INSTALL DEBIAN 11 SYSTEM WITH APACHE, POSTGRESQL, PHP, PHPPGADMIN

Author : Bastien Dumont

Table of Contents

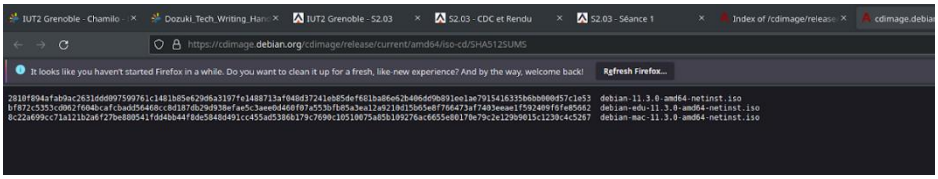
Chapter 1 - Install Debian system

A: Before Installation - ISO image

First, you need to check if the ISO image you downloaded is the good one. You can find it by clicking on this link:

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

Scroll down and click and the link names “SHA512SUM” not the dot sign.
You will fall on the web page and now you can check with the image ISO you downloaded which is the good one and if you have the good one.



Your ISO image is a good one.

CONGRATULATIONS

You’ve passed the first step. You can go on but be careful this is only the beginning of the QEMU/KVM adventure, there is a long way to go.

B: Installation of Debian system on Qemu/KMV

You can now install you Debian 11 system on a virtual machine Qemu/KMV with this command:

lance_qemu="qemu-system-x86_64 -M 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"

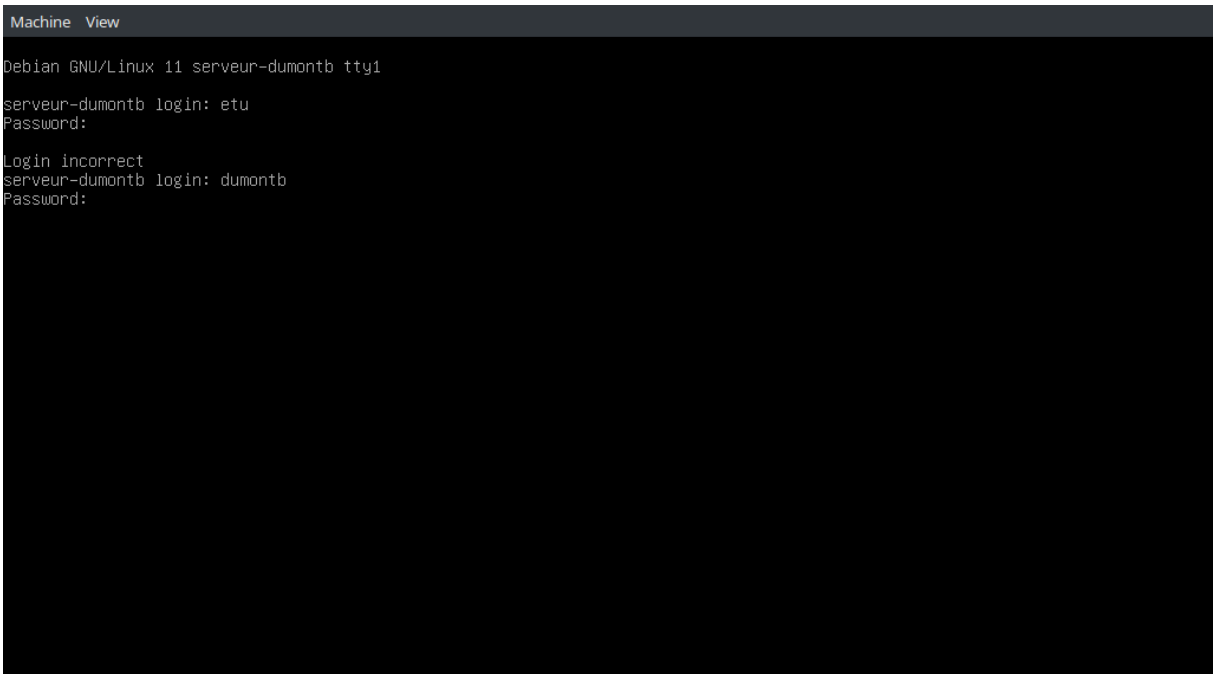
Explanation:

