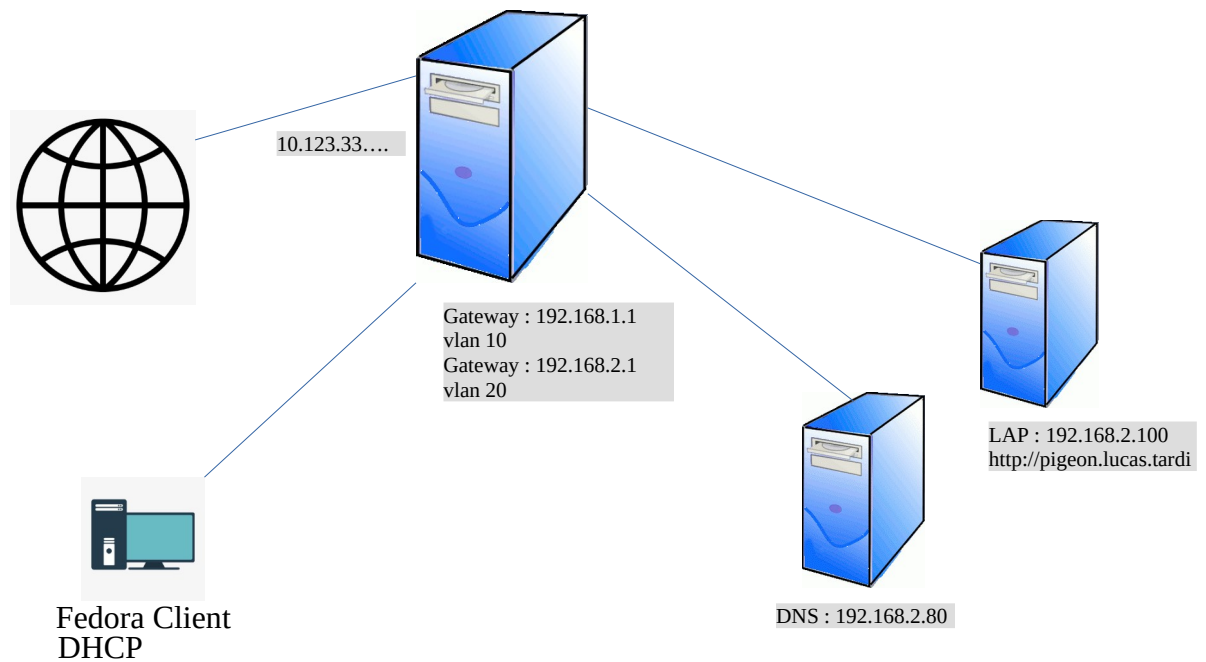


mise en place d'un serveur LAMP

infrastructure réseau

infrastructure réseau utiliser pour mettre en place le serveur LAMP



	Adresse IP machine	Passerelle	Vlan
Fedora client	DHCP (192.168.1.30-50)	192.168.1.1	10
DNS	192.168.2.80	192.168.2.1	20
LAMP	192.168.2.100	192.168.2.1	20
Site principal	http:pigeon.lucas.tardi		
Site secondaire Vhost	https:enclume.lucas.tardi		

Autoriser le DNS sur le VLAN Client

configurer les autorisations via les ACL sur le DNS

Configuration du fichier `/etc/bind/named.conf.options` pour autoriser les hôtes d'un autre réseau de communiquer avec le DNS

```
GNU nano 3.2 /etc/bind/named.conf.options
options{
  directory "/var/cache/bind";

  // If there is a firewall between you and nameservers you want
  // to talk to, you may need to fix the firewall to allow multiple
  // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

  // If your ISP provided one or more IP addresses for stable
  // nameservers, you probably want to use them as forwarders.
  // Uncomment the following block, and insert the addresses replacing
  // the all-0's placeholder.

  forwarders {
    8.8.8.8;
  };

  //=====
  // If BIND logs error messages about the root key being expired,
  // you will need to update your keys.  See https://www.isc.org/bind-keys
  //=====
  dnssec-validation auto;

  listen-on-v6 { any; };

  version "not currently available";

  recursion yes;

  allow-recursion {192.168.1.0/24};
  allow-query {192.168.1.0/24};
};
```

