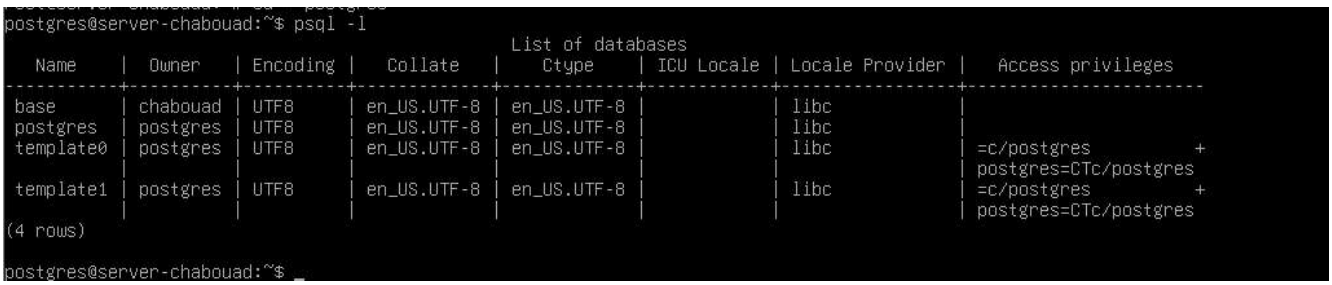


We can use PostgreSQL on your virtual machine, but now we want to be able to connect to PostgreSQL from your Linux station to your virtual machine. To do this, we need to modify PostgreSQL’s settings.
Firstly, return to the Unix account postgres. Here use the nano application to edit the configuration file by executing this command ‘nano /etc/postgresql/15/main/postgresql.conf’. You need to find the section “CONNECTIONS AND AUTHENTICATION”. You can use the command Ctrl-w to search within the file. In that section, you have to uncomment and modify a line to “listen_addresses = '*'”. After your modification, save the file with the command Ctrl-o and exit with the command Ctrl-x. Now, we will change the authentication rules file. To modify it, execute this command ‘nano /etc/postgresql/15/main/pg_hba.conf’. In this file, you have to add the line:
“#IPv4 remote connections:
host all all 0.0.0.0/0 scram-sha-256”

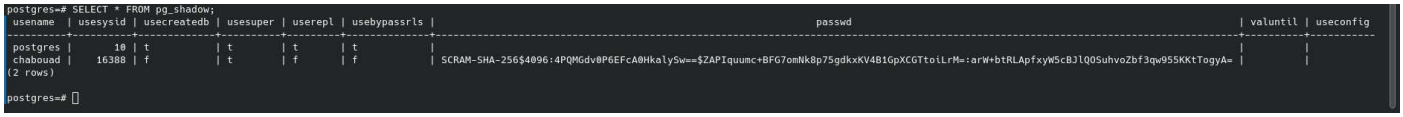
After this modification, go back to the root account to restart the PostgreSQL server. To do this, execute the command ‘service postgresql restart’. Now you can connect to your PostgreSQL server from the terminal of your Linux station. To make a connection, execute the command ‘psql -h localhost -U postgres *login’*. When you are connected, you can change your password with the command ‘\password’. You can continue to modify your database.



If you show the database now, we can see that there are a new database with your login as the owner. It’s the database that’s you created.



Finally, we can verify if the passwords were hashed with the SHA-256 hashing function through the ‘pg-shadow’ table. To see what this table contains use the command “SELECT * FROM pg-shadow;”.



In this example, the password was hashed with the hashing SHA-256 function.

○ PHP installation

Now let’s install PHP. Firstly, return to the root shell. If you want, you can show information about the package with the command ‘apt show *package_name’*. Here, the package name is ‘php’. In order to install PHP, we use the command ‘apt install *package_name’*. Now PHP is installed on your virtual machine. The next step is to verify if this installation is correct. To do this, we will save a new PHP file that we can view through the web browser of your Linux station. First, navigate to the folder ‘/var/www/html/’. When you are in this folder, create a new file ‘info.php’. To do this you can use the command “nano info.php”, which will create and open the new file. In the new file, you need to write this:

“

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

“

Save and exit the file. Now, go to the web browser of your Linux station and search http://localhost:8080/info.php. If the installation is successful, a web page displaying the principal feature of your PHP installation should appear.