- ***qemu-system-x86_64:*** **version of Qemu.**
- ***-device VGA,xres=1024,yres=768:*** **size of the window open by the virtual machine.**
- ***hostfwd=tcp::2222-:22:*** **port of the virtual machine "22" and host machine "2222" for ssh.**
- ***hostfwd=tcp::4443-:443: :*** **port of the virtual machine "443" and host machine "4443" for HTTPS protocol.**
- ***hostfwd=tcp::8080-:80:*** **port of the virtual machine "80" and host machine "8080" for HTTP protocol.**
- ***hostfwd=tcp::5432-:5432:*** **port of the virtual machine "5432" and host machine "5432" for the connection to PostgreSQL.ss**

The command will open a window to set up the installation. Choose without graphical interface, you won't need it for what we intend to do. Then, the installation will demand you some settings. Here the most important and below two screenshots for as an example:

- **Hostname: server-YOUR_LOGIN**
- **Root password: The password for the superuser important to keep it in your memory**
- **User account: Your full name**
- **Username: Your login (1st screenshot)**
- **Partition disks: Guided- use entire disk & All files in one partition**
- **Partition disks second time: Select yes**
- **Software selection: Uncheck "Debian Desktop" if it is and check "ssh server" if it's not (2nd screenshot)**
- **Install GRUB: Select yes**
- **Device for boot loader: /dev/sda**

After all these steps the virtual machine will restart, and this window will come up. Enter your username and user password.



AND THEN… you are now connected to your virtual machine. You can turn it down

Icone important: **To stop your virtual machine you must execute the command ***poweroff* with the root user. ***su* - and then the root password.**

C: Characteristics and use of QEMU/KMV virtual machine

Here you are, connected to your Debian 11 system on your new favorite virtual machine. But you don't know everything about it. Well, that's completely OKAY! I'm here to give you some tips to know about it and what you can do.

First thing you should know is the IP address and the MAC address. You can find them with the command ***ip addr*. Look at the screenshot below.

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Apr  8 08:44:09 CEST 2022 on tty1
dumontb@serveur-dumontb:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 52:54:00:12:34:56 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s2
       valid_lft 86378sec preferred_lft 86378sec
    inet6 fec0::5054:ff:fe12:3456/64 scope site dynamic mngtmpaddr
       valid_lft 86383sec preferred_lft 14383sec
    inet6 fe80::5054:ff:fe12:3456/64 scope link
       valid_lft forever preferred_lft forever
dumontb@serveur-dumontb:~$ _
```

Then your machine can reach the outside in particular domain name system. You can try with a random DNS with ***traceroute* and see what happens.

```
dumontb@serveur-dumontb:~$ traceroute google.com
traceroute to google.com (142.250.179.78), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  0.299 ms  0.350 ms  0.226 ms
 2  sw-dg-40d-1-tx.iut2.upmf-grenoble.fr (192.168.141.19)  2.365 ms  2.522 ms  2.353 ms
 3  rt-wan.iut2.upmf-grenoble.fr (193.55.51.1)  1.189 ms  1.358 ms  1.843 ms
 4  r-viallet1.grenet.fr (193.54.184.185)  1.161 ms  0.857 ms  1.584 ms
 5  * * *
 6  te1-4-grenoble-rtr-021.noc.renater.fr (193.51.181.94)  1.974 ms  1.845 ms  1.465 ms
 7  te-0-1-0-12-ren-nr-lyon2-rtr-091.noc.renater.fr (193.51.180.67)  7.236 ms ten0-0-0-12-ren-nr-lyon2-rtr-091.noc.renater.fr (1
93.51.177.57)  6.899 ms  7.216 ms
 8  xe-1-0-1-marseille2-rtr-131.noc.renater.fr (193.51.177.196)  8.556 ms te1-5-marseille2-rtr-021.noc.renater.fr (193.51.177.16
9)  7.529 ms xe-0-0-14-marseille2-rtr-131.noc.renater.fr (193.51.180.105)  7.204 ms
 9  72.14.218.132 (72.14.218.132)  17.740 ms  15.247 ms  14.861 ms
10  74.125.244.211 (74.125.244.211)  7.452 ms 108.170.252.243 (108.170.252.243)  7.410 ms 74.125.244.216 (74.125.244.216)  6.751
 ms
11  216.239.35.201 (216.239.35.201)  15.700 ms 216.239.35.209 (216.239.35.209)  12.653 ms 216.239.35.201 (216.239.35.201)  15.37
3 ms
12  209.85.142.200 (209.85.142.200)  13.434 ms 209.85.255.107 (209.85.255.107)  13.431 ms  13.453 ms
13  108.170.244.193 (108.170.244.193)  12.795 ms 108.170.245.1 (108.170.245.1)  13.727 ms  13.625 ms
14  142.251.49.133 (142.251.49.133)  12.575 ms  12.775 ms 142.251.49.131 (142.251.49.131)  12.975 ms
15  par21s19-in-f14.1e100.net (142.250.179.78)  13.098 ms  12.815 ms  13.001 ms
dumontb@serveur-dumontb:~$ _
```

But for now, there is not too many things on it. All you have to do is to install new packets! Will see later useful packets for developers but now just show how to install packets. The command ***apt install PACKET'S NAME* is made for it. You can try with the packet micro.

(A screenshot of the files /etc/fstab)