Autoriser le VLAN10 de communiquer avec le DNS

allow-recursion [192.168.1.0/24](#)

allow-query [192.168.1.0/24](#)

[adresses de réseaux du client Fedora](#)

mettre en place sur le DNS l'adresse du nom de domaine du serveur LAMP

```
GNU nano 3.2 /etc/bind/db.2.168.192.in-addr.arpa
$TTL 3600
$ORIGIN 2.168.192.in-addr.arpa.
@      IN SOA  debian.lucas.tardi. root.lucas.tardi. (
        20220504;
        3h;
        1h;
        1w;
        1h);
@      IN NS   debian.lucas.tardi.
80     IN PTR  debian.lucas.tardi.
100    IN PTR  pigeon.lucas.tardi.

GNU nano 3.2 /etc/bind/db.lucas.tardi
$TTL 3600
$ORIGIN lucas.tardi.
@      IN SOA  debian.lucas.tardi. root.lucas.tardi. (
        20220407;
        3h;
        1h;
        1w;
        1h);
@      IN NS   debian.lucas.tardi.
debian      IN A  192.168.2.80
pigeon     IN A  192.168.2.100
```

Pour configurer le nom de domaine du serveur LAMP sur le DNS

Configurer le fichier `/etc/bind/db.lucas.tardi`

```
Pigeon IN A 192.168.2.100
nom de la machine  adresse de la machine
```

Puis configuration le fichier `/etc/bind/db.2.168.192.in-addr.arpa`

```
100 IN PTR pigeon.lucas.tardi
adresse partie Hôte nom de domaine du serveur LAMP
```

Mise en place de serveur Web

Installation de apache2

apache : apache est un logiciel libre qui à pour but de créé un serveur HTTP permanentant de stocker des pages Web est de les mettre en service

mettre a jour le serveur

```
Sudo apt update
Sudo apt upgrade
```

vérifier que le serveur Web possède adresse en Static

```
GNU nano 3.2 /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*
#The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens18

# iface ens18 inet dhcp
iface ens18 inet static
address 192.168.2.100/24
gateway 192.168.2.1
nameserver 162.168.2.80
```

Installation des packets apache2

```
Sudo apt install -y apache2
```

après l'installation d'apache2

il faut activer le service apache2

```
Sudo systemctl enable apache2
```

redémarrer le service apache2

```
Sudo systemctl restart apache2  
ou  
sudo service apache2 reload
```

