Flavien Riondet

Virtual machine installation guide

In this explanation, we're going to install a virtual machine with no graphical interface and several applications running on it.

1/ Installing the machine

For begin the installation we go search ISO image on internet.
(https://www.debian.org/)
Then install it using the command :
lance_qemu="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":

Explanation of the main elements of the command :

-lance_qemu=: This is a variable that stores the QEMU command.

-qemu-system-x86_64: This is the name of the QEMU executable for virtualization of x86_64 systems.

-machine q35: This specifies that QEMU must use the Q35 machine model. The Q35 machine model is an emulated hardware platform for x86 machines.

-cpu host: This tells QEMU to emulate the host machine's processor.

-m 4G: This sets the amount of RAM allocated to the virtual machine to 4 GB.

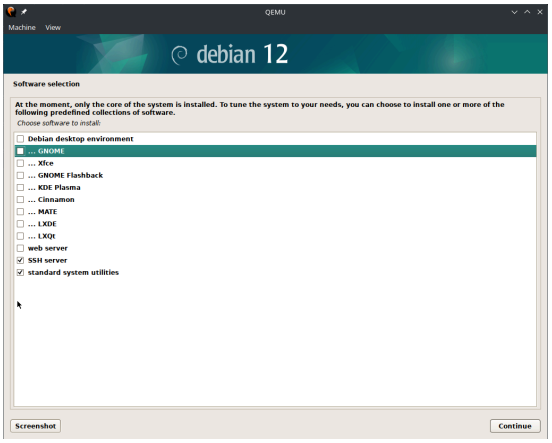
-enable-kvm: Enables KVM (Kernel-based Virtual Machine) support for hardware acceleration.

In short, this command line launches QEMU with the specified configuration to create a virtual machine with a specific screen resolution, 4 GB memory allocation, host machine CPU emulation and hardware acceleration enabled.

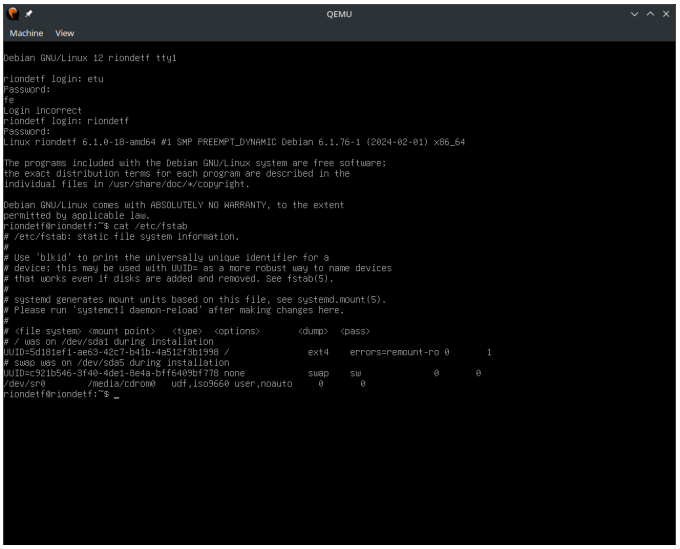
Next, you need to choose a configuration for your machine. Here is an example of a simple configuration, but feel free to do it differently :

- Language : English
- Location : other/Europe/France
- Locales : United States, en_US.UTF-8
- Keyboard : French
- Hostname: use server-“your username”.
- Root Password: a simple password is recommended, like “root”.
- User Account - Full Name: your full name, for example
- User Name: enter your user name

- User Password: enter a simple password.
- 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
- Device for boot loader : /dev/sda
-



To finish the installation, “reboot” the virtual machine by typing poweroff from the console. Your machine should look like this, with a user for me called “riondetf”.