```
root@serveur-dumontb:~# 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=d3044519-47d2-43ca-bfbc-fe75c2e5f47d /               ext4    errors=remount-ro 0
1
# swap was on /dev/sda5 during installation
UUID=5ff9ce4b-db0d-47f3-9f22-0667898b1192 none            swap    sw              0       0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
```

CONGRATULATIONS !!!

You've just completed the chapter 1. Is one step more in your adventure. Go on !

Useful tips : You can connect to your virtual machine directly on your bash the linux computer you use. The command is : *****ssh YOUR_LOGIN@localhost -p 2222.* **All the screens used for the next part will be taken on a bash. You can type **systemctl status ssh to see the status of the protocol. As you can see on the screenshot below it's currently running**

```
root@serveur-dumontb:~# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2022-04-08 09:46:02 CEST; 49s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 385 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 435 (sshd)
      Tasks: 1 (limit: 4679)
     Memory: 6.1M
        CPU: 69ms
     CGroup: /system.slice/ssh.service
             └─435 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Apr 08 09:46:02 serveur-dumontb systemd[1]: Starting OpenBSD Secure Shell server...
Apr 08 09:46:02 serveur-dumontb sshd[435]: Server listening on 0.0.0.0 port 22.
Apr 08 09:46:02 serveur-dumontb sshd[435]: Server listening on :: port 22.
Apr 08 09:46:02 serveur-dumontb systemd[1]: Started OpenBSD Secure Shell server.
Apr 08 09:46:16 serveur-dumontb sshd[519]: Accepted password for dumontb from 10.0.2.2 port>
