Network services installation guide

Camus Mathieu A1

In this guide, you'll learn how to install an operating system on a Qemu/KVM virtual machine and configure it to host the following network services:

- Apache
- Postgresql
- PHP

Summary

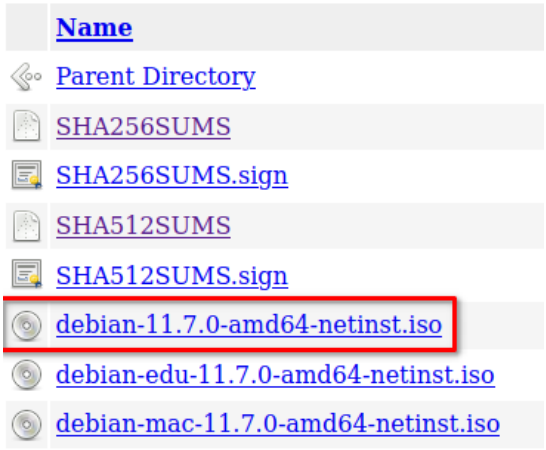
Operating system

Before installing any network services, you need to set up a server to host them. For this installation, we will use a Qemu/KVM virtual machine which will act as the server. We will install Debian 11 as the operating system.

Download ISO file

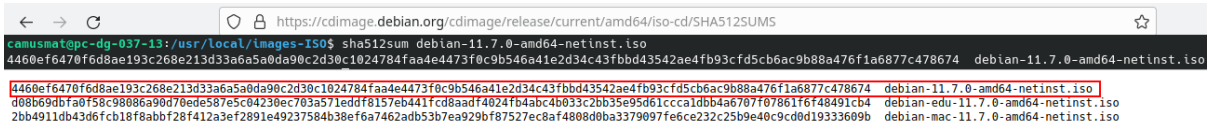
To download the iso file of Debian 11, you need to visit

this website. Here, you must scroll to the bottom of the page and download the file named *debian-11.X.X-amd64-netinst.iso*.



Then, verify the checksum to make sure that the file you have downloaded is correct. For this, you have to run the following command: `sha512sum` *`PATH/TO/ISO/FILE`*

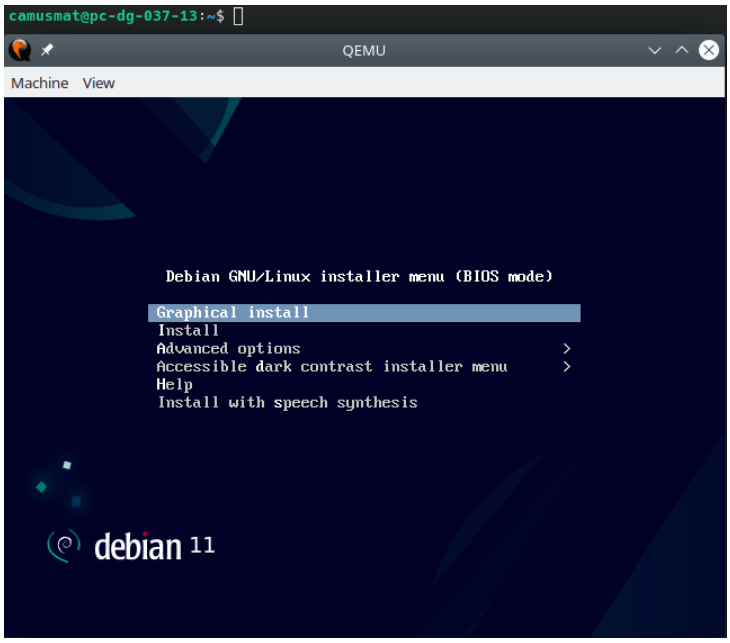
Then compare the result with the characters in the SHA512SUMS file, which you can find on the ISO image download page.



