IUT2 Université Grenoble Alpes

Workstation installation guide

debian

Célian Yckache A1

Table des matières

This is a guide that helps you to install your own workstation. It will show you how to install the whole system without a graphic interface and with the most important packages that you need for database management. The Linux system that we are going to install is Debian 11, for x86 64 bits processors with the netinst iso image on a virtual machine Qemu/KVM.

1. The iso image

First you have to find the iso image on another computer on this website :
https://cdimage.debian.org/cdimage/release/current/amd64/iso-cd/

You have to choose this one : 

Download this file too : 

And then compare them to verify the integrity of iso image.
If they are the same, you can begin the installation. You must use the command “S2.04-lance-installation”, this command uses the “S2.03-commun” file. This file contains a script that loads the system. There is the script describes :

```
#!/bin/bash

# We create the image on the local disk initially.
# After installation, we move it to Erebus4.

image_locale="/donnees/TP-infobut1/Debian-S2.03-$LOGNAME.img"
image_nfs="/users/Stockage-HDD/images-kvm/S2.03/images/Debian-S2.03-
$LOGNAME.img"

# We have 4 cases
# This instruction verifies if the local image is the same as the one in the server erebus4, if
# not, the script displays a message and stop everything
if [ -e "$image_nfs" ] && [ -e "$image_locale" ]
then
        echo "Situation anormale:"
        echo "Vous avez une image locale sur cette station Linux et une autre sur erebus4"
        echo "Veuillez en supprimer une et recommencer..."
        exit
fi

# This one verifies if there are no one of the two images and if it’s true, the script displays a
# message and create a local disc image.
if [ ! -e "$image_nfs" ] && [ ! -e "$image_locale" ]
then
        echo "Création d'une image disque locale sur cette station Linux ..."
        image="$image_locale"
        qemu-img create "$image" 4G
        sync
        echo "Fini."
```

```
fi

# This third one verifies if the server image is already existing, if it’s true, the script displays a
# message and assigns the image path to the variable “image”.
if [ -e "$image_nfs" ]
then
        echo "Image trouvée sur le serveur erebus4:"
        image="$image_nfs"
        echo "$image"
fi

# This fourth one verifies if the local image already exists, if it’s true, the script displays
# a message and assigns the image path to the variable “image”.
if [ -e "$image_locale" ]
then
        echo "Image trouvée sur le disque de cette station Linux:"
        image="$image_locale"
        echo "$image"
fi

drive="format=raw,file=$image,discard=unmap"

# Image ISO
# The definition of the variable iso, containing the path of the ISO image, is obtained by
# listing the corresponding file in the directory /usr/local/images-ISO using the naming
# pattern debian-*-netinst.iso.
iso=$(ls /usr/local/images-ISO/debian-*-netinst.iso)

# Command launching Qemu
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"

# The last instruction defines a variable “lance_qemu”, it launch the software Qemu with
# differents parameters like such as the virtual machine used (`q35`), the processor (`cpu
# host`), the memory (`4G`), enabling KVM virtualization (`-enable-kvm`), the VGA graphics
# card with a resolution of `1024x768`, GTK display without automatic resizing (`-display
# gtk,zoom-to-fit=off`), the virtual hard drive specified by the variable `drive`, the network
# card (`e1000`) with specified port forwarding (`hostfwd`).
```

2. Configuration of the installation

Second, you must configure your system.
When the landing is complete you must choose “install without graphic”, then you must configure the system. Follow the following steps, if the choice is not specified, you have to choose the default one :

Language : English

Location : other/Europe/France

Locales : United States, en_US.UTF-8

Keyboard : French

Hostname : use server-"Your-login" like this :

Root Password : The password has to be one you'll remember like “root”, you can use “show password” to be sure.

User Account - Full Name : your entire name, for example "Jean Toto"

User Name : use your UGA login

User Password : Enter a simple password like “etu”. Check the "Show Password" box to make sure that the entered password is the one you want.

Partition disks : Guided - use entire disk

Partition disks : All files in one partition

Partition disks : Yes

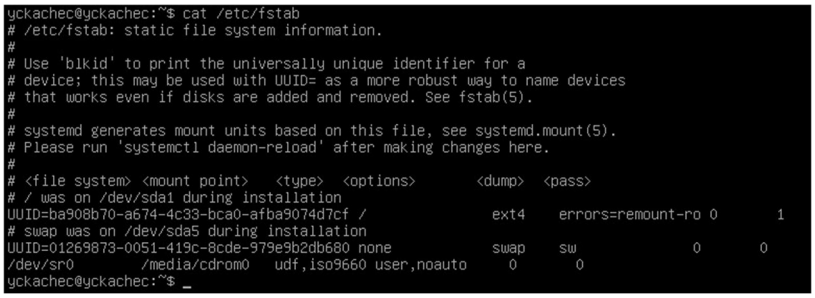
Software Selection : To check if "Debian desktop" is not checked and "ssh server" is checked