Apr 08 09:46:16 serveur-dumontb sshd[519]: pam_unix(sshd:session): session opened for user >
```

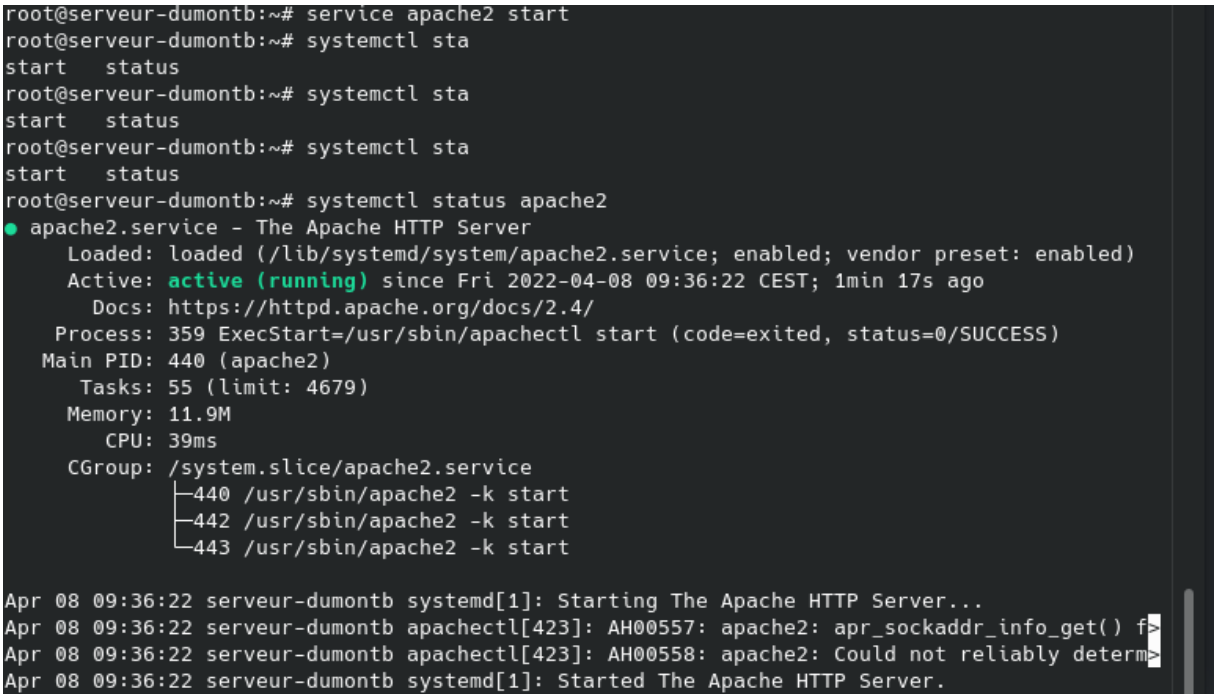
Now that you have a Debian 11 system perfectly install. It’s time to install packages to be a good and a complete developer.

CHAPTER 2 – Install Apache

A: Installation Step

Firstable, you are going to install an Apache server. But it deserves a little explanation. Apache is a software, it consits to make link between the client request and the differents web browsers which are the server. When you navigate on internet and you click on a link to go on a web page. The web browser will send a request to Apache and it will returns an answer with all the files demand incuding text and images. This is what Apache is for.

But how to install it. First, go on your root user. And then, it's very easy, it's like intalling micro before but by changing the name of the package. Type the command ***apt install Apache2* and Y to confirm the installation. To turn it start it type ***service apache2 start.* You can look at it status to make sure is currrently running. The command is : ***systemctl status apache2*. You can look at the screenshot



If it's not written "active (running)" you can restart the apache2 package by typing ***systemlctl restart apache2* and it will be good. If all works correctly, you can go on.

B : Resquest/Answer – Apache serveur

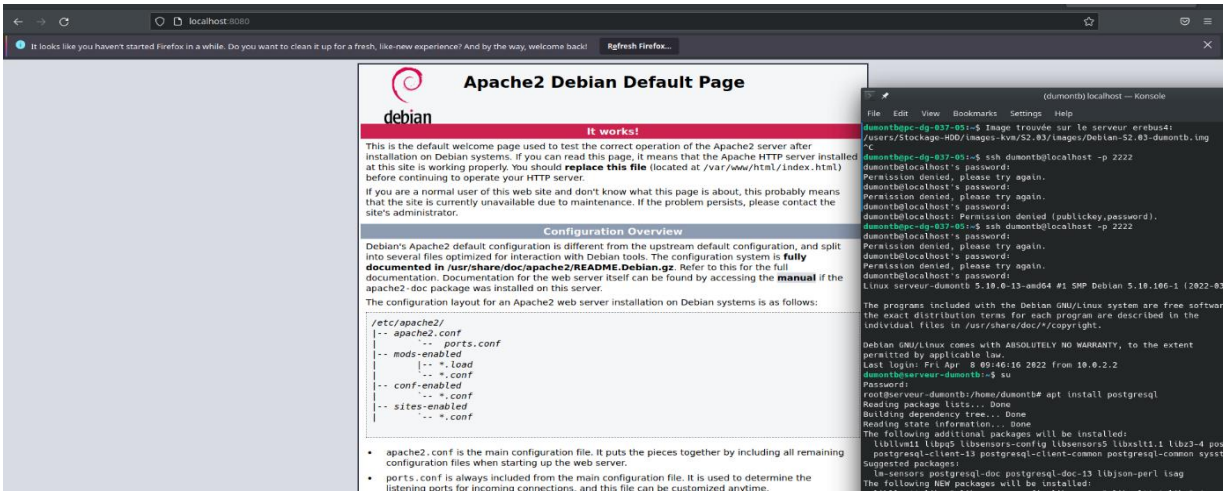
It is impossible to show a web page on your virtual machine but you can still send a request to a server web with the software “telnet”. Write this string “`HEAD / HTTP/1.0`” following by two back to line. The server should answer

“`HTTP/1.1 200 OK`”. You can look at the screenshot below.