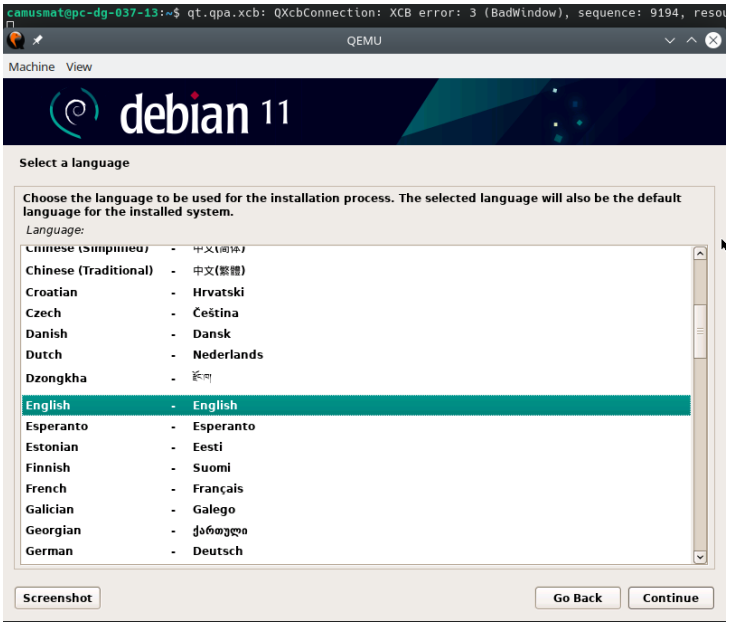
Installation

Here are the steps to install Debian 11:

1. Select *Graphical install*

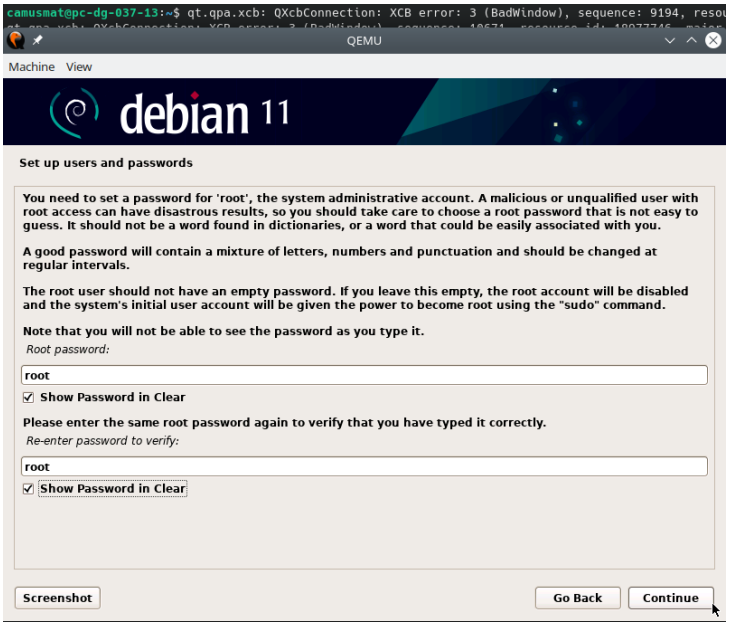


2. Select your language, your keyboard layout, etc.



3. Enter the hostname (unique name for a device on a network.) of your server

4. Set up the root user.



5. Set up the regular user.

6. Choose the partitioning method. For this installation, we will select “Guided - use entire disks”

7. Choose whether you want to put all files in the same partition or separate home directories. Here, we'll choose all files in one partition.

8. Select Yes to write the changes to the disks.

9. Disable all graphical interfaces for a server installation. Select only SSH server and standard system utilities.

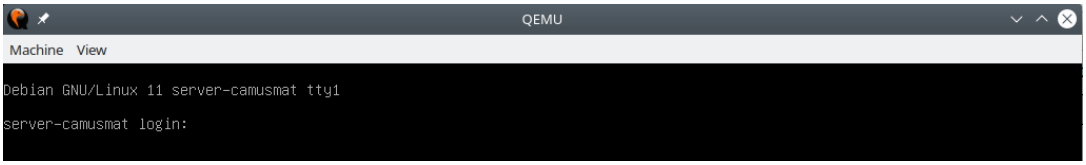
10. Select Yes to install GRUB boot loader.

