

# Mission 3

## Installation d'un environnement web sur Proxmox :

- Après avoir créé le conteneur nous devons créer un utilisateur pour ne pas avoir à utiliser le compte root lors de la configuration de la VM.

```
root@CT271:~# adduser gestion
Adding user `gestion' ...
Adding new group `gestion' (1000) ...
Adding new user `gestion' (1000) with group `gestion' ...
Creating home directory `/home/gestion' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
```

## Commande : -adduser

- On installe le service apache2.

Commande : -apt install apache2 -y

- On installe le paquetage PHP.

```
apt -y install apt-transport-https lsb-release ca-certificates curl  
  
root@CT271:/etc/apache2/sites-available# wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg  
  
sudo bash -c 'echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main" > /etc/apt/sources.list.d/php.list'  
  
sudo apt -y install php
```

## Commande : -apt install / -wget / -bash

// pour visualiser la version de php installée on utilise la commande `php -v`

- On installe les paquetages php pour faire fonctionner MariaDB

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

## Commande : -apt install

```
// pour visualiser la version du serveur apache2 on utilise la commande -apache2 -V  
// pour connaître la version du moteur de script php on utilise la commande php -V
```

## Installation de MariaDB :

- On commence par installer les paquets MariaDB

```
sudo apt -y install mariadb-server
```

Commande : -apt install

- On crée un utilisateur admin qui permettra d'administrer MariaDB

```
root@CT271:/home/gestion# sudo mariadb -u root -e "CREATE USER admin@'%'; GRANT ALL PRIVILEGES ON *.* to admin@'%' IDENTIFIED BY 'P@$$word1' WITH GRANT PRIVILEGES;"
```

Commande SQL: CREATE USER, GRANT ALL PRIVILEGES ON, IDENTIFIED BY, WITH GRANT PRIVILEGES

## Connexion à MariaDB :

- On se connecte au service MariaDB

```
sudo mariadb
```

Commande: -sudo

- On change le mot de passe pour root

```
SET PASSWORD FOR 'root'@'localhost' = PASSWORD('P@$$word1');
```

Commande SQL: SET PASSWORD FOR = PASSWORD

## Installation de phpMyAdmin :

- On va rechercher les paquets sur internet avec :

```
wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz
```

Commande: -wget + URL

- On décomprime le dossier zip :

```
tar xvf phpMyAdmin-latest-all-languages.tar.gz
```

Commande: -tar + acronymes xvf pour extraire le dossier dans le dossier courant

- On modifie le fichier de configuration :

```
config.sample.inc.php
```

Commande: -nano

- On change la ligne 14 par ceci :

```
'  
$cfg['TempDir'] = '/var/www/phpmyadmin/tmp';
```

- On se déplace ici :

```
/etc/apache2/conf-available#
```

Commande: -cd

- On ajoute dans le fichier `phpmyadmin.conf` le code suivant :

```
Alias /phpmyadmin /var/www/phpmyadmin  
  
<Directory /var/www/phpmyadmin/>  
    AddDefaultCharset UTF-8  
    <IfModule mod_authz_core.c>  
        <RequireAny>  
            Require all granted  
        </RequireAny>  
    </IfModule>  
</Directory>  
  
<Directory /var/www/phpmyadmin/setup/>  
    <IfModule mod_authz_core.c>  
        <RequireAny>  
            Require all granted  
        </RequireAny>  
    </IfModule>  
</Directory>
```

- On vérifie si le fichier n'a pas d'erreurs :

Commande: -a2enconf (« phpmyadmin.conf »)

