

Mise en place d'un environnement type pour l'équipe de développement web (Debian, apache, mariaDB)

Table des matières

- Mission :.....	2
- Création d'un master de base Debian.....	2
- Création du serveur apache PHP phpmyadmin.....	5
Installation d'apache2 :.....	5
Installation de PHP.....	6
Installation des paquets PHP pour mariaDB.....	7
- Création du serveur de base de données mariaDB.....	7
Ajout d'un utilisateur pour l'administration.....	7
Permettre l'accès a distance sur le serveur mariaDB.....	8

- Mission :

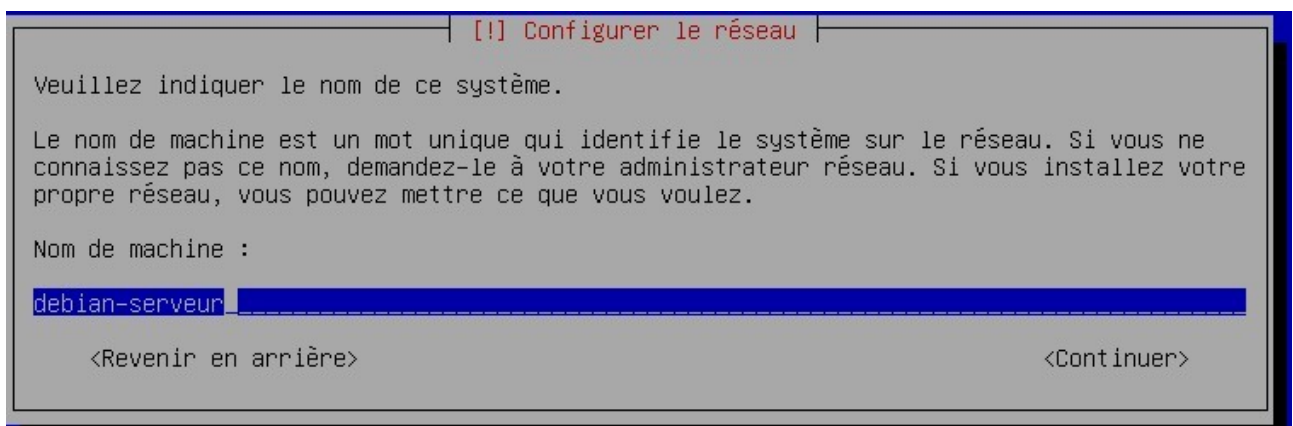
préparer un environnement Web type pour la réalisation et les tests des applications Web destinées à l'équipe ANNA ou aux élèves et aux enseignants.

Nous aurons deux serveurs, un premier avec apache2, PHP, et PHPmyadmin
un deuxième avec mariaDB. Tous deux tournant sur debian.

- Création d'un master de base Debian

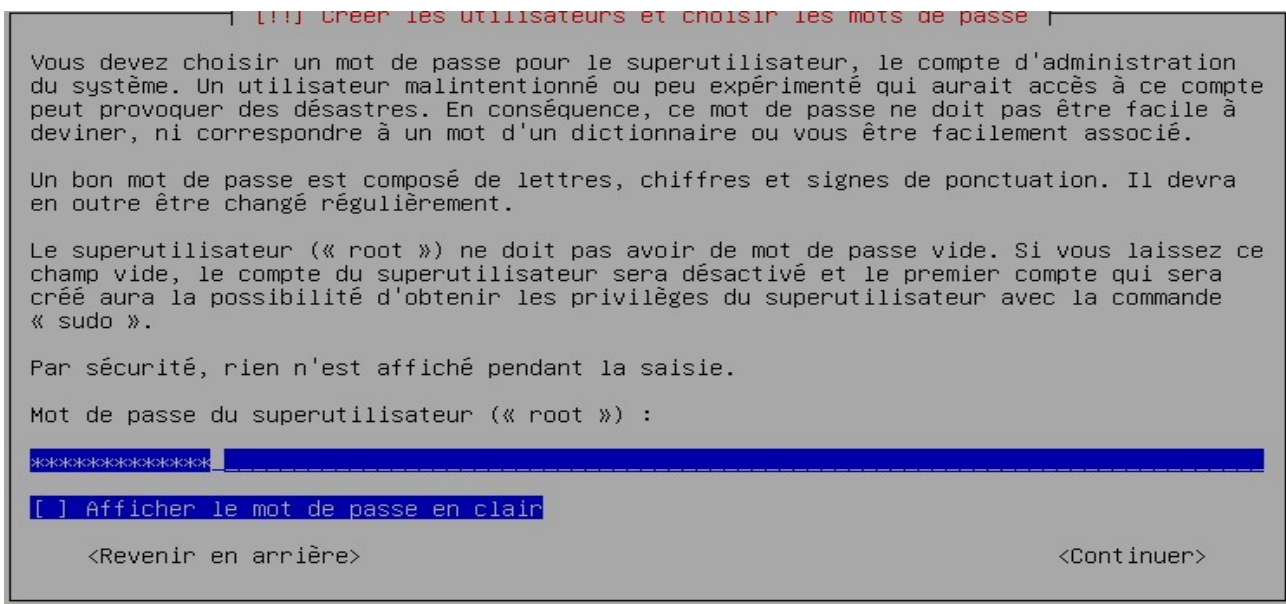
Nous allons procéder à une installation basique de Debian afin de pouvoir la répliquer selon les besoins.