11. Select the partition on which you want to install GRUB.

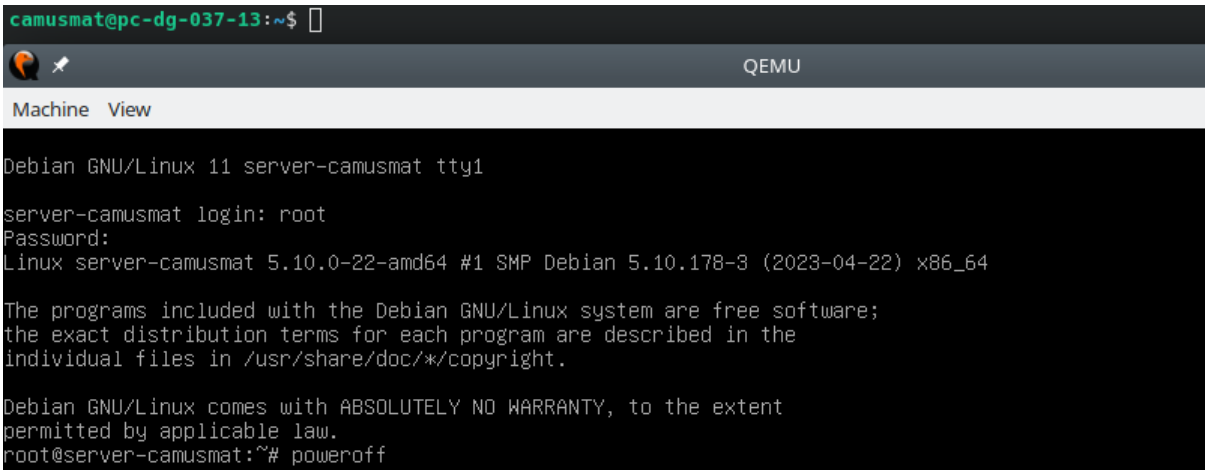
12. The installation is complete!

Once installed

When the installation is completed, you should see this interface:



You must log in as root and type the command `poweroff` to shut down the machine.

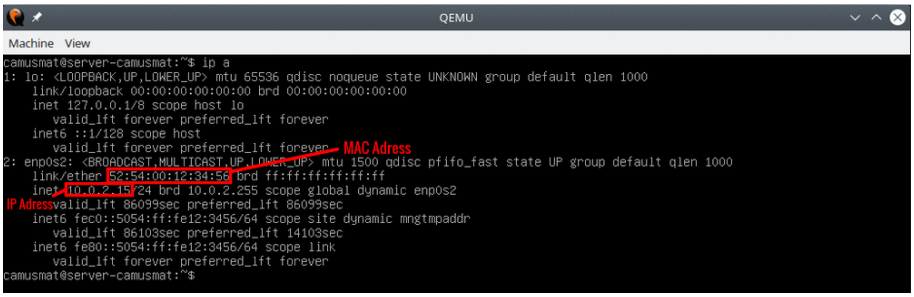


You must use this command to shut down the server properly.

Installation check

Network access

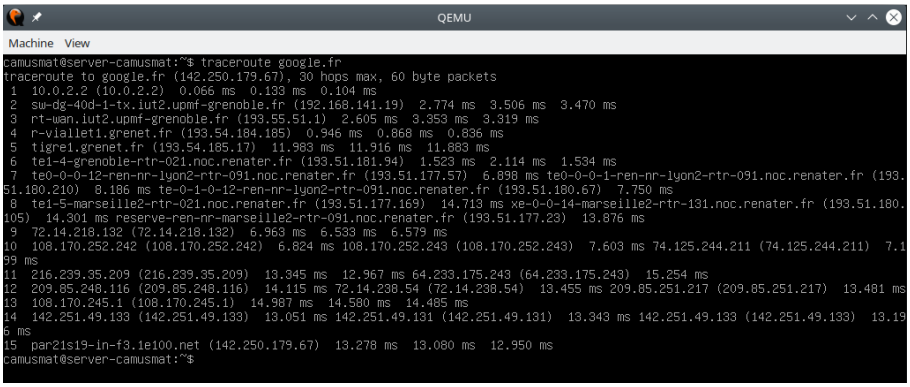
1. Check the machine's ethernet and ip characteristics with the command `ip a`



2. Check that the machine can access outside with the command:

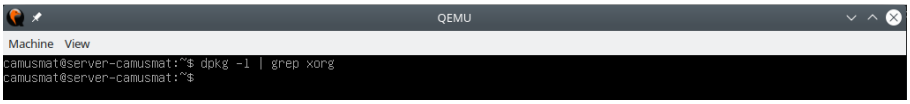
   `traceroute EXTERNAL_SERVER`

   For exemple `traceroute google.fr`
   You should get a result similar to this one:



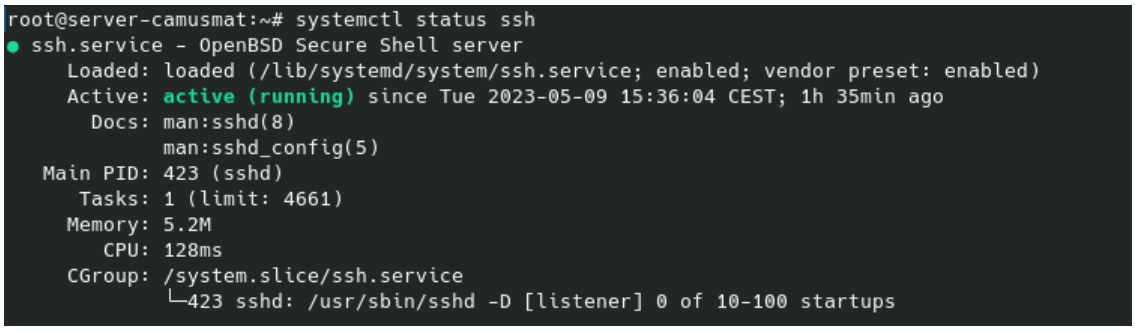
3. Check that no Xorg server is installed with the following command: `dpkg -l | grep xorg`

   If it is not installed, nothing should be displayed.



SSH connection

First, check that an SSH server is running on the virtual machine with the following command: `sytemctl status ssh`



To connect to your server via an SSH connection from the machine running the virtual machine, type this command in a terminal: `ssh LOGIN@localhost -p 2222`

```
camusmat@pc-dg-037-13:~$ ssh camusmat@localhost -p 2222
The authenticity of host '[localhost]:2222 ([127.0.0.1]:2222)' can't be established.
ECDSA key fingerprint is SHA256:9f/qeZb1Ycp1J3f0RvsHqiWRB7beQfxr/nR/93qX3Hw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2222' (ECDSA) to the list of known hosts.
camusmat@localhost's password:
Linux server-camusmat 5.10.0-22-amd64 #1 SMP Debian 5.10.178-3 (2023-04-22) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri May  5 16:03:11 2023
camusmat@server-camusmat:~$
camusmat@server-camusmat:~$
```

Once logged in, you can change to the root user with the following command: `su -`. Then enter the root password.

```
camusmat@server-camusmat:~$ su -
Password:
root@server-camusmat:~#
```

To test that everything works, you can install a package, such as micro, with the following command (as root) : `apt install micro`

Network services

Apache

Installation

To install Apache, enter the following command as root (source): `apt install apache2`

Installation check

Next, check that apache has been started with the following command: `systemctl status apache2`

You should get this result:

```
root@server-camusmat:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-05-09 15:39:56 CEST; 4min 19s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 904 (apache2)
      Tasks: 55 (limit: 4661)
     Memory: 9.0M
        CPU: 41ms
     CGroup: /system.slice/apache2.service
             ├─904 /usr/sbin/apache2 -k start
             ├─906 /usr/sbin/apache2 -k start
             └─907 /usr/sbin/apache2 -k start

May 09 15:39:55 server-camusmat systemd[1]: Starting The Apache HTTP Server...
May 09 15:39:55 server-camusmat apachectl[903]: AH00558: apache2: Could not reliably determine the server's fully qualified dom
May 09 15:39:56 server-camusmat systemd[1]: Started The Apache HTTP Server.
```

If you don't get this result, restart apache2 with the following command and try again: `systemctl restart apache2`.

As the server has no graphical interface, it is impossible to display an HTML page. However, you can check that the Apache server is running using telnet. To do this, follow these steps:

1. `telnet localhost 80`

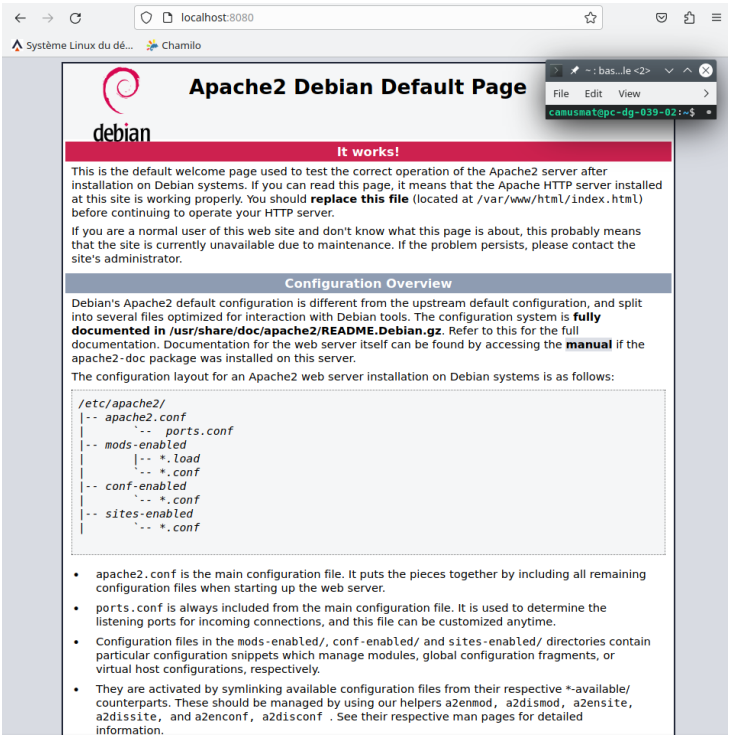
2. `HEAD / HTTP/1.0`
3. The server should reply `HTTP/1.1 200 OK` as follows:

```
root@server-camusmat:~# exit
logout
camusmat@server-camusmat:~$ telnet localhost 80
Trying ::1...
Connected to localhost.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Tue, 09 May 2023 13:46:22 GMT
Server: Apache/2.4.56 (Debian)
Last-Modified: Tue, 09 May 2023 13:39:54 GMT
ETag: "29cd-5fb42e3f4981c"
Accept-Ranges: bytes
Content-Length: 10701
Vary: Accept-Encoding
Connection: close
Content-Type: text/html

Connection closed by foreign host.
```