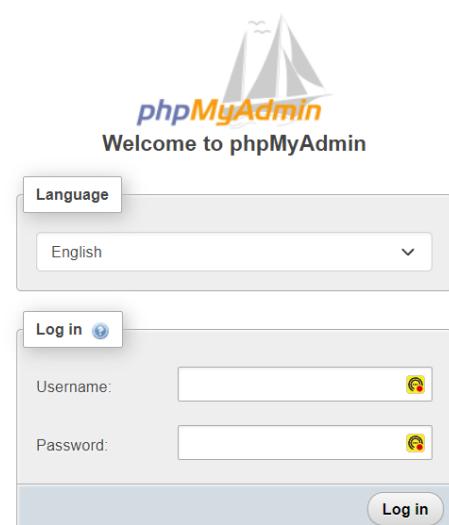
- On redémarre le service apache2

Commande: -systemctl restart (« apache2 »)

- Normalement nous pouvons accéder à phpmyadmin avec l'adresse IP suivit du /phpmyadmin:

10.187.35.132/phpmyadmin/

- Résultat:



### Création de la base de données pour Geststages :

- On importe le dossier de création de base de données de notre machine à la VM

```
scp -r creationbdd gestion@10.187.35.147:/home/gestion
```

Commande : - scp -r (« répertoire ») + nom de fichier + user de la vm @ + @IP + dossier de la vm destination

- On tape la commande de création de la base via ce dossier importé

```
sudo mariadb -u admin -p < creationdbb.sql
```

Commande : sudo mariadb -u utilisateur -p < + fichier SQL

- On voit que la base a bien été créée

```
MariaDB [ (none) ]> show databases;
+-----+
| Database           |
+-----+
| bdd_geststages    |

```

Commande : show databases;

### Ajout de l'application Geststages :

- On crée un nouvel utilisateur et on l'ajoute au groupe dev:

Commande: -adduser (« nathan »)

```
-adduser (« nathan ») dev
```

- On donne les droits d'accès au groupe dev sur le dossier Geststages:

```
:/var/www/html/geststages# sudo chown -R www-data:dev /var/www/html/geststages
```

Commande: -sudo chown -R (« récursive ») + groupe + dossier visé

- On donne le droit en écriture sur le dossier Geststages

```
/var/www/html/geststages# sudo chmod g+w -R /var/www/html/geststages
```

Commande: -chmod + paramètres (« w (write) ») + -R (« récursive ») + dossier visé

- On a bien les modifications qui ont été faites

```
drwxrwxr-x 5 www-data dev 4096 Mar 20 10:25 geststages
```

Commande: ls -l /var/www/html

- On modifie le fichier `/var/www/html/geststages/include/fonction/f_bdd.php`

```
$bdd = new PDO('mysql:host=10.187.35.147;port=3306;dbname=bdd_geststages;charset=utf8');
```

on remplace l'adresse ip de base par celle de notre vm puisque la base de données se trouve sur la même vm

- On crée un compte avec toutes les habilitations sur la base de données bdd\_geststages:

```
CREATE USER 'nomcompte'@'%' IDENTIFIED BY 'P@$$word1';
```

```
GRANT ALL PRIVILEGES ON bdd_geststages.* TO 'nomcompte'@'%';
```

```
flush privileges;
```

Commande SQL: CREATE USER, GRANT ALL PRIVILEGES ON (“nomfichier”) TO (“utilisateurs”), flush privileges.

- On a maintenant accès au site de gestion des stages via l’URL :  
`10.187.35.147/geststages`
- On obtient ceci :

Gestion des stages

Vous n'êtes pas connecté.  
Identifiez-vous pour poursuivre la navigation.

Login :

Mot de passe :

Vous êtes :

Etudiant

Professeur

Connexion

- On se connecte avec les informations suivantes :

Compte élève :

- login : benpas01
- mot de passe : benpas01

Compte enseignant :

- login : shaann01
- mot de passe : shaann01

- Nous sommes maintenant connectés et avons accès au site Geststage



## Installation du WordPress sur Linux :

- On exécute la commande :

```
wget https://wordpress.org/latest.zip
```

Commande : -wget

- On crée la base de données pour le wordpress sur mariadb :

```
MariaDB [ (none) ]> CREATE DATABASE wordpress;
```

- On crée un utilisateur qui servira à configurer la base de données

```
MariaDB [ (none) ]> CREATE USER 'adminwordpress'@'localhost' IDENTIFIED BY 'root';
```