2/ Install apache

Then, if you want to use this machine as a server, you'll need to install apache. First of all, you need to log on to the machine as superuser with the command :

```
su -
```

And type the root password you entered during configuration.
To install it, type the command :

```
apt install apache2
```

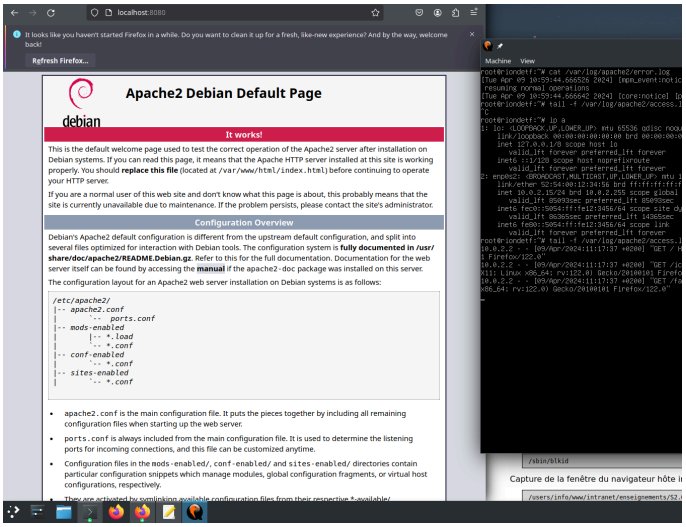
then answer each choice with “yes”. You can see the status of apache by typing systemctl status apache2 and start it with systemctl start apache2. To check that you have completed the above steps correctly, write :

```
telnet localhost 80
```

and once connected, a few seconds later write :

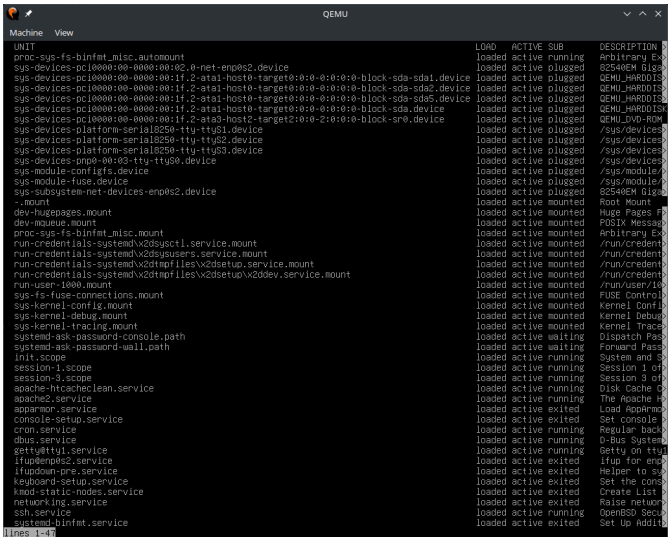
```
HEAD / HTTP/1.0
```

Remember to note the / with a space on each side, as this is to indicate the root.
If you get this response: HTTP/1.1 200 OK, then everything is working correctly.
You can now try typing http://localhost:8080 in the host machine's browser and look at the page that tells you everything is working properly.

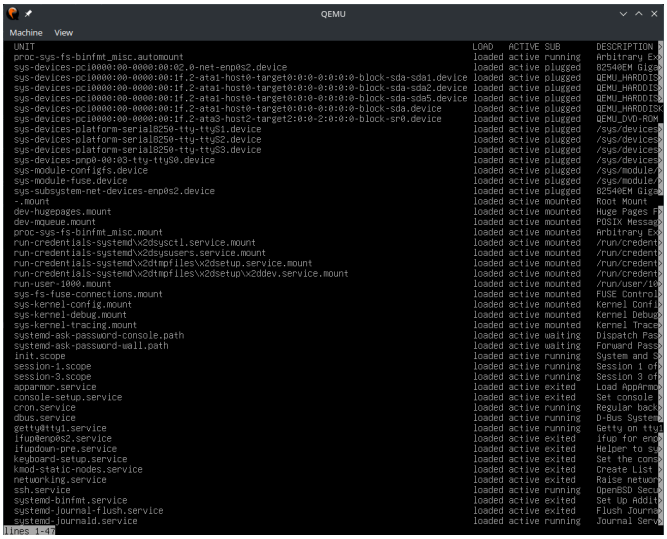


The systemctl command lets you view and manage the status of linux services.