You can also access Apache server HTML pages from the machine running the VM, thanks to the redirection of port 80 to port 8080 in the VM launch script.

In a browser on the host machine, access the following URL: `http://localhost:8080`

You should see the following page:



Postgresql

Installation

To install Postgresql, enter the following command as root (source): `apt install postgresql`

Installation check

First, check that the postgresql server is running on the virtual machine with the following command: `sytemctl status postgresql`

```
root@server-camusmat:~# systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
     Active: active (exited) since Tue 2023-05-09 16:21:19 CEST; 50min ago
   Main PID: 4380 (code=exited, status=0/SUCCESS)
      Tasks: 0 (limit: 4661)
     Memory: 0B
        CPU: 0
     CGroup: /system.slice/postgresql.service
```

Then, log in with the postgres login using the following command from the root account: `su - postgres`

You can list default tables with `psql -l` as shown below:

```
root@server-camusmat:~# su - postgres
postgres@server-camusmat:~$ psql -l
                                 List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
-----------+----------+----------+-------------+-------------+-----------------------
 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
(3 rows)

postgres@server-camusmat:~$ _
```

Usage

Connect to PostgreSQL with the command: `psql`

Create a new user with the command: `CREATE USER username;`

Then, you can create a database for this user with the command:

`CREATE DATABASE basename WITH OWNER=username;`

You can check the creation of a user with `\du` and you can check the creation of a new base with `\l`.

Here is an example:

```
postgres=# CREATE USER camusmat;
CREATE ROLE
postgres=# CREATE DATABASE camusmatbase WITH OWNER=camusmat;
CREATE DATABASE
postgres=# \du
                                   List of roles
 Role name |                         Attributes                         | Member of
-----------+------------------------------------------------------------+-----------
 camusmat  |                                                            | {}
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

postgres=# \l
                                   List of databases
     Name     |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
--------------+----------+----------+-------------+-------------+-----------------------
 camusmatbase | camusmat | 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=# _
```

To create tables, you need to connect to psql with the user who owns the database.

To do so, log out of PostgreSQL with `\q`, then change the user to your database owner with `su username`. Then reconnect to the database with the command: `psql basename`.

Finally, create a table with `CREATE TABLE tablename ([type columnname …]);`

You can insert values with this command:

`INSERT INTO tablename VALUES (some values) [, (other values) …];`

Here is an example:

```
camusmat@server-camusmat:~$ psql camusmatbase
psql (13.10 (Debian 13.10-0+deb11u1))
Type "help" for help.

camusmatbase=> CREATE TABLE MATABLE (id int primary key, info varchar);
CREATE TABLE
camusmatbase=> INSERT INTO matable (0, 'info1'), (1, 'info2'), (2, 'info3');
ERROR:  syntax error at or near "0"
LINE 1: INSERT INTO matable (0, 'info1'), (1, 'info2'), (2, 'info3')...
                             ^
camusmatbase=> INSERT INTO matable VALUES (0, 'info1'), (1, 'info2'), (2, 'info3');
INSERT 0 3
camusmatbase=> SELECT * FROM matable ;
 id | info
----+-------
  0 | info1
  1 | info2
  2 | info3
(3 rows)
```

Access PostgreSQL from the host machine

To access PostgreSQL from the host machine, you need to modify some configuration files. You can use nano to edit a file.