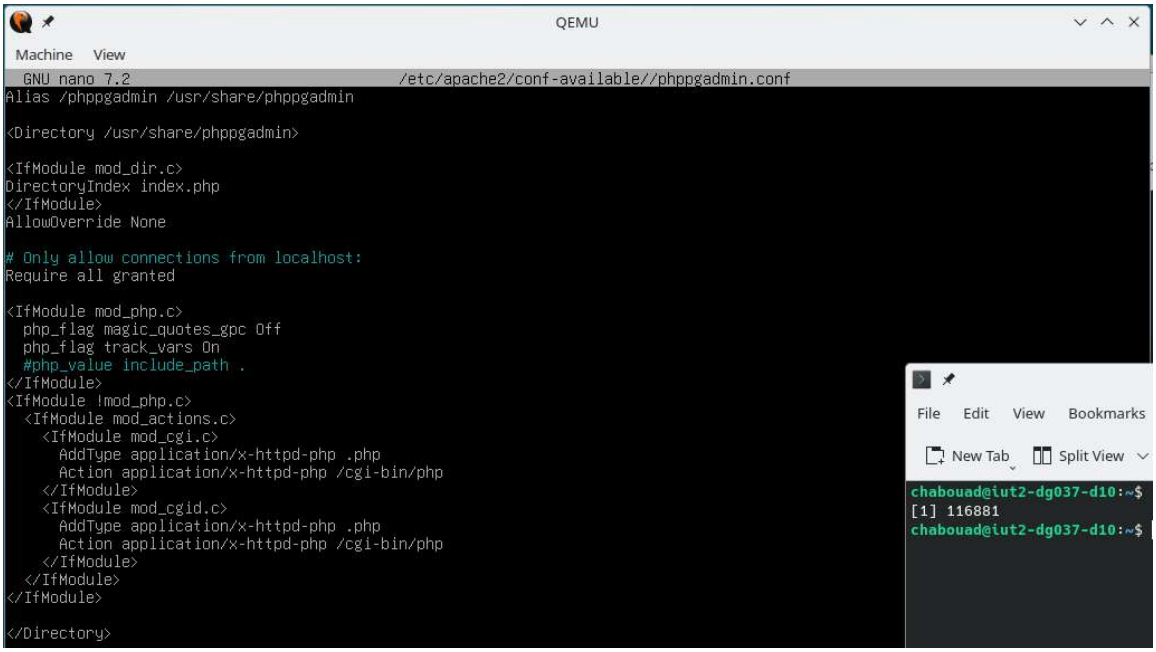
○ PhpPgAdmin installation

Now, let’s proceed with the final installation: PhpPgAdmin. To install PhpPgAdmin, we need to add a new source to apt. To do this, navigate to the folder ‘/etc/apt/’ and open the file ‘sources.list’ with nano. In this file you have to add this line:

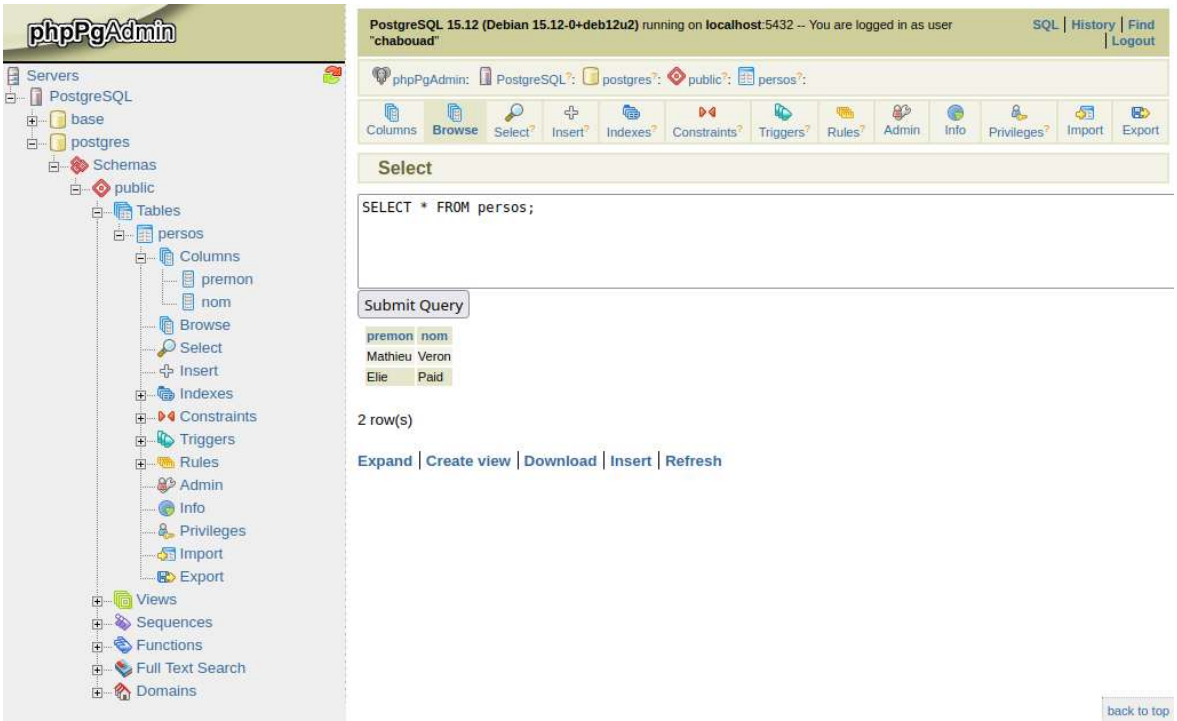
deb http://deb.debian.org/debian bookworm-backports main

