

# 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 :**

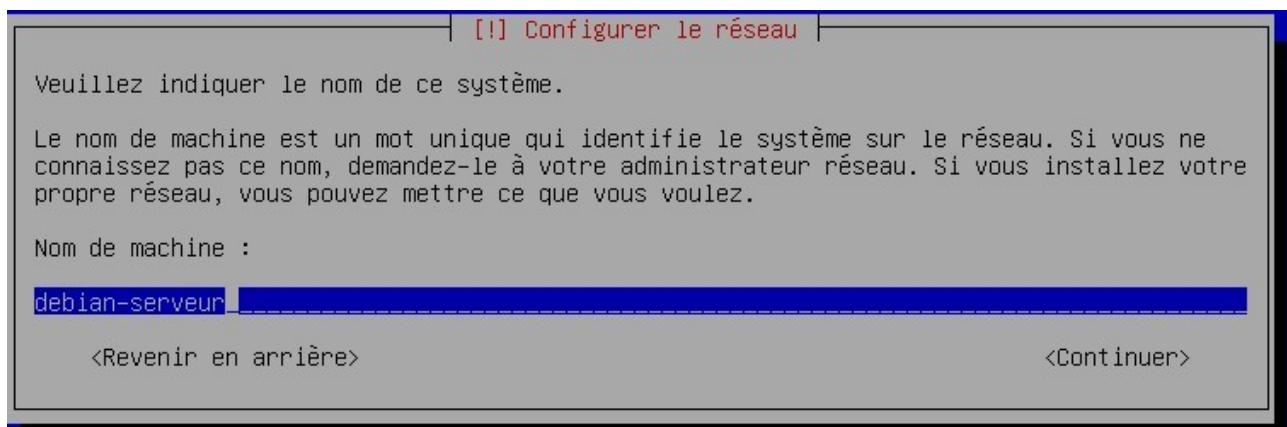
préparer un environnement Web type pour la réalisation et les tests des applications Web destinées à l'équipe ANNA ou aux élèves et aux enseignants.

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

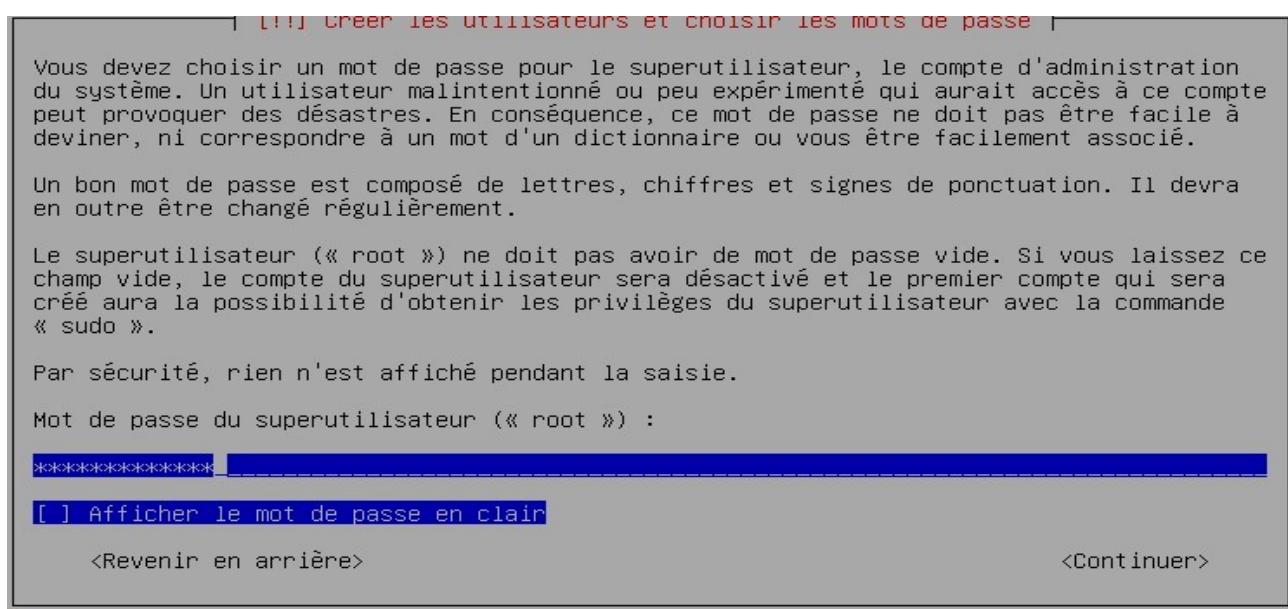
## - **Création d'un master de base Debian**