First, you need to modify the `postgresql.conf` file located at `/etc/postgresql/13/main/`. Uncomment the line where it is written “listen_addresses” and change the value to `'*'` as shown below.

```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

listen_addresses = '*'_                 # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' for all
                                        # (change requires restart)
port = 5432                             # (change requires restart)
max_connections = 100                   # (change requires restart)
#superuser_reserved_connections = 3     # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
                                        # (change requires restart)
#unix_socket_group = ''                 # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
                                        # (change requires restart)
#bonjour = off                          # advertise server via Bonjour
                                        # (change requires restart)
#bonjour_name = ''                      # defaults to the computer name
                                        # (change requires restart)
```

Uncomment the line where it is written “password_encryption” and change the value to `scram-sha-256` as shown below:

```
# - Authentication -

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

In the `pg_hba.conf` file located in the same directory, modify as follows:

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

# TYPE  DATABASE        USER            ADDRESS                 METHOD

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

Finally, restart postgresql with the command `systemctl restart posgresql` as root. You should be able to connect to PostgreSQL from the host machine with `psql -h localhost -U username basename`

```
camusmat@pc-dg-039-02:~$ psql -h localhost -U camusmat camusmatbase
Password for user camusmat:
psql (13.10 (Debian 13.10-0+deb11u1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

camusmat@camusmatbase=> \du
                                   List of roles
 Role name |                         Attributes                         | Member of
-----------+------------------------------------------------------------+-----------
 camusmat  |                                                            | {}
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

camusmat@camusmatbase=> \d
          List of relations
 Schema |  Name   | Type  |  Owner
--------+---------+-------+----------
 public | matable | table | camusmat
(1 row)

camusmat@camusmatbase=>
```

Password hashing with the SHA-256 function (pg_shadow content):

```
postgres=# SELECT * FROM pg_shadow;
 usename  | usesysid | usecreatedb | usesuper | userepl | usebypassrls |
       passwd                                                  | valuntil | useconfig
----------+----------+-------------+----------+---------+--------------+----------------------------------------------------------
----------------------------------------------------------------------------+----------+-----------
 postgres |       10 | t           | t        | t       | t            |
                                                                            |          |
 camusmat |    16385 | f           | f        | f       | f            | SCRAM-SHA-256$4096:rXSLqjdKGCd5BvP16enMyA==$XpfQmSooHnX
u2N6pO11pQ3W9ESEAvglkqdtqLy1F5PE=:98ckmXnbau4fGR2/uG4fPBAYfNzPqJwOoMKDijfrbOY= |          |
(2 rows)

(END)
```

PHP

PHP

Installation

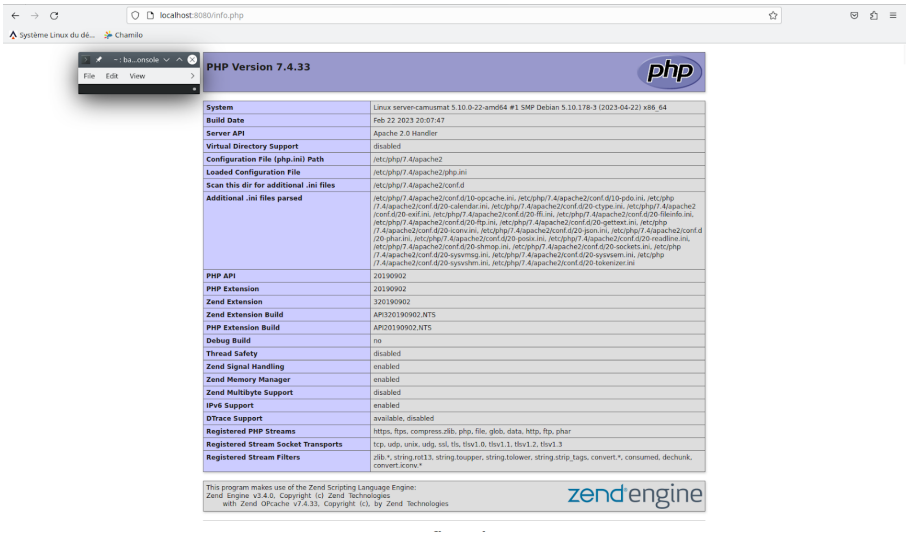
To install PHP, enter the following command as root (source): `apt install php-common libapache2-mod-php php-cli`

Installation check

To check the installation, place an `info.php` file in the `/var/www/html/` directory, containing the following code:

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

Once done, access the URL `http://localhost:8080/info.php` from a browser on the host machine. You should see the next page:



PhpPgAdmin

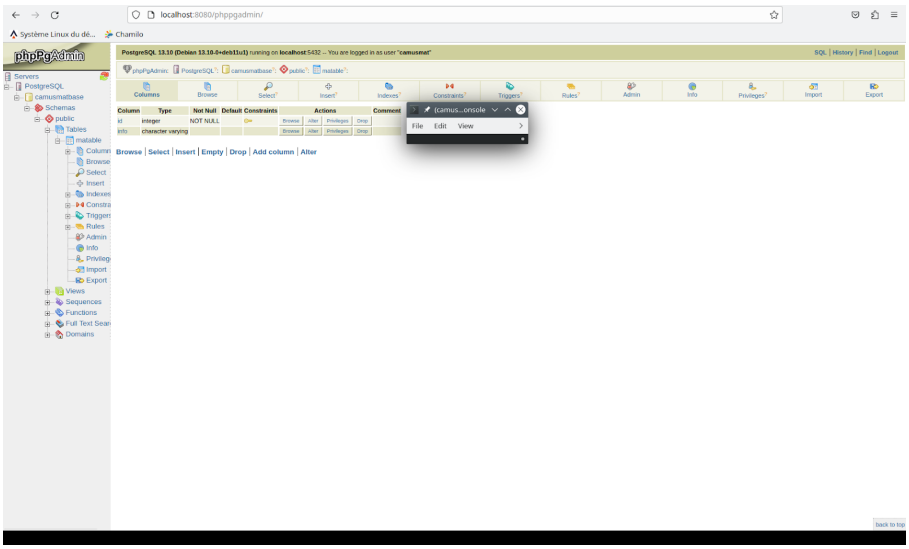
Installation

To install PhpPgAdmin, enter the following command as root (source): `apt install phppgadmin`

Edit the `phppgadmin.conf` configuration file located in the `/etc/apache2/conf-enabled/` directory as follow:

```
Alias /phppgadmin /usr/share/phppgadmin

<Directory /usr/share/phppgadmin>

<IfModule mod_dir.c>
DirectoryIndex index.php
</IfModule>
AllowOverride None

# Only allow connections from localhost:
#Require local
Require all granted

<IfModule mod_php.c>
  php_flag magic_quotes_gpc Off
  php_flag track_vars On
  #php_value include_path .
</IfModule>
<IfModule !mod_php.c>
  <IfModule mod_actions.c>
    <IfModule mod_cgi.c>
      AddType application/x-httpd-php .php
      Action application/x-httpd-php /cgi-bin/php
    </IfModule>
    <IfModule mod_cgid.c>
      AddType application/x-httpd-php .php
      Action application/x-httpd-php /cgi-bin/php
    </IfModule>
  </IfModule>
</IfModule>

</Directory>
```

You can now access the PhpPgAdmin interface via the URL `http://localhost:8080/phppgadmin/` on the host machine.

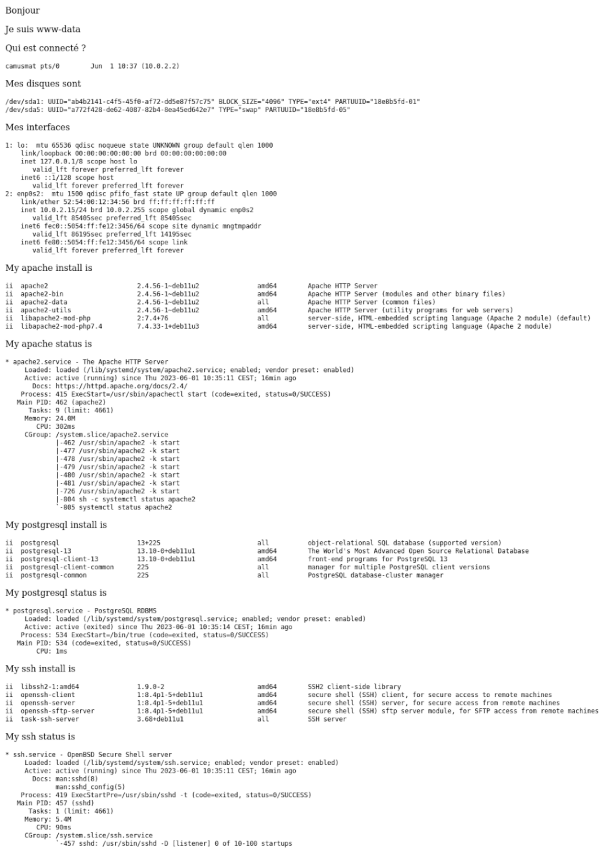


Other

`cat /etc/fstab` command result:

```
root@server-camusmat:~# exit
logout
camusmat@server-camusmat:~$ 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=ab4b2141-c4f5-45f0-af72-dd5e87f57c75 /               ext4    errors=remount-ro 0       1
# swap was on /dev/sda5 during installation
UUID=a772f428-de62-4087-82b4-8ea45ed642e7 none            swap    sw              0       0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
camusmat@server-camusmat:~$
```

Capture of the PHP page given in the specifications:



Bonjour

Je suis www-data

Qui est connecté ?