Au premier écran notable nous entrons un nom, ici debian-serveur



The screenshot shows a terminal window titled "[!] Configurer le réseau". The text inside reads: "Veuillez indiquer le nom de ce système. Le nom de machine est un mot unique qui identifie le système sur le réseau. Si vous ne connaissez pas ce nom, demandez-le à votre administrateur réseau. Si vous installez votre propre réseau, vous pouvez mettre ce que vous voulez. Nom de machine :". Below the text, the input field contains "debian-serveur". At the bottom, there are two buttons: "<Revenir en arrière>" and "<Continuer>".

Un mot de passe solide pour l'utilisateur root



The screenshot shows a terminal window titled "[!] Créer les utilisateurs et choisir les mots de passe". The text inside reads: "Vous devez choisir un mot de passe pour le superutilisateur, le compte d'administration du système. Un utilisateur malintentionné ou peu expérimenté qui aurait accès à ce compte peut provoquer des désastres. En conséquence, ce mot de passe ne doit pas être facile à deviner, ni correspondre à un mot d'un dictionnaire ou vous être facilement associé. Un bon mot de passe est composé de lettres, chiffres et signes de ponctuation. Il devra en outre être changé régulièrement. Le superutilisateur (« root ») ne doit pas avoir de mot de passe vide. Si vous laissez ce champ vide, le compte du superutilisateur sera désactivé et le premier compte qui sera créé aura la possibilité d'obtenir les privilèges du superutilisateur avec la commande « sudo ». Par sécurité, rien n'est affiché pendant la saisie. Mot de passe du superutilisateur (« root ») :". Below the text, the input field is filled with asterisks. At the bottom, there is a checkbox labeled "[] Afficher le mot de passe en clair" and two buttons: "<Revenir en arrière>" and "<Continuer>".

Pour ce qui est du nom de domaine nous laissons par défaut

[I] Configurer le réseau

Le domaine est la partie de l'adresse Internet qui est à la droite du nom de machine. Il se termine souvent par .com, .net, .edu, ou .org. Si vous paramétrez votre propre réseau, vous pouvez mettre ce que vous voulez mais assurez-vous d'employer le même nom sur toutes les machines.

Domaine :

<Revenir en arrière> <Continuer>

On crée un utilisateur et on définit son mot de passe

[III] Créer les utilisateurs et choisir les mots de passe

Un compte d'utilisateur va être créé afin que vous puissiez disposer d'un compte différent de celui du superutilisateur (« root »), pour l'utilisation courante du système.

Veuillez indiquer le nom complet du nouvel utilisateur. Cette information servira par exemple dans l'adresse origine des courriels émis ainsi que dans tout programme qui affiche ou se sert du nom complet. Votre propre nom est un bon choix.

Nom complet du nouvel utilisateur :

BTSSIO

<Revenir en arrière> <Continuer>

Pour la partie partitionnement on utilise un disque entier et tout dans une seule partition

[II] Partitionner les disques

Le programme d'installation peut vous assister pour le partitionnement d'un disque (avec plusieurs choix d'organisation). Vous pouvez également effectuer ce partitionnement vous-même. Si vous choisissez le partitionnement assisté, vous aurez la possibilité de vérifier et personnaliser les choix effectués.