```
dumontb@serveur-dumontb:~$ telnet localhost 80
Trying ::1...
Connected to localhost.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 09 May 2022 14:26:45 GMT
Server: Apache/2.4.53 (Debian)
Last-Modified: Fri, 08 Apr 2022 07:27:25 GMT
ETag: "29cd-5dc1f86ae785d"
Accept-Ranges: bytes
Content-Length: 10701
Vary: Accept-Encoding
Connection: close
Content-Type: text/html
```

On the other hand, you can show an web page directly on your host machine. To do this you need to redirect a port on the host machine (for example 8080) to port 80 (default port for web servers) on the virtual machine. This is make you the command to install the virtual machine as explained on the chapter 1. Now on your host machine, go on a web browser and search the URL : http://localhost:8080 and you should fall on the default web page of your virtual machine’s apache server.



You can click on this link to go further :
https://httpd.apache.org/docs/2.4/en/install.html

CHAPTER 3 : Install PostgreSQL

A: Installation step and use from the virtual machine

The second package to install is PosgreSQL. But what is it ? It's a database management system which uses the language SQL used for the database management.

To install it, start with the command ***apt install postgresql,* it will install the client and the serveur on your virtual machine. Use ***systemctl status postgresql* to verify if it's currently running else restart it with ***systemctl restart postgresql.*

```
root@serveur-dumontb:/home/dumontb# systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enable>
     Active: active (exited) since Mon 2022-05-09 15:15:21 CEST; 24min ago
   Main PID: 1797 (code=exited, status=0/SUCCESS)
      Tasks: 0 (limit: 4679)
     Memory: 0B
        CPU: 0
     CGroup: /system.slice/postgresql.service

May 09 15:15:21 serveur-dumontb systemd[1]: Starting PostgreSQL RDBMS...
May 09 15:15:21 serveur-dumontb systemd[1]: Finished PostgreSQL RDBMS.
lines 1-11/11 (END)
```

To log in the posgres server, type the command ***su – postgres.* You are now connected to the postgres serveur with the user postgres which is the superUser (root equivalent).

```
root@serveur-dumontb:/home/dumontb# su - postgres
postgres@serveur-dumontb:~$ psql
```

Then, psql to log in postgreSQL with the command ***psql*. And you are now conneted as the superUser.

```
postgres@serveur-dumontb:~$ psql
psql (13.5 (Debian 13.5-0+deb11u1))
Type "help" for help.
```

To test postgreSQL create a user named YOUR_LOGIN with the following password YOUR_LOGIN. Then, create a database. You can type ***psql -l* to see all the database Disconnet from the postgres user and reconnect with the user you just create → ***psql -h postgres-info NAME_BASE -U NOM_USER*. Then, you can create a table and put some data inside it. *|d* to visualize them.**

```
postgres=# CREATE USER dumontb with password 'dumontb';
CREATE ROLE
postgres=# CREATE DATABASE base;
CREATE DATABASE
```

```
postgres@serveur-dumontb:~$ psql -l
                                 List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
-----------+----------+----------+-------------+-------------+-----------------------
 base      | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
(4 rows)

postgres@serveur-dumontb:~$
```

B : Use postgreSQL from host machine

This is cool but we can do it better. There is a way to use the postgreSQL base you just create directly on your host machine. To do this, you need to edit some configuration files. The two files are “pg_hba.conf” and “postgresql.conf”. Both are in the directories “/etc/postgresql/13/main/”. For, the first one “pg_hba.conf”, add a line “host all all 0.0.0.0/0 scram-sha-256” at the end of the file. And change the first line no commented by replacing “md5” by “trust”. For the second file, scroll down until you see the line “password_encryption = “ change the value after the equal by “scram-sha-256”. And “crtl + S” to save the file.

You can now go on your host machine, open a bash and enter the command : ***psql -h localhost -U USERNAME NAME_BASE.* Then, type the password, it should be the your username. Annd here you are, connected to your postgreSQL base with your user you create from your virtual machine. *|d* to see you table.**

Check that the password encrytion is the one you change. Normally ‘sram-sha-256”. Type ***select * from pg_shadow* and you should see the good password encryption.(All the screen shots to the step B)

```
# Database administrative login by Unix domain socket
local   all             postgres                                md5

# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             127.0.0.1/32            md5
# IPv6 local connections:
host    all             all             ::1/128                 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     peer
host    replication     all             127.0.0.1/32            md5
host    replication     all             ::1/128                 md5
host    all             all             0.0.0.0/0               scram-sha-256
```