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

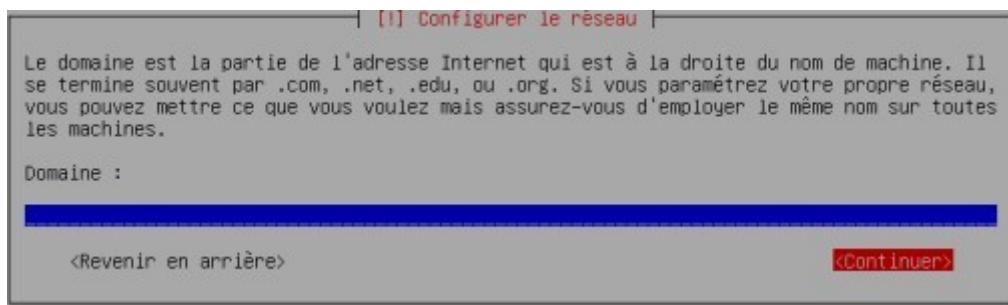
Au premier écran notable nous entrons un nom, ici debian-serveur



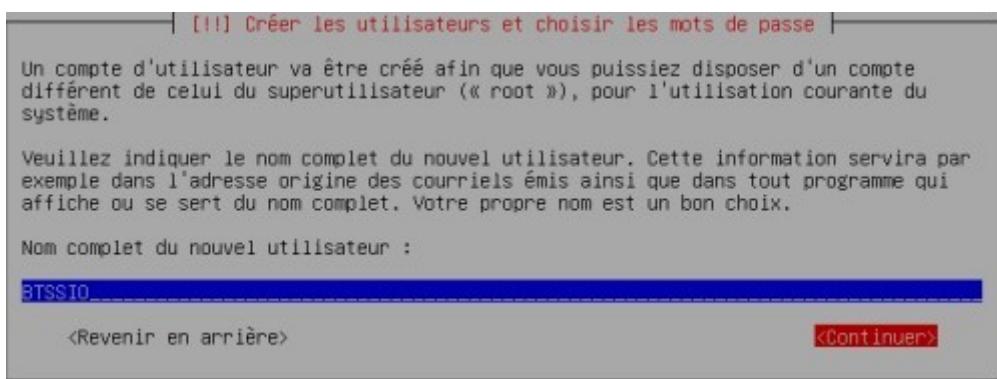
Un mot de passe solide pour l'utilisateur root



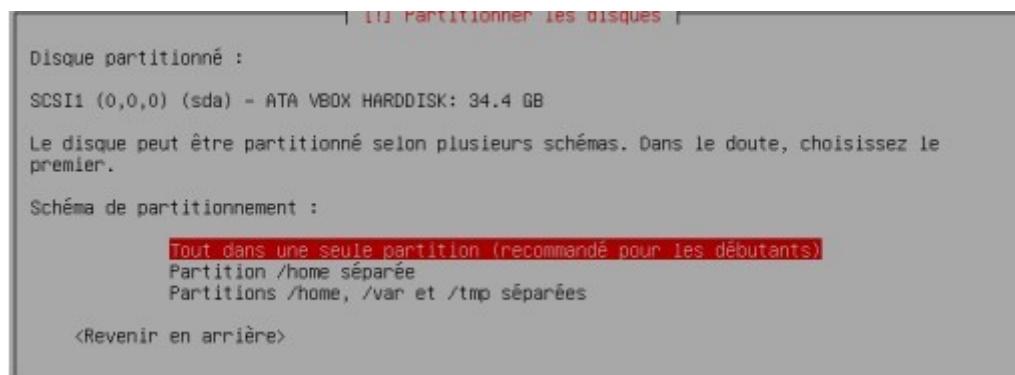
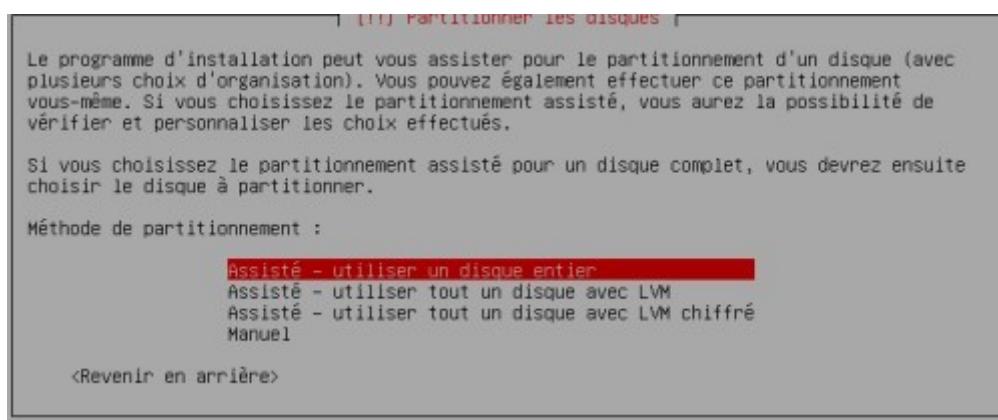
Pour ce qui est du nom de domaine nous laissons par défaut