Si vous choisissez le partitionnement assisté pour un disque complet, vous devrez ensuite choisir le disque à partitionner.

Méthode de partitionnement :

Assisté - utiliser un disque entier
Assisté - utiliser tout un disque avec LVM
Assisté - utiliser tout un disque avec LVM chiffré
Manuel

<Revenir en arrière>

[II] Partitionner les disques

Disque partitionné :

SCSI1 (0,0,0) (sda) - ATA VBOX HARDDISK: 34.4 GB

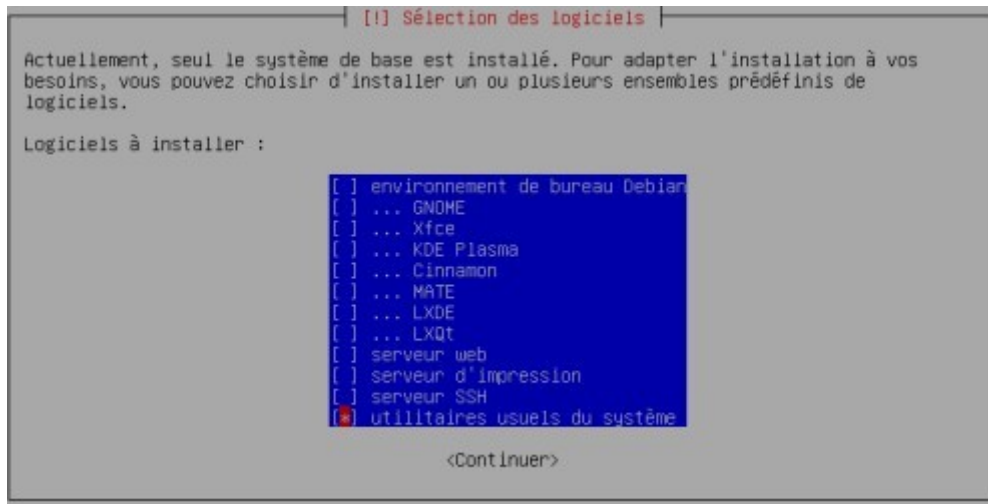
Le disque peut être partitionné selon plusieurs schémas. Dans le doute, choisissez le premier.

Schéma de partitionnement :

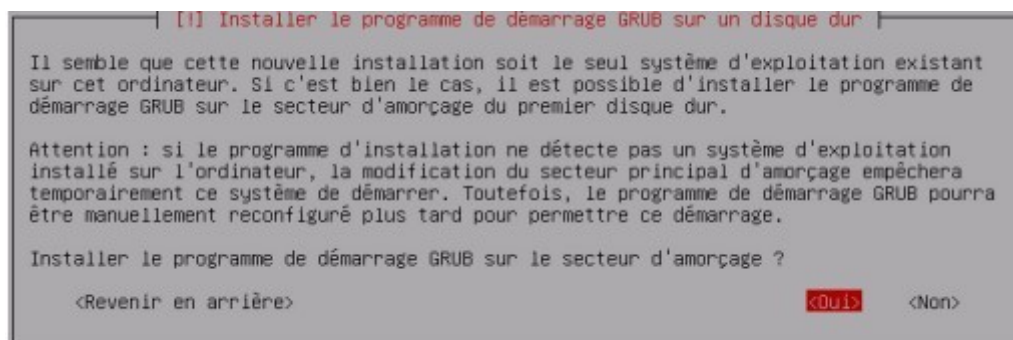
Tout dans une seule partition (recommandé pour les débutants)
Partition /home séparée
Partitions /home, /var et /tmp séparées

<Revenir en arrière>

Pour la sélection des logiciels nous ne laissons que les utilitaires usuels du système, il n'y aura pas d'interface graphique.



Pour terminer nous installons GRUB sur le secteur d'ammorçage, soit le seul disque dur present



Nous pouvons désormais créer nos deux serveurs.

- Création du serveur apache PHP phpmyadmin


Depuis un clone de notre master Debian, nous commençons par changer son hostname en modifiant le fichier */etc/hostname* , on le nommera SRV-WEB, on fera de même dans le fichier */etc/hosts*

Nous pouvons maintenant commencer l'installation d'**apache** et **PHP**

Installation d'apache2 :

- apt install apache2

Le service est installé. Le site web de test est situé dans */var/www/html* et est visible en tapant l'ip du serveur dans un navigateur du réseau.



