

SAE 2.03

Installation Guide

English version

Milwenn **FRANCEUS-COINTREL**

Julien **RENAUD**

Héloïse **RIGAUX**

Alexandre **DESCHANEL**

Nathan-Raphaël **LOPEZ-ROJAS**

S1

Preamble

This document indexes the various installation steps of the different elements composing a **LAMP** stack destined for coworking spaces.

In order to create our **LAMP** stack, the chosen elements are the following :

- The **VMware Workstation** software to create our virtual machine
- The **Ubuntu 22.04.4 LTS** distribution for the operating system
- The **Apache** web server along with its **PHP** module
- The **MySQL** software for database management

Once the virtual machine is installed on your local computer, make sure to have administrator (or **root**) privileges. In this guide, we will consider that you have admin and installation rights. Before installing the softwares, make sure that your operating system is up to date by entering the following command on the terminal¹ :

```
sudo apt update && sudo apt dist-upgrade -y
```

I. Virtual Machine

Before beginning the installation of the virtual machine, you will need to download the ISO image file of the desired operating system (here we choose [Ubuntu](#)) and the [VMware Workstation](#) software.

A. Installation of the virtual machine

1. On **VMware**, click on **File**, then on **New Virtual Machine**, then on continue
2. Check the third option (for installing the ISO later on), then click on continue
3. Check **Linux**, and select **Ubuntu** for the version, then click on continue
4. Give a name to the virtual Machine, then click on continue
5. Enter the desired storage capacity, then click on continue
6. Click on the **Customize Hardware** button
7. In the following categories :
 - **Memory** : enter the desired amount of RAM (4GB minimum)
 - **Processors** : choose the number of cores you want to allocate
 - **CD/DVD** : check "Use ISO image file", then select your ISO file
8. Click on **Close, Finish** and start the VM

¹ CTRL + T to open a terminal on LINUX

B. Installation of the operating system

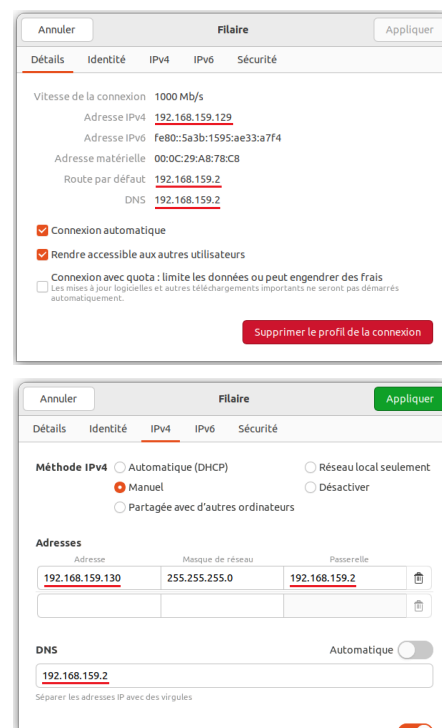
You can install any LINUX distribution you desire with or without a GUI. In this guide, we chose to install Ubuntu with its GUI as it is user-friendly.

9. After booting the VM, select **Try or Install Ubuntu**
10. Select the English language, then click on **Install Ubuntu**
11. Select your keyboard layout, then click on continue
12. Select **Normal Installation**, then click on continue
13. In **Installation type**, select **Erase disk and install Ubuntu**, click on **Install Now**
14. Select your time zone, enter your personal information, and click on finish

C. Fixed IP address assignment

15. On Ubuntu, in the network settings, set the **IPv4** method to **Manual** to assign a fixed IP address to be able to communicate with the host. Make sure you assign a valid and available IP address, netmask, gateway and DNS, as shown in the example on the right.

⇒ **NB:** To access via SSH/SFTP, execute the following command :
`sudo apt install openssh-server`



II. HTTP web server + PHP module

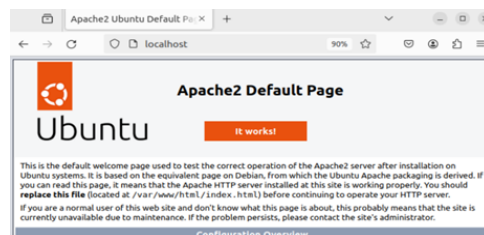
A. Core installation & configuration of Apache web server

1. Open a terminal and install *Apache* via the following command :

`sudo apt install apache2 -y`

⇒ Check that the web server was correctly installed by opening the web browser and using the following link : localhost. You should see the default *Apache* web page.

2. Assign ownership of the `/var/www/html` directory to your current user in order to be able to upload your site with the following command : `sudo chown -R $(whoami) /var/www/html`



- To change the port which the web server is listening to, you can edit line **5** of the **/etc/apache2/ports.conf** file, where the **Listen** attribute is mentioned. Replace **80** with the desired available port. If you do so, make sure you edit as well the port mentioned in the first line of the configuration file **/etc/apache2/sites-available/000-default.conf**.

⇒ After all the modifications, restart the apache2 service by executing the following command : **sudo systemctl restart apache2**

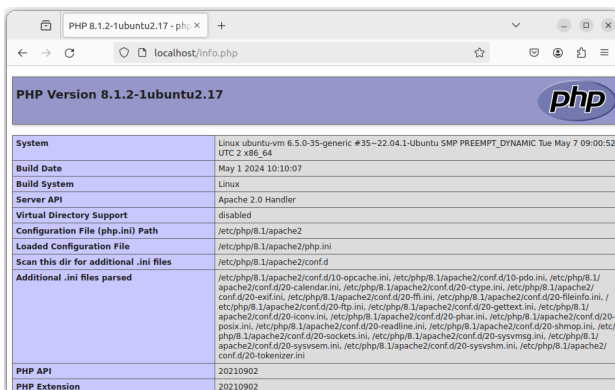
Verify the newly selected port on the web browser.