```
  GNU nano 5.4                /etc/postgresql/13/main/postgresql.conf

# - TCP settings -
# see "man tcp" for details

#tcp_keepalives_idle = 0                # TCP_KEEPIDLE, in seconds;
                                        # 0 selects the system default
#tcp_keepalives_interval = 0            # TCP_KEEPINTVL, in seconds;
                                        # 0 selects the system default
#tcp_keepalives_count = 0               # TCP_KEEPCNT;
                                        # 0 selects the system default
#tcp_user_timeout = 0                   # TCP_USER_TIMEOUT, in milliseconds;
                                        # 0 selects the system default


# - Authentication -

#authentication_timeout = 1min          # 1s-600s
password_encryption = scram-sha-256             # md5 or scram-sha-256
#db_user_namespace = off
```

```
# - Connection Settings -

listen_addresses = '*'                  # what IP address(es) to listen on;
```

```
dumontb@pc-dg-037-05:~$ psql -h localhost -U dumontb base
Password for user dumontb:
psql (13.5 (Debian 13.5-0+deb11u1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: o
ff)
Type "help" for help.
```

```
dumontb@base=> select * from table_simple ;
   nom1    |   nom2    |   nom3
-----------+-----------+-----------
 donnée 1 | donnée 2 | donnée 3
 donnée 4 | donnée 5 | donnée 6
(2 rows)
```

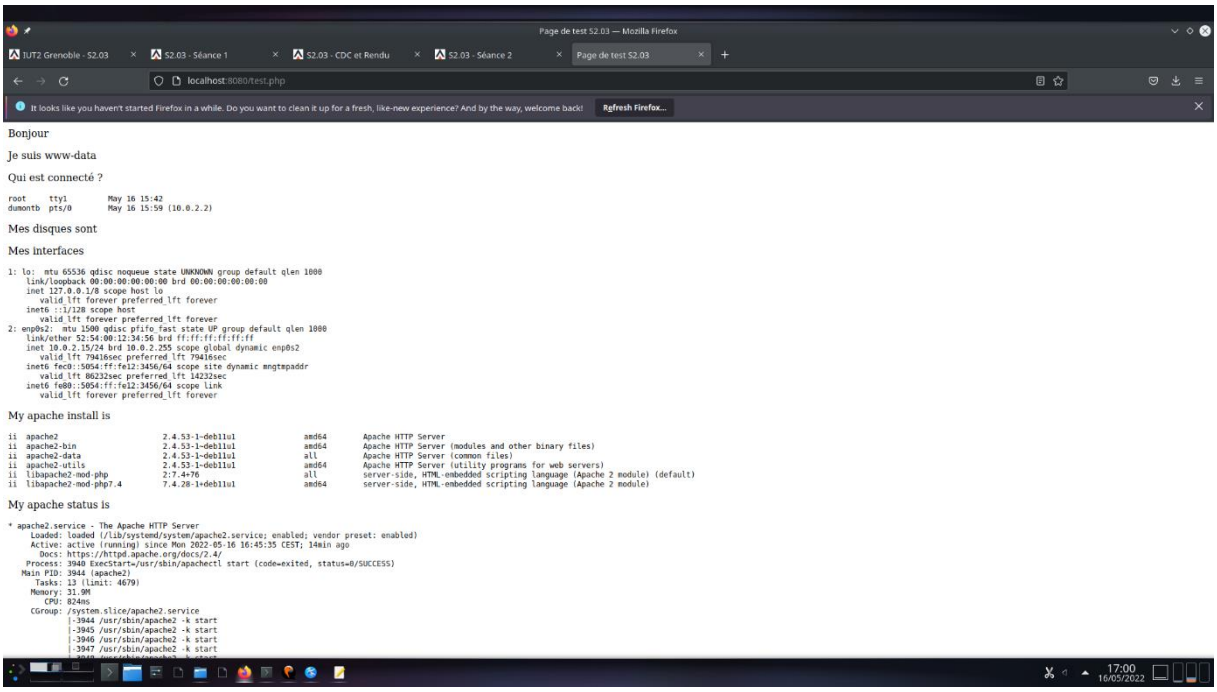
```
                    |           |
 dumontb  |    16384 | f          | f        | f       | f            | SCRAM-SHA-256$4096:
Nw9/F5ummPr/gCB3Qozp1A==$9pYcvGufg6AFhw1Dcm6wOjDp+0lJTr0Eaxn5QkpDwuw=:O88J+AQM2MmmLq4SPVzC4Z
apBPqP3UR5QjnvktZTO2Q= |           |
(2 rows)
```

CHAPTER 4 – Install PHP

In the chapter, you are going to intall PHP. It's a prgramming language mainly used to create dynalmics web pages throught an HTTP server. PHP will be intalled on the Apache serveur of your vitual machine.

To download it, the command is : ***apt install php-common libapache2-mod-php php-cli*. After the download, you need to restart apache2. To do this just make the command : ***systemctl restart apache2*. And php is now installed in you virtual machine.