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at */var/www/html/index.html*) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in */usr/share/doc/apache2/README.Debian.gz***. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the *apache2-doc* package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- *apache2.conf* is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- *ports.conf* is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the *mods-enabled/*, *conf-enabled/* and *sites-enabled/* directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective **-available/* counterparts. These should be managed by using our helpers *a2enmod*, *a2dismod*, *a2ensite*, *a2dissite*, and *a2enconf*, *a2disconf* . See their respective man pages for detailed information.
- The binary is called *apache2*. Due to the use of environment variables, in the default configuration, *apache2* needs to be started/stopped with */etc/init.d/apache2* or *apache2ctl*. **Calling */usr/bin/apache2* directly will not work with the default configuration.**

- Les logs sont disponible dans ***/var/log/apache2***

Installation de PHP

- apt install PHP

Pour vérifier que apache exécute bien les scripts PHP, nous pouvons créer un fichier de test dans le dossier ***/var/www/html***

- création du fichier :

vi /var/www/html/test.php

puis comme contenu du fichier :



<?PHP

phpinfo();

?>

Dans un navigateur nous pouvons maintenant taper l'adresse ip_serveur/test.php

le script s'affiche bien

PHP Version 7.4.33	
	
System	Linux SRV-WEB 5.10.0-21-amd64 #1 SMP Debian 5.10.162-1 (2023-01-21) x86_64
Build Date	Nov 8 2022 11:40:37
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-fli.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902.NTS
PHP Extension Build	API20190902.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib *, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*
This program makes use of the Zend Scripting Language Engine: Zend Engine v3.4.0, Copyright (c) Zend Technologies with Zend OPcache v7.4.33, Copyright (c), by Zend Technologies	
	

Installation des paquets PHP pour mariaDB

Pour que les applications PHP puissent utiliser la base de données mariaDB il faut encore installer deux paquets , un module pour apache et un pour PHP

apt install libapache2-mod-php php-mysql

- Création du serveur de base de données mariaDB

Sur un nouveau clone du master Debian que nous renommons SRV-BDD, nous installons le paquet mariaDB :

apt install mariadb-server

une fois installé nous lançons la commande suivante pour finaliser la configuration :

mysql_secure_installation

Nous aurons a répondre a plusieurs questions, comme préciser le mot de passe root de mariaDB (ce n'est pas celui de Debian) .

A la question **Switch to unix_socket authentication**, cela autorisera uniquement les connexions sur la base de données MariaDB depuis le serveur lui même. Donc nous répondrons N car le service PHP n'est pas installé sur le même serveur.

Pour le reste nous répondrons Y pour supprimer les base de test et utilisateurs anonymes.

Ajout d'un utilisateur pour l'administration

Pour pouvoir gérer la base de données nous allons ajouter un utilisateur nommé admin

mariadb -u root -e

CREATE USER 'admin'@'%' IDENTIFIED BY 'P\$\$\$word' ;

GRANT ALL PRIVILEGES ON *.* to admin@'%' WITH GRANT OPTION;

FLUSH PRIVILEGES ;

Permettre l'accès a distance sur le serveur mariaDB

Pour pouvoir accéder au serveur mariaDB depuis un autre ordinateur, il faut modifier la ligne **bind-address=127.0.0.1** par **bind-address = 0.0.0.0** dans le fichier **/etc/mysql/mariadb.conf.d/50-server.conf**

Enfin, nous redémarrons le service mariaDB avec la commande **systemctl restart mariadb**

Nous pouvons vérifier que le service est bien en écoute avec la commande **ss -nlt**

```
gestion@SRV-BDD:~$ ss -nlt
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port
LISTEN     0            80          0.0.0.0:3306             0.0.0.0:*
LISTEN     0           128          0.0.0.0:22              0.0.0.0:*
LISTEN     0           128          [::]:22                 [::]:*
```

mariadb utilise le port 3306.

A ce stade le serveur mariaDB est installé et prêt a être utilisé, nous pouvons essayer de nous connecter pour vérifier le bon fonctionnement de mariaDB

```
root@SRV-BDD:~# mariadb -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 30
Server version: 10.5.18-MariaDB-0+deb11u1 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0,005 sec)

MariaDB [(none)]>
```

Nos deux serveurs sont maintenant opérationnels pour l'équipe de développement web