Test de la vérification de la connexion au serveur web avec le client

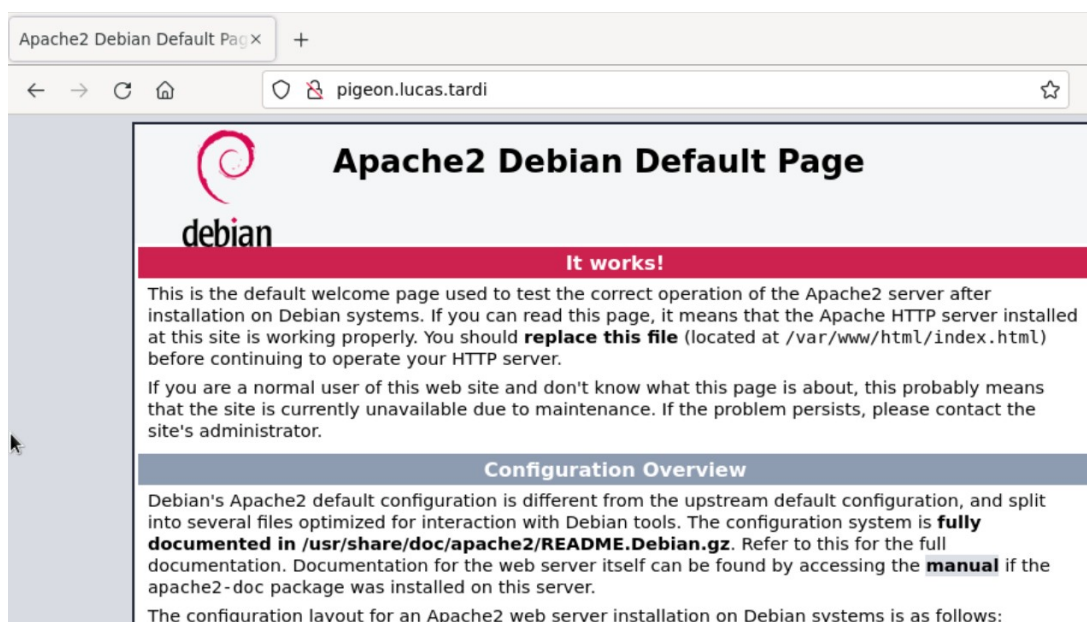
pour voir si le client peut se connecter au serveur Web il doit lancer son moteur de recherche puis marquer dans l'URL l'adresse du serveur

```
192.168.2.100
```

S'il arrive à accéder à cette page apache alors le serveur Web est mis en place.

Puis il faut vérifier si le serveur Web possède un nom de domaine grâce au DNS pour cela il faut marquer le nom de domaine du serveur Web saisi dans le DNS pour arriver sur la même page précédent.

```
http://pigeon.lucas.tardi
```



installation du module PHP

installation des packets PHP

```
Sudo apt install -y php
```

voir la version du module PHP

```
sudo php -v
```

les fichier servi par apache2 sont les dossier de mise en place des fichiers HTML , PHP , CSS ...
qui se trouve dans `/var/www/html`

puis les fichier de configuration qui se trouve dans `/etc/apache/`

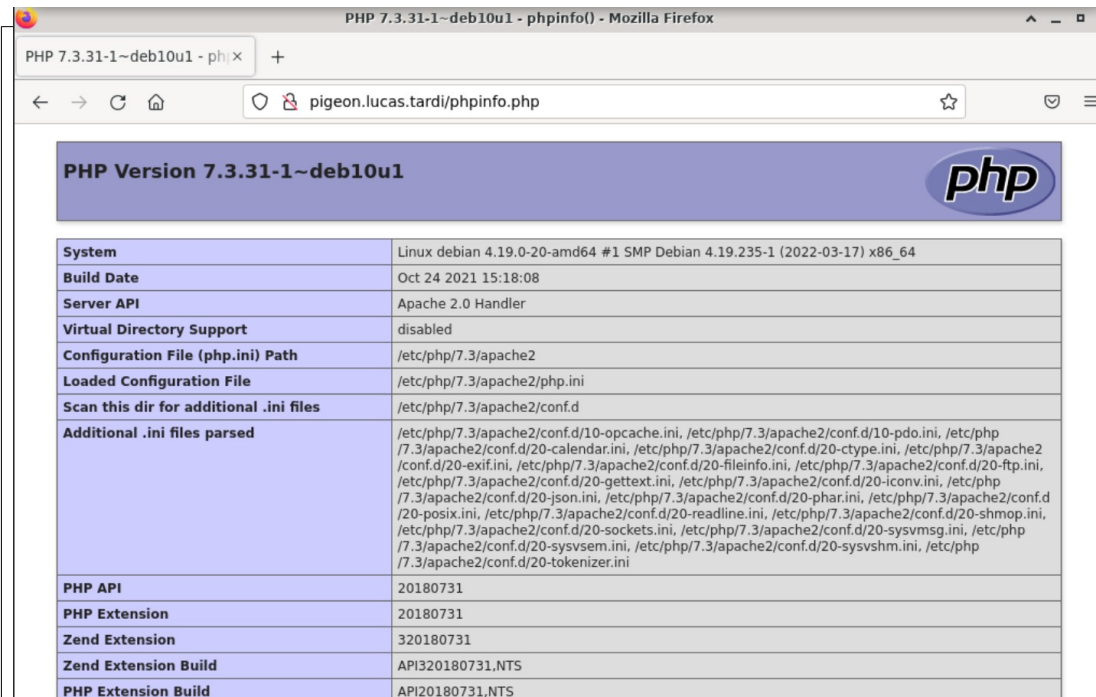
pour voir les caractéristiques de PHP il faut créé un fichier PHP dans l'emplacement `/var/www/html`

```
cd /var/www/html
touch phpinfo.php création d'un fichier php
nano phpinfo.php éditer le fichier php
mettre dans le fichier :
<?php
phpinfo() ;
?>
```