B. PHP module installation

- Install the *PHP* module and its dependencies via the following command :

sudo apt install libapache2-mod-php php-mysql -y

⇒ To test if the module works, you can generate the PHP information page file with the following command : **echo "<?php phpinfo();?>" > /var/www/html/info.php**. Go back to the same previous link on the browser and add **/info.php** at the end of it. Your page must look like the following :



PHP Version 8.1.2-1ubuntu2.17	
System	Linux ubuntu-vm 6.5.0-35-generic #35~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Tue May 7 09:00:52 UTC 2 x86_64
Build Date	May 1 2024 10:10:07
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-ldap.ini, /etc/php/8.1/apache2/conf.d/20-mbstring.ini, /etc/php/8.1/apache2/conf.d/20-mysql.ini, /etc/php/8.1/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini
PHP API	20210902
PHP Extension	20210902

Beware : Never leave a phpinfo page publicly accessible. To delete it you can execute the succeeding command :

rm /var/www/html/info.php

III. MySQL database server

A. Core Installation MySQL server

- Install **MySQL Community 8.4.0**, by executing the following command on a terminal : **sudo apt install mysql-server -y**
- Once the installation is finished, if you want to change the port and the ip address used to access the database, turn off the **mysql** service by running this command : **sudo systemctl stop mysql.service**. Otherwise, skip to step 4.
- Edit the corresponding lines of the default **mysqld.cnf** configuration file located in **/etc/mysql/mysql.conf.d/**. You can use the **nano** text editor on the terminal by running the following command : **sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf**

- a. To change the access port, go to line **21**, to the commented **port** attribute. Uncomment the line by deleting the hash sign, then change the port number on the right side of the '=' character. In our example, we use port **24750**. Make sure to use an available port.

```
GNU nano 6.2 /etc/mysql/mysql.conf.d/mysqld.cnf *
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# Here is entries for some specific programs
# The following values assume you have at least 32M ram
[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file           = /var/run/mysqld/mysqld.pid
# socket             = /var/run/mysqld/mysqld.sock
port                = 24750
# datadir            = /var/lib/mysql
```

- b. To change the access ip address go to line **31**, to the **bind-address** attribute. Enter the new ip address on the right side of the '=' character. Make sure to use the ip address of a valid host. In our example, we'll stay with the default ip address : **127.0.0.1**.

```
GNU nano 6.2 /etc/mysql/mysql.conf.d/mysqld.cnf *
port                = 24750
# datadir            = /var/lib/mysql
#
# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html
#
# tmpdir              = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address         = 10.0.0.1
mysqlx-bind-address  = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size      = 16M
# max_allowed_packet = 64M
# thread_stack        = 256K
# thread_cache_size   = -1
```

- c. Restart the mysql service by executing the following command :

sudo systemctl start mysql

B. Super-user root configuration

4. Secure the MySQL server by deactivating some parameters by running the following command : **sudo mysql_secure_installation**. Follow these steps in order :

- The program will ask you if you want to enable the secure passwords component. Enter **Y** to accept and choose the level of security between **0** and **2**. Otherwise, enter **N** to decline.
- Then, delete anonymous users by typing **Y** in the terminal.
- Furthermore, the program will ask you if you want to disable connection to the super-user outside of the local machine. It is recommended to disable it by typing **Y**. Otherwise type **N**.
- Next, delete the test database by typing **Y**.
- Finally, reload the privilege tables by typing **Y** in the terminal.

5. Change the super-user's password by executing the succeeding commands :

- **sudo mysql**

⇒ This command opens a terminal with administrator rights.

NB : Starting from the next step, all command execution will send a response, noticing if the execution was successful or has failed. If successful the response should take the following form '**Query OK, x rows affected**'. If the execution fails, make sure that you did not forget any steps or restart the MySQL server installation from scratch.

- **ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'your_password';**
⇒ Replace the **your_password** field with a secured password. This mysql command sets the given password to the super-user.
- **exit**
⇒ This command allows you to exit the mysql terminal.

C. Database and user creation

*For security reasons, it's a standard practice to use one user **per** service, with **specific privileges**. For our coworking case, we'll create a user named **web** with **select, insert, update** and **delete** privileges on a database named **webphpdb**.*

6. Access the mysql terminal again via the new secure configuration by executing the following command : **mysql -u root -p**
⇒ Once the command has been executed, the terminal will prompt you for a password. Use your superuser password.
7. Create your new user with the followed command : **CREATE USER 'web'@'%' IDENTIFIED BY 'your_password';**
⇒ Replace the **your_password** field with your new user's password. Make sure you don't use the same password as the superuser.
8. Create the database to link to the user with the following command : **CREATE DATABASE webphpdb;**
9. Assign the necessary privileges to the corresponding database user via the following command : **GRANT SELECT, INSERT, UPDATE, DELETE ON webphpdb.* TO 'web'@'%';**
⇒ This command assigns the **selection, insertion, update**, and **delete** privileges in the **webphpdb** database for the web user.
10. Exit the mysql terminal by executing : **exit**