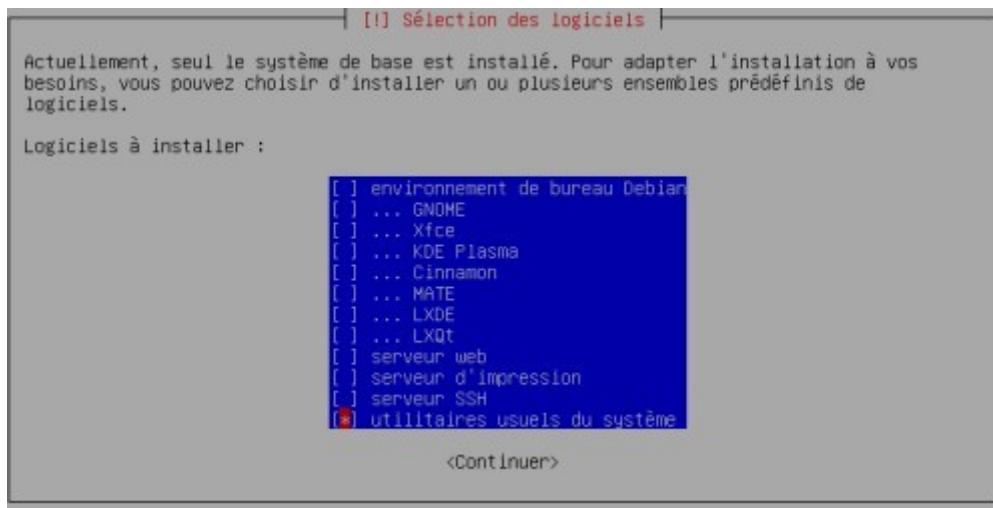
On crée un utilisateur et on définit son mot de passe



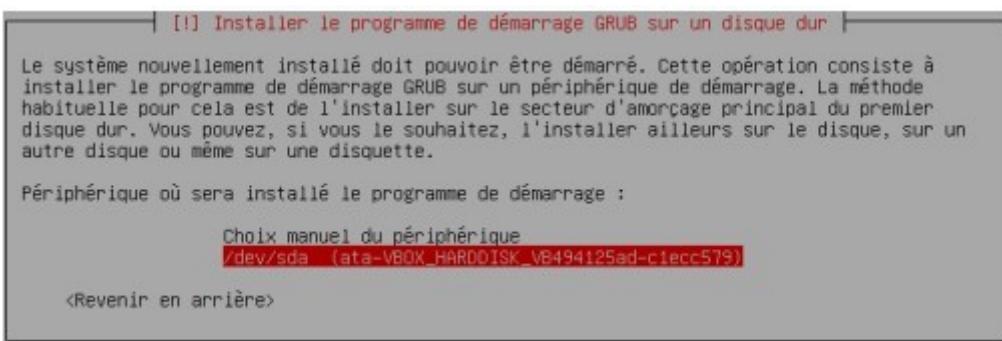
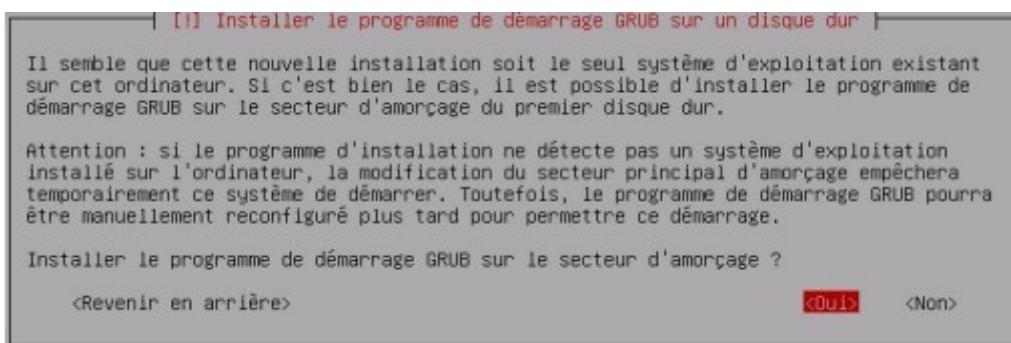
Pour la partie partitionnement on utilise un disque entier et tout dans une seule partition



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'amorçage, soit le seul disque dur présent



Nous pouvons désormais créer nos deux serveurs.