après avoir créé le fichier il faut se rendre sur le client puis marque

<http://pigeon.lucas.tardi/phpinfo.php>


Si une nouvelle page s'affiche alors le module PHP fonction



PHP 7.3.31-1~deb10u1 - phpinfo() - Mozilla Firefox

PHP 7.3.31-1~deb10u1 - phix +

← → ↻ 🏠 🛒 pigeon.lucas.tardi/phpinfo.php ☆ 📄 ☰

PHP Version 7.3.31-1~deb10u1 

System	Linux debian 4.19.0-20-amd64 #1 SMP Debian 4.19.235-1 (2022-03-17) x86_64
Build Date	Oct 24 2021 15:18:08
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/20-calendar.ini, /etc/php/7.3/apache2/conf.d/20-ctype.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /etc/php/7.3/apache2/conf.d/20-fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-gettext.ini, /etc/php/7.3/apache2/conf.d/20-iconv.ini, /etc/php/7.3/apache2/conf.d/20-json.ini, /etc/php/7.3/apache2/conf.d/20-phar.ini, /etc/php/7.3/apache2/conf.d/20-posix.ini, /etc/php/7.3/apache2/conf.d/20-readline.ini, /etc/php/7.3/apache2/conf.d/20-shmop.ini, /etc/php/7.3/apache2/conf.d/20-sockets.ini, /etc/php/7.3/apache2/conf.d/20-sysmsg.ini, /etc/php/7.3/apache2/conf.d/20-sysvsem.ini, /etc/php/7.3/apache2/conf.d/20-sysvshm.ini, /etc/php/7.3/apache2/conf.d/20-tokenizer.ini
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731.NTS
PHP Extension Build	API20180731.NTS

Mise en place de vhosts

définition d'un Vhosts

un Vhosts a pour but d'héberger plusieurs site sur un même serveur web

La configuration des Virtual Hosts se fait dans ce répertoire */etc/apache2/sites-available/*

pour mettre en place le Vhost il lui faut un dossier dans */var/www/* ou il contiendra les fichier HTML PHP...

création d'un dossier

```
Cd /var/www/  
mkdir enclume.tardi nom du dossier du vhost
```

mettre en place un fichier HTML

```
Cd /var/www/enclume.tardi  
touch index.html  
nano index.html  
  
<h1>bonjour</h1>
```

créé un Vhost

pour créé le fichier de configuration du Vhost il faut aller dans le dossier /etc/apache2/sites-available/

pour créé un fichier.conf

```
Cd /etc/apache2/sites-available/  
touch enclume.lucas.tardi.conf nom du site + le nom de domaine du lamp  
nano enclume.lucas.tardi.conf
```

Mettre dans le fichier enclume.lucas.tardi.conf les information saisie dans la capture d'écran

```
GNU nano 3.2 enclume.lucas.tardi.conf  
  
<IfModule mod_ssl.c>  
  <VirtualHost *:80>  
    ServerName enclume.lucas.tardi  
    Redirect / https://enclume.lucas.tardi  
  
    ServerName enclume.lucas.tardi  
    ServerAlias enclume.lucas.tardi  
    ServerAdmin webmaster@localhost  
    DocumentRoot /var/www/enclume.tardi  
  
    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,  
    # error, crit, alert, emerg.  
    # It is also possible to configure the loglevel for particular  
    # modules, e.g.  
    #LogLevel info ssl:warn  
  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined
```