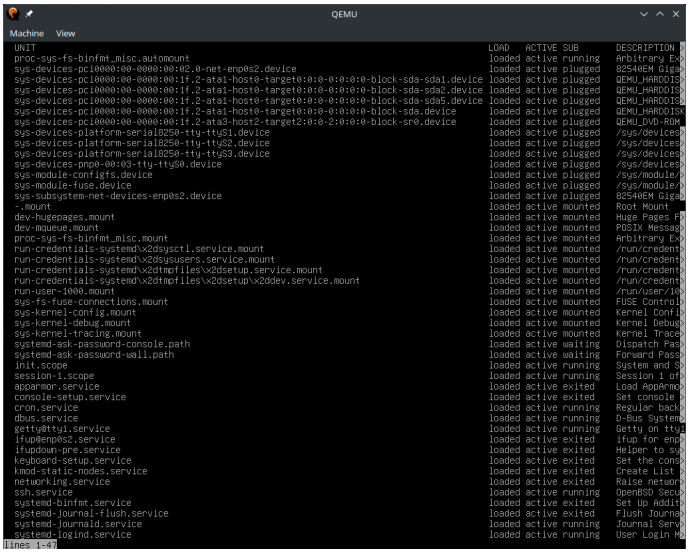
systemctl :



systemctl apache :



systemctl ssh :



---

3 /Install postgresql

Put yourself in administrator mode and write as usual:

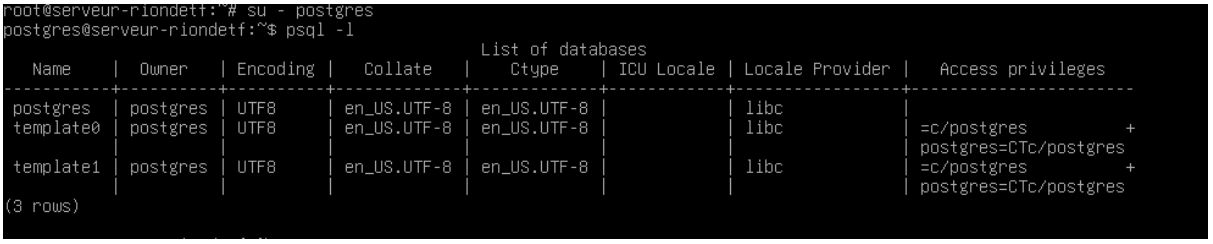
```
su -
```

```
apt install postgresql
```

You'll still need to answer “yes” to all the questions. Once this is done, you can log in by writing :

```
su - postgresql
```

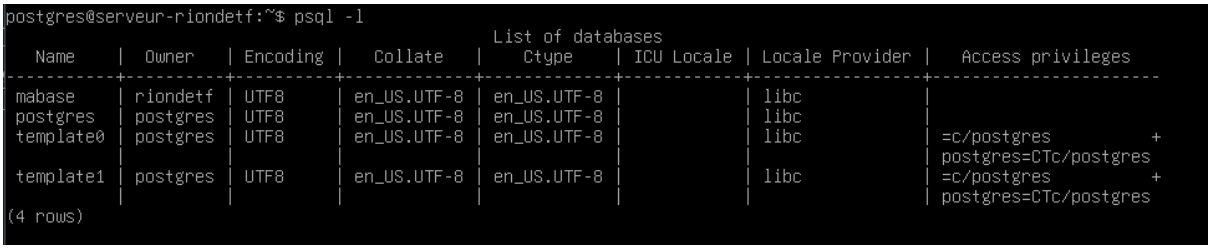
and display the base tables with psql -l.