Install GRUB : Yes

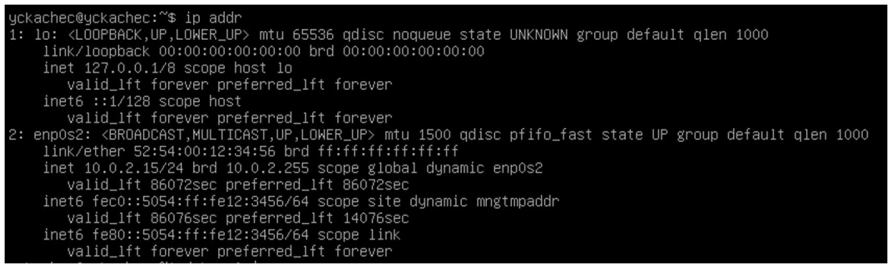
Device for boot loader : /dev/sda

3. Configuration of the system and verifications

After waiting for all the installations to complete, you must verify if the virtual machine is functional. First verify if you have the same result after execution of this command : "cat /etc/fstab"

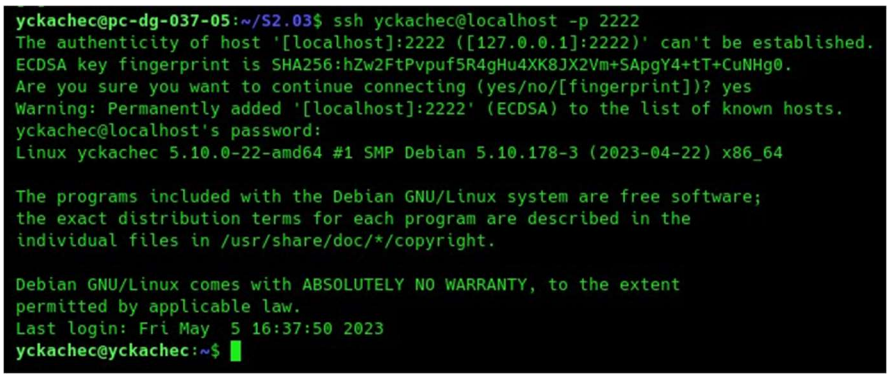


Now check the virtual machine ip address and ethernet and verify that you can access to the outside from it, you can use the command “ip addr”. Your result have to look like this :

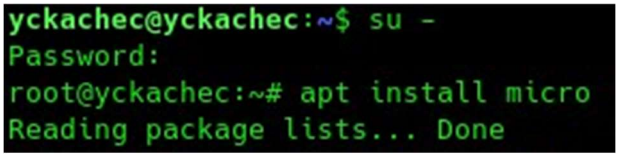


And now, check if you can't access the xorg server, it is used for graphics interface, that is not the case for you. So, use the command "dpkg -l | grep xorg", and nothing must append.

Finally, you must check if you can access your virtual machine from the physical one. You can try to connect by using this command : "ssh yourLogin@localhost -p 2222", replace yourLogin by the login you used in hostname step when you did the configuration :



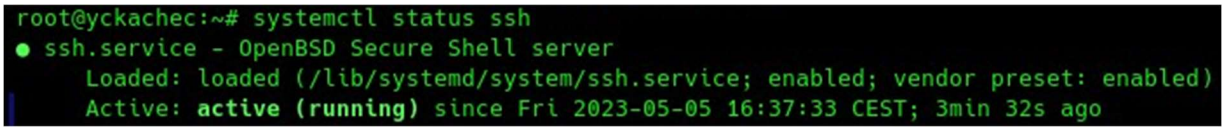
Now you are connected, you can access the root account with this command : "su -" and enter your root password. Now try to install packages like "micro" text editor to check if you can install packages from the outside.



4. Installation of most importants packages

1) SSH

You have to install ssh package, you only need to use this command : "apt install ssh"
After you have to check if it's running, try the command "systemctl status ssh" if it's not running, you can restart it with the command : "systemctl restart ssh"
There is the result of the status if it is running :



2) Apache2

You have to install apache2 package, you only need to use this command : “apt install apache2”
After you have to check if it’s running, try the command “systemctl status apache2” if it’s not running, you can restart it with the command : “systemctl restart apache2”
There is the result of the status if it is running :

```
root@yckachec:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2023-05-05 16:43:18 CEST; 58s ago
```

You are not on a graphic interface, so you can’t visualize an html page, but you can connect to the apache server and see if it works. To do that, use this command : “telnet localhost 80” and add this characters : HEAD / HTTP/1.0 like this :