Ensuite aller dans `/etc/apache2/sites-enabled/` et créer un lien symbolique vers le fichier `conf` du virtual-host.

```
ln -s chemin/vers/le/dossier/existant /chemin/vers/le/liensymbolique
```

après activer le site

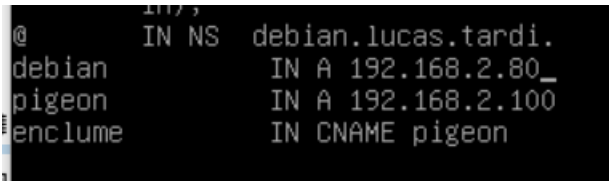
```
a2ensite enclume.lucas.tardi
```

recharger et redémarrer le service apache2

```
service apache2 reload  
systemctl restart apache2
```

DNS

sur le DNS il faut configurer le nom de domaine du Vhost dans `/etc/bind/db.lucas.tardi` puis ajouté le nom du site puis mettre `IN` avec l'option `CNAME` avec le nom de domaine du serveur Lamp : pigeon



```
@ IN NS debian.lucas.tardi.  
debian IN A 192.168.2.80_  
pigeon IN A 192.168.2.100  
enclume IN CNAME pigeon
```

relancer le service DNS

```
Systemctl restart bind9
```

aller sur le client fedora est aller sur le moteur de recherche puis marque le nom de domaine du vhost

nom de domaine du vhost

```
http://enclume.lucas.tardi
```

si vous arriver sur un pages avec le texte saisie sur dans le fichier HTML dans `/var/www/enclume.tardi` alors le vhost est bien configurer

mise en place d'un certificat

générer un certificat

Un certificat a pour but de garantir la confidentialité des données et rassure les internautes en chiffrant la connexion entre le client et le serveur.

Création d'un certificat auto-signé

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

-days 365 indique que le certificat dure 1 ans rsa : 2048 indique le chiffrement de la clé du certificat

ensuite il faut rentrer des informations sur le l'emplacement de la création du certificat

```
root@debian:/var/www/enclume.tardi# cd
root@debian:~# openssl req -x509 -no
-nodes -noout
root@debian:~# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/
certs/apache-selfsigned.crt
Generating a RSA private key
.....+++++
...+++++
writing new private key to '/etc/ssl/private/apache-selfsigned.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:_
```

après avoir saisi les informations du certificat.
deux nouveaux fichiers ont été créé dans :

emplacement des certificat HTTPS

/etc/ssl/certs/apache-selfsigned.crt

emplacement de la key du certificat

/etc/ssl/private/apache-selfsigned.key

pour utiliser le certificat sur le Vhost il faut changer le fichier enclume.lucas.tardi
dans /etc/apache2/sites-available/

```
GNU nano 3.2 enclume.lucas.tardi.conf
<!--Module mod_ssl.c-->
<VirtualHost *:80>
    ServerName enclume.lucas.tardi
    Redirect / https://enclume.lucas.tardi
</VirtualHost>

<VirtualHost _default_:443>
    ServerName enclume.lucas.tardi
    ServerAlias enclume.lucas.tardi
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/enclume.tardi

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf

    #
    # SSL Engine Switch:
    # Enable/Disable SSL for this virtual host.
    SSLEngine on

    #
    # A self-signed (snakeoil) certificate can be created by installing
    # the ssl-cert package. See
    # /usr/share/doc/apache2/README.Debian.gz for more info.
    # If both key and certificate are stored in the same file, only the
    # SSLCertificateFile directive is needed.
    SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

    #
    # Server Certificate Chain:
    # Point SSLCertificateChainFile at a file containing the
    # concatenation of PEM encoded CA certificates which form the
```