```
camusmat pts/0        Jun  1 10:37 (10.0.2.2)
```

Mes disques sont

```
/dev/sda1: UUID="ab4b2141-c4f5-45f0-af72-dd5e87f57c75" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="18e8b5fd-01"
/dev/sda5: UUID="a772f428-de62-4087-82b4-8ea45ed642e7" TYPE="swap" PARTUUID="18e8b5fd-05"
```

Mes interfaces

```
1: lo:  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:  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 85405sec preferred_lft 85405sec
    inet6 fec0::5054:ff:fe12:3456/64 scope site dynamic mngtmpaddr
       valid_lft 86195sec preferred_lft 14195sec
    inet6 fe80::5054:ff:fe12:3456/64 scope link
       valid_lft forever preferred_lft forever
```

My apache install is

```
ii  apache2                  2.4.56-1~deb11u2            amd64        Apache HTTP Server
ii  apache2-bin              2.4.56-1~deb11u2            amd64        Apache HTTP Server (modules and other binary files)
ii  apache2-data             2.4.56-1~deb11u2            all          Apache HTTP Server (common files)
ii  apache2-utils            2.4.56-1~deb11u2            amd64        Apache HTTP Server (utility programs for web servers)
ii  libapache2-mod-php       2:7.4+76                    all          server-side, HTML-embedded scripting language (Apache 2 module) (default)
ii  libapache2-mod-php7.4    7.4.33-1+deb11u3            amd64        server-side, HTML-embedded scripting language (Apache 2 module)
```

My apache status is

```
* apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-06-01 10:35:11 CEST; 16min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 415 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 462 (apache2)
      Tasks: 9 (limit: 4661)
     Memory: 24.0M
        CPU: 302ms
     CGroup: /system.slice/apache2.service
             |-462 /usr/sbin/apache2 -k start
             |-477 /usr/sbin/apache2 -k start
             |-478 /usr/sbin/apache2 -k start
             |-479 /usr/sbin/apache2 -k start
             |-480 /usr/sbin/apache2 -k start
             |-481 /usr/sbin/apache2 -k start
             |-726 /usr/sbin/apache2 -k start
             |-804 sh -c systemctl status apache2
             `-805 systemctl status apache2
```

My postgresql install is

```
ii  postgresql               13+225                      all          object-relational SQL database (supported version)
ii  postgresql-13            13.10-0+deb11u1             amd64        The World's Most Advanced Open Source Relational Database
ii  postgresql-client-13     13.10-0+deb11u1             amd64        front-end programs for PostgreSQL 13
ii  postgresql-client-common 225                         all          manager for multiple PostgreSQL client versions
ii  postgresql-common        225                         all          PostgreSQL database-cluster manager
```

My postgresql status is

```
* postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
     Active: active (exited) since Thu 2023-06-01 10:35:14 CEST; 16min ago
    Process: 534 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 534 (code=exited, status=0/SUCCESS)
        CPU: 1ms
```

My ssh install is

```
ii  libssh2-1:amd64          1.9.0-2                     amd64        SSH2 client-side library
ii  openssh-client           1:8.4p1-5+deb11u1           amd64        secure shell (SSH) client, for secure access to remote machines
ii  openssh-server           1:8.4p1-5+deb11u1           amd64        secure shell (SSH) server, for secure access from remote machines
ii  openssh-sftp-server      1:8.4p1-5+deb11u1           amd64        secure shell (SSH) sftp server module, for SFTP access from remote machines
ii  task-ssh-server          3.68+deb11u1                all          SSH server
```

My ssh status is

```
* ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-06-01 10:35:11 CEST; 16min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 419 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 457 (sshd)
      Tasks: 1 (limit: 4661)
     Memory: 5.4M
        CPU: 98ms
     CGroup: /system.slice/ssh.service
             `-457 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
```

Disk space at end of installation:

```
camusmat@server-camusmat:~$ df -h | grep sda1
/dev/sda1       3.0G  1.4G  1.5G  49% /
```