There is also a way to visualise the PHP file you make on your host machine. You just have to put the file.php into the directories "/var/www/html/". After that, go on your host machine, open a web browser and enter the following url : http://localhost:8080/YOUR_FILE.php". You would be able to see your file.php on a web page. Here is an example.



If you want to go further :
https://www.php.net/manual/en/install.unix.php

CHAPTER 5 – Install PhpPgAdmin

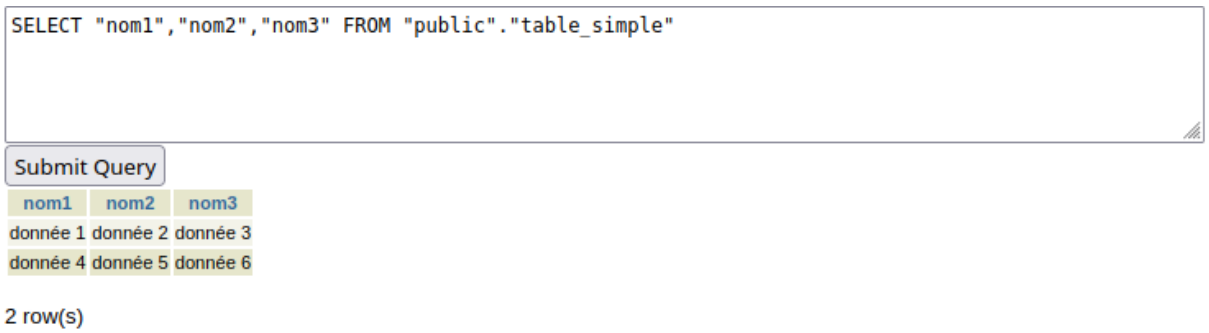
A : Intallation step

The final package that you will installed is PhgPgAdmin. It's a web application made with PHP language. It's used to the management of database postgreSQL.

To download it, you need to enter : ***apt install -y phppgadmin apache2.* Then, ***nano /etc/phppgadmin/config.inc.php* to edit this configuration file. Scroll down until the line "$conf['owned_only'] = '' and put false instead of the the value after the equal. Save the modification. You need to edit a second configuration file. This one is named "phppgadmin.conf". Do the same command but just replace the name of the file. Change the line "require local" by "require all granted". You can save and restart apache2 : ***systemctl restart apache2*.

B : Access to PhpPgAdmin

To access to it, put the url : http://localhost:8080/phppgadmin on a web browser of your host machine. You will fall on a web page of phpPgAdmin. You can connect to you database you the user you create. And then... IT'S GOOD. You can access to all your table and you can make some request like a select for example.

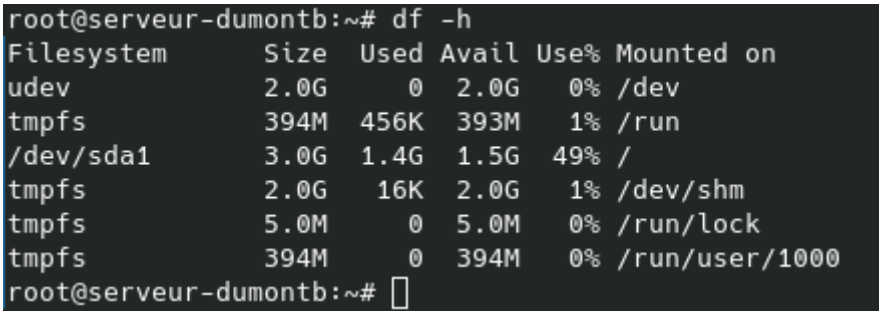


CONGRATULATIONS !!

You can now manage you database with phppgadmin easily on a web browser of your host machine.

Chapter 6 – The end

You have now a QEMU/KMV virtual machine with a Debian 11 system and 4 packages : Apache, PostgreSQL, Php and PhpPgAdmin. It's interesting to know how much GO you used to install all. Type the command ***df -h* and you will able to see that.



Thanks for reading this manuals. Hope it helped you and you learned some interesting things.

Bastien Dumont A1.