Dans le fichier enclume.lucas.tardi
il faut mettre un redirection sur le site en
HTTP
pour l'envoyer sur le HTTPS sur le port 443
puis autoriser le SSL avec « SSLEngine on »

est données l'emplacement du certificat avec

SSLCertificateFile
/etc/ssl/certs/apache-selfsigned.crt

et l'emplacement de la key du certificat

SSLCertificateKeyFile
/etc/ssl/private/apache-selfsigned.key

SSL (secure sockets layer) est un protocole pour navigateurs Web et serveurs qui permet l'authentification, le chiffrement et le déchiffrement des données envoyées sur l'Internet.

un le fichier default.conf dans /etc/apache2/sites-available/ peut être copié pour faciliter la configuration.

Après avoir changé les informations sur le fichier enclume.lucas.tardi.conf

il faut activer le module ssl pour activer le certificat.

activé le le module ssl pour le certificat

```
a2enmod ssl
```

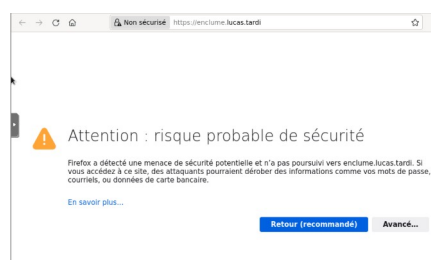
redémarrer le service apache2

```
Sudo systemctl restart apache2  
ou  
sudo service apache2 reload
```

puis aller sur le client fedora puis marque l'adresse du site avec HTTPS

<https://enclume.lucas.tardi>

pour la première fois le site nous avertis que le site possède un certificat qu'il ne connaît pas il faut accepter les risque puis vous arriverait sur le site auparavant



pour voir si la redirection de lien fonctionne on saisie

<http://enclume.lucas.tardi>

le site dois nous redirigé sur le site en HTTPS.

Pour voir les informations du certificat il faut aller (pour Firefox) aller sur le petit cadenas puis afficher le certificat

debian.lucas.tardi	
Nom du sujet	
Pays	FR
État / Province	france
Localité	Aubusson
Organisation	Tardi-industries
Unité organisationnelle	caribou
Nom courant	debian.lucas.tardi
Nom de l'émetteur	
Pays	FR
État / Province	france
Localité	Aubusson
Organisation	Tardi-industries
Unité organisationnelle	caribou
Nom courant	debian.lucas.tardi
Validité	
Pas avant	Thu, 12 May 2022 09:01:13 GMT
Pas après	Fri, 12 May 2023 09:01:13 GMT

Mise en place de SGBD

un SGBD (Système de gestion de base de données) est un logiciel système servant à stocker, à manipuler ou gérer, où à partager des données dans une base de donnée.

installation des packets mariadb

```
sudo apt-get install mariadb-server
```

installation des packets de sécurité mysql

```
mysql_secure_installation
```

Connexion à mariadb

```
Mariadb -u root -p
```

créer un utilisateur

```
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'caribou';  
GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost';  
FLUSH PRIVILEGES
```

le but de créer un utilisateur avec des droits d'administration sans passer par le compte root est de pouvoir le désactiver plus tard le compte root ce qui permet de renforcer la sécurité de mariadb

connexion à distance a mariadb

pour se connecter a mariadb via une connexion il faut installer sur la machine cliente *mariadb-client*

```
Apt install mariadb-client
```

après l'installation pour pouvoir se connecter a distance a mariadb on utilise la commande suivante :

```
Mariadb -u nom utilisateur -p -h adresse du serveur de base de donnée
```

Sources :

<https://forum-francophone-linuxmint.fr/viewtopic.php?t=13032>

<https://www.it-connect.fr/installer-un-serveur-lamp-linux-apache-mariadb-php-sous-debian-11/>

<https://doc.ubuntu-fr.org/lamp>