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

HTTP/1.1 200 OK
Date: Fri, 05 May 2023 14:48:37 GMT
Server: Apache/2.4.56 (Debian)
Last-Modified: Fri, 05 May 2023 14:43:17 GMT
ETag: "29cd-5faf34f3e461b"
Accept-Ranges: bytes
Content-Length: 10701
Vary: Accept-Encoding
Connection: close
Content-Type: text/html

Connection closed by foreign host.
```

You can also open the apache server on your local machine, to do it you just need to type http://localhost:8080 in your browser. If everything work, you will come to this page :

3) PostgreSQL

To perform database management, you need to install the PostgreSQL package. As the others packages you just installed, you can use apt library with this command : “apt install posgresql”, then you can check if it’s running or not with the root command “systemctl status postgresql” :

```
root@yckachec:~# 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 16:09:10 CEST; 1min 49s ago
```

And if not, you can restart it with the command “systemctl restart postgresql”

To check the install, you can connect to postgresql with the login postgres with this command in root : “su - postgres”. You can now see the default databases with “psql -l” as you can see :

```
root@yckachec:~# su - postgres
postgres@yckachec:~$ 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@yckachec:~$
```

You can also do it from the local machine if you want.
Now you must create a user with your login. You just must type the command “CREATE USER YOUR_LOGIN” and replace YOUR_LOGIN with your login uga.
You can create your own database, to do it you can type “CREATE DATABASE mabase WITH OWNER = YOUR_LOGIN”.
You must check if they have been created successfully :

```
postgres@yckachec:~$ psql -l
                                  List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
-----------+----------+----------+-------------+-------------+-----------------------
 mabase    | yckachec | 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)
```

You can add some data in your database, you can create tables and fill them in like this :

```
mabase=# create table test( fname varchar, lname varchar);
```

```
mabase=# select * from test ;
 fname | lname
-------+-------
(0 rows)

mabase=# insert into test values ('celian', 'yckache')
mabase-# ;
INSERT 0 1
mabase=# select * from test ;
 fname  |  lname
--------+---------
 celian | yckache
(1 row)

mabase=# _
```

You can also do it from your local machine :

```
yckachec@mabase=> select * from test;
 fname | lname
-------+-------
(0 rows)

yckachec@mabase=> \d
          List of relations
 Schema |  Name   | Type  |  Owner
--------+---------+-------+----------
 public | matable | table | postgres
 public | test    | table | postgres
(2 rows)
```

To access this new user from the outside, you must change some settings, like for the password. First type this command :

```
root@yckachec:~# nano /etc/postgresql/13/main/postgresql.conf
```

Second, you must modify this white line :

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

# - Connection Settings -

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

Now there is another file to modify :

```
root@yckachec:~# nano /etc/postgresql/13/main/pg_hba.conf _
```

You have to add this line at the end of the text file :

```
#IPv4 remote connections:
host  all all 0.0.0.0/0 scram-sha-256
```

And replace all of the md5 by scram-sha-256 (the MD5 hash is considered obsolete) :

```
cal" is for Unix domain socket connections only
   all            all                                     peer
4 local connections:
   all            all            127.0.0.1/32             scram-sha-256
6 local connections:
   all            all            ::1/128                  scram-sha-256
ow replication connections from localhost, by a user with the
lication privilege.
   replication    all                                     peer
   replication    all            127.0.0.1/32             scram-sha-256
   replication    all            ::1/128                  scram-sha-256
 remote connections:
 all all 0.0.0.0/0 scram-sha-256



yckachec:~# service postgresql restart
```

As you can see at the bottom of this screenshot, you must restart postgresql with this command.
Now modify your password as follows :

```
postgres@yckachec:~$ psql
psql (13.10 (Debian 13.10-0+deb11u1))
Type "help" for help.

postgres=# alter user yckachec password 'etu';
ALTER ROLE
postgres=#
```

Now, in order to verify if everything worked, connect to the PostgreSQL database and enter the following command :

```
postgres=# select * from pg_shadow ;
 usename  | usesysid | usecreatedb | usesuper | userepl | usebypassrls |
       passwd                                                                | valuntil | useconfig
-----------+----------+-------------+----------+---------+--------------+-----------------------------------------------------------------------
-------------------------------------------------------------------------+----------+-----------
 postgres |       10 | t           | t        | t       | t            |
                                                                           |          |
 yckachec |    16384 | f           | f        | f       | f            | SCRAM-SHA-256$4096:stCyrmhd8euoAyYg8WHSTg==$pGdisWQLrAs
zZTm1zIl7CTCtbzA/RcmVty2AQKMeHYo=:l9PLEprb2e7KLh6xpcYuef/hDqtLGE42m+pVVs/GoqM= |          |
(2 rows)
```

4) PHP

First install php package on your virtual machine with this command :