Now it's up to you to enter the tables you want to create your database, but first don't forget to create a new user:

```
CREATE USER “votre nom d’utilisateur”
```

```
CREATE DATABASE “nom_base’ OWNER “nom_role;
```



-

```
CREATE TABLE matable (
   prenom          varchar(80),
   age        int,
);
```

-

```
INSERT INTO matable (prenom, age) VALUES (‘flavien’, 19);
```

To access postgres on the host machine, go to the main folder with :

```
cd /etc/postgresql/15/main/
```

Then modify the pg_hba.conf file by writing :

```
nano pg_hba.conf
```

In this file, add the line in IPv4:

```
host all all 0.0.0.0/0 scram-sha-256
```

And edit a second file

```
nano postgresql.conf
```

Then look for a first line containing listen_adresses and remove the # in front of it, then replace after equal with “*”.
And a second line password_encryption = scram-sha-256 where you must also remove the # in front.

Create your password with :

```
\password “user name”
```

You can now connect to the host machine by writing this command:

```
psql -h localhost -U riondetf -d “you’r base”
```

```
mabase=# \password riondetf
Enter new password for user "riondetf":
Enter it again:
mabase=# _
riondetf@pc-dg-033-15:~$ psql -h localhost -U riondetf -d mabase
Password for user riondetf:
psql (15.6 (Debian 15.6-0+deb12u1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

mabase=>
```

You can then look in the databases on the virtual machine to find the lines of pg_shoadow that contain the encrypted password you entered.

```
 riondetf |    16388 | t          | f        | f       | f           | SCRAM-SHA-256$4096:d/ArSRqJo8nSZscd8fj7jA==$OykwOLK86ZK
38PPAauLF90H/xFDFQpaD9QszXXDMeN8=:wNIB7h7pw7HyEGqPKRv7ZvMy/8n83BonVDQT5ixQSDc= |          |
(2 rows)
```

4/ Install php

Put yourself in administrator mode with the command su -.

Install php: apt install php

Create a file in the /var/www/html directory:

cd /var/www/html, pour se déplacer vers une destination

touch info.php, for create the file

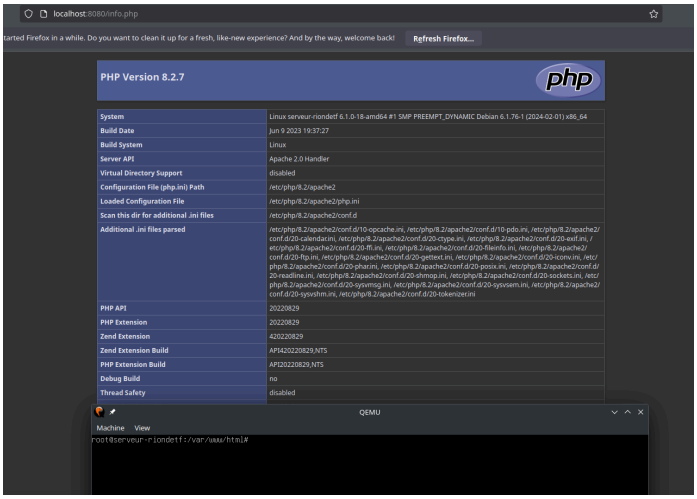
nano info.php, for open the file

then write these lines:

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

```
?>
```

On the host machine, go to: http://localhost:8080/info.php and you can see on the page that the server is running.



5/ Install phppgadmin

Finally, we'll look at how to install phppgadmin.

```
su -
```

```
apt install phppgadmin
```

Now we need to find the Connection.php file and modify it with this command:

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

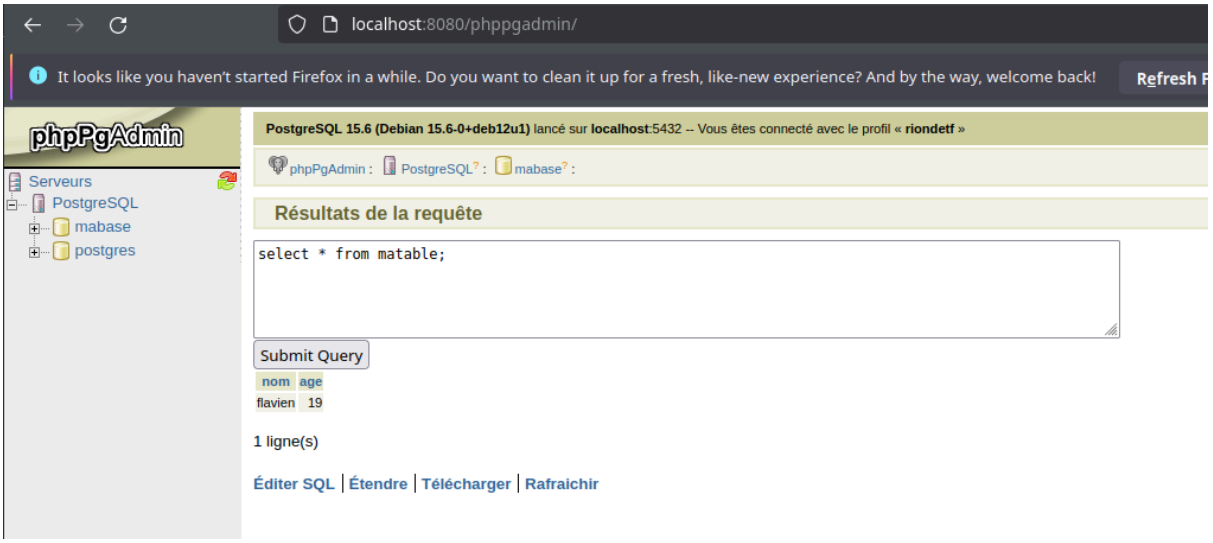
Once the file is found, change the line :

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

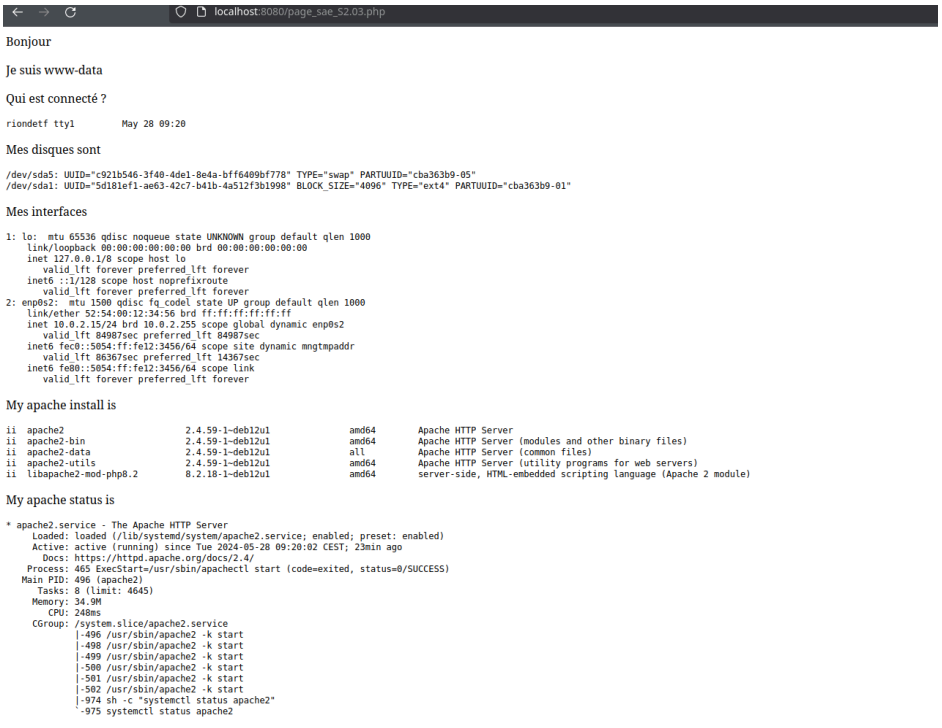
at

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

Now you can go and look at your tables on the host machine in the browser as shown below:



The virtual machine according to the browser :



The remaining space on the machine with df -h :

```
riondetf@serveur-riondetf:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           392M  480K  392M   1% /run
/dev/sda1       3.0G  1.6G  1.3G  56% /
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
```

Then, for safety reasons, always perform these two commands regularly:

```
root@serveur-riondetf:~# apt update_
```

```
root@serveur-riondetf:~# apt upgrade_
```