Save and exit this file. Before starting the installation, run the command ‘apt update’. Now we can install PhpPgAdmin using the command ‘apt install phpphadmin/bookworm-backports’. After the installation, go to the folder ‘/usr/share/doc/phppgadmin’ and open the file ‘README.Debian’. You need to read this file. After this, we have to modify a configuration file. Go to the folder

‘/etc/apache2/conf-available’ and open the file ‘phppgadmin.conf’ with nano. You have to change the line after “# Only allow connections from localhost:”. In the end, your file should look like this:



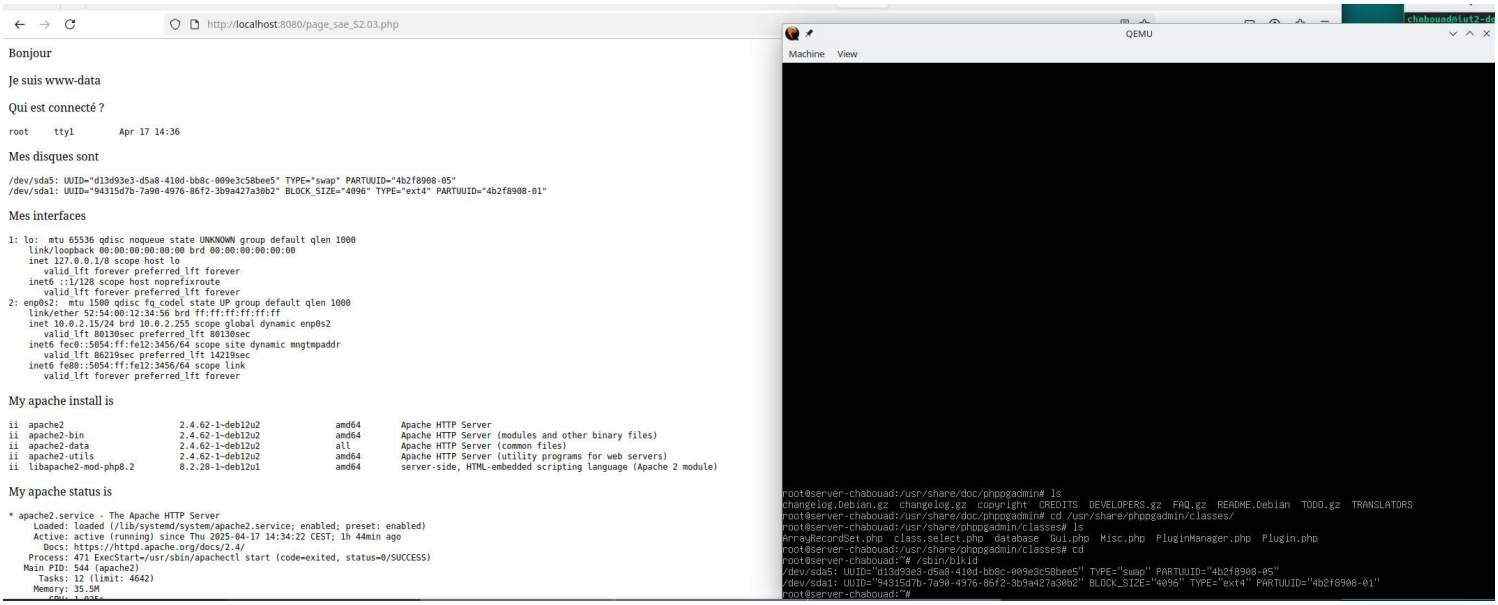
Now everyone can access your PhpPgAdmin web interface. We have to restart Apache. Execute this command: ‘systemctl restart apache2’. Now, all the modifications we have made are recognized by PhpPgAdmin. You can verify if your your installation is successful by going to the web site http://localhost:8080/phppgadmin/ with the web browser of your Linux station. On this web site you can modify your database like this:



3 – Upload a new web page

Now we will upload a new web page. To do this, we need to move the file ‘page_sae_S2.03.php’ from your Linux station to the virtual machine. To copy from your Linux station to your virtual machine use the command: ‘scp -P 2222 /users/info/www/intranet/enseignements/S2.03/page_sae_S2.03.php localhost:/tmp/’. Now, the file has been copied to your virtual machine in the folder ‘/tmp/’. You have to move the file from the folder ‘/tmp/’ to the folder ‘/var/www/html’. If you go on the website http://localhost:8080/page_sae_S2,03.php, you can see the page created with the file ‘page_sae_S2.03.php’.

With the command ‘/sbin/blkid’ we can see the disk used by your virtual machine, and you can compare with the disk information on the website. Here is an example:



One last thing that you can do is to see the storage space used and the storage place available on your virtual machine after all these installations. To see this, you can use the command ‘df -h’. This is an example of what you might see:

```
root@server-chabouad:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           392M  492K  392M   1% /run
/dev/sda1       3.0G  1.6G  1.3G  57% /
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
tmpfs           392M     0  392M   0% /run/user/1000
root@server-chabouad:~# _
```

In the first row, there is the path to the file system. In the row 3 and 4 there are the used and available space, and in row 5 there is percentage of used storage space.