## - **Création du serveur apache PHP phpmyadmin**

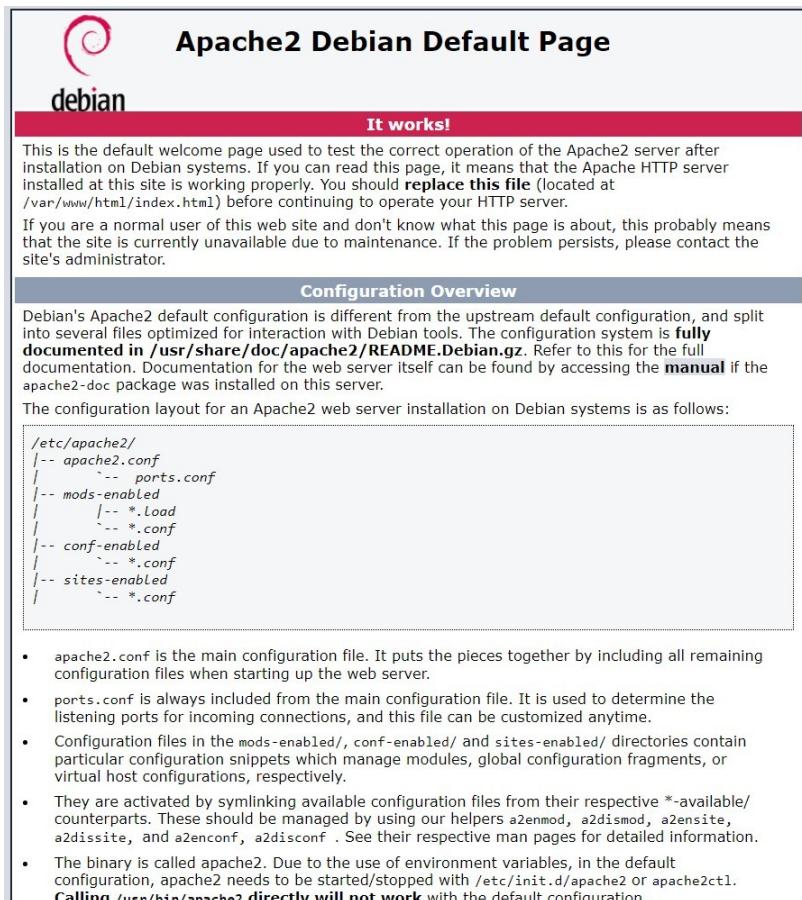
Depuis un clone de notre master Debian, nous commençons par changer son hostname en modifiant le fichier **/etc/hostname**, on le nommera SRV-WEB, on fera de même dans le fichier **/etc/hosts**

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.



The screenshot shows the Apache2 Debian Default Page. At the top, there is a logo for the Debian operating system. Below the logo, the page title is "Apache2 Debian Default Page". A red banner across the top contains the text "It works!". The main content area starts with a paragraph explaining that this is the default welcome page for testing the Apache2 server. It mentions that if the page is readable, the server is working properly and should be replaced with a custom index.html file. Below this, another paragraph states that if the site is currently unavailable, it might be due to maintenance and to contact the administrator. A "Configuration Overview" section follows, which details the layout of the Apache2 configuration files. It shows a hierarchical structure starting from /etc/apache2/ and listing sub-directories like apache2.conf, ports.conf, mods-enabled, conf-enabled, and sites-enabled, each containing various configuration files. A list of bullet points at the bottom explains the purpose of these files, mentioning that apache2.conf is the main configuration file, ports.conf handles listening ports, and the other files manage modules, global configuration fragments, and virtual host configurations. It also notes that configuration files are symlinked from their respective \*\_available counterparts and provides links to the man pages for a2enmod, a2dismod, a2ensite, a2dissite, and a2enconf, a2disconf.

- Les logs sont disponibles 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- <b>amd64</b> #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-apache.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-ctype.ini, /etc/php/7.4/apache2/conf.d/20-xml.ini, /etc/php/7.4/apache2/conf.d/20-fileno.ini, /etc/php/7.4/apache2/conf.d/20-fileno.ini, /etc/php/7.4/apache2/conf.d/20-ftp.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-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.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-json.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-phar.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-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.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-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.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-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.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-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.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	
<b>zendengine</b>	

## 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**

## - Crédation 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 à répondre à 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 bases 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 à 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            [::]:*
gestion@SRV-BDD:~$
```

mariadb utilise le port 3306.

A ce stade le serveur mariaDB est installé et prêt à 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