```
# apt install php-common libapache2-mod-php php-cli
```

To check the installation, create a php file name info.php with these lines :
You need to place it in the directory /var/www/html/. Type in your browser http://localhost:8080/info.php :

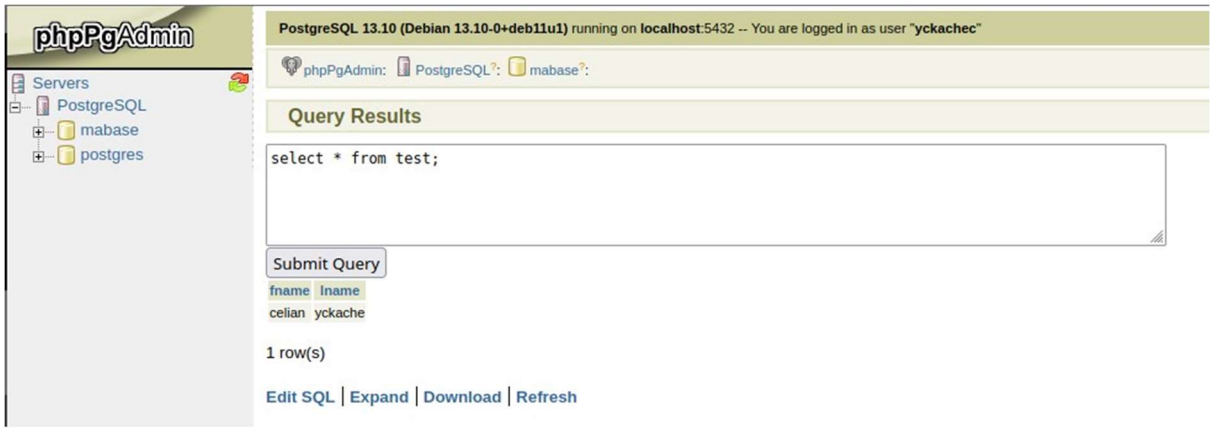
If this appears, everything's working.

5) PhpPgAdmin

To have another interface to query your database, you can install PhpPgAdmin.
In the root, use the command : “apt install phppgadmin”. To access this interface, follow these steps :

- Type http://localhost:8080/phppgadmin/
- Click on “servers” and “PostgreSQL”
- Connect with your login and password
- Then you can access to your databases

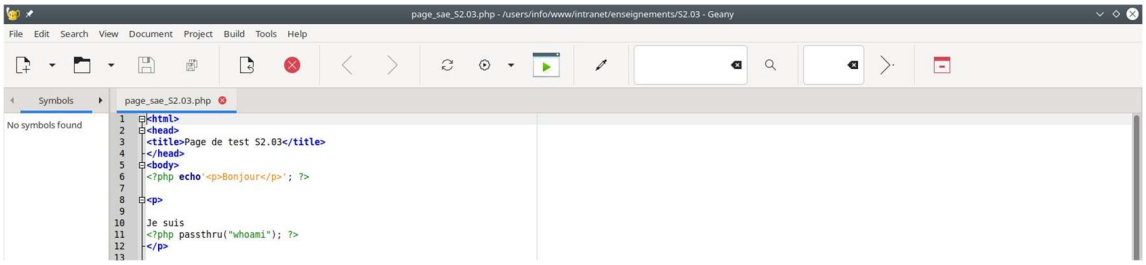
It must look like this when you type a request :



You can now access to php and use it, after execute this command :

```
root@yckachec:~# /sbin/blkid
/dev/sda1: UUID="ba908b70-a674-4c33-bca0-afba9074d7cf" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="f9661100-01"
/dev/sda5: UUID="01269873-0051-419c-8cde-979e9b2db680" TYPE="swap" PARTUUID="f9661100-05"
root@yckachec:~#
```

You can answer this php file :
“/users/info/www/intranet/enseignements/S2.03/page_sae_S2.03.php”



5. In the end

In the end, you can check if there is enough space available in your virtual machine :

```
root@yckachec:~# df -H
Filesystem      Size  Used Avail Use% Mounted on
udev            2.1G     0  2.1G   0% /dev
tmpfs           412M  508k  411M   1% /run
/dev/sda1       3.2G  1.5G  1.6G  49% /
tmpfs           2.1G   17k  2.1G   1% /dev/shm
tmpfs           5.3M     0  5.3M   0% /run/lock
tmpfs           412M     0  412M   0% /run/user/1000
root@yckachec:~#
```

Now your system is successfully installed and configured, you can use it for your work.