- On lui accorde les droits de modifications :

```
MariaDB [ (none) ]> GRANT ALL PRIVILEGES ON wordpress.* TO adminwordpress@localhost;
Query OK, 0 rows affected (2.911 sec)

MariaDB [ (none) ]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)
```

- On dézippe le fichier téléchargé précédemment :

```
unzip latest.zip -d /var/www/html
```

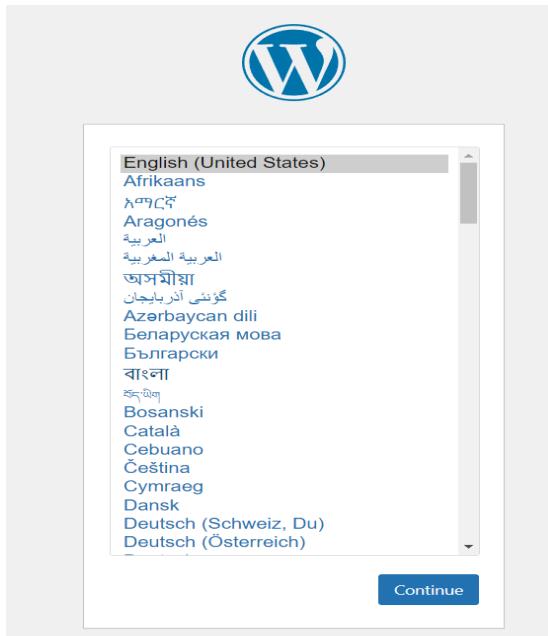
Commande : `-unzip`

- On le place dans le dossier `/var/www/html`

Commande : `-mv`

- On peut maintenant configurer le wordpress à partir de l'adresse ip de notre VM suivie de `/wp-admin/setup-config.php`

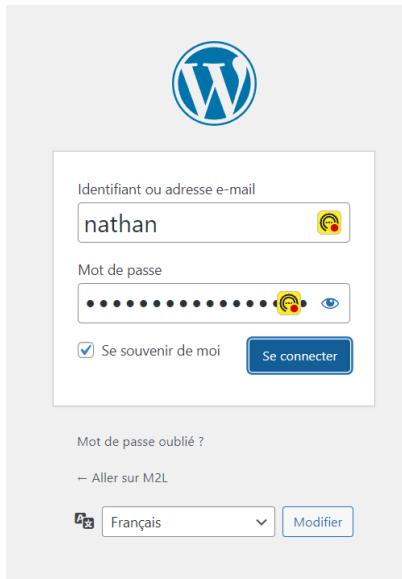
Ce qui donne :



- On suit et exécute les demandes de renseignement (base de données, utilisateur de la base, mdp...)

A screenshot of the WordPress welcome screen. At the top, there's a large blue 'W' logo. Below it, the word 'Bienvenue' is displayed in a large, dark font. A horizontal line follows. Below this, a paragraph of text reads: 'Bienvenue dans la très célèbre installation en 5 minutes de WordPress ! Vous n'avez qu'à remplir les informations demandées ci-dessous et vous serez prêt à utiliser la plus extensible et puissante plateforme de publication de contenu au monde.' Another horizontal line follows. Below this, the section 'Informations nécessaires' is shown with a subtitle: 'Veuillez renseigner les informations suivantes. Ne vous inquiétez pas, vous pourrez les modifier plus tard.' There are two input fields: one for 'Titre du site' and one for 'Identifiant'. Below the 'Identifiant' field, a note states: 'Les identifiants ne peuvent utiliser que des caractères alphanumériques, des espaces, des tirets bas ("\_"), des traits d'union ("-"), des points et le symbole @.'

- Une fois la fin de l'installation faite on se connecte avec les informations renseignées précédemment :



## Résultat :

- Pour pouvoir de nouveaux accéder à ce panneau de configuration wordpress il faudra taper :

**<https://192.168.228.50/wp-admin/>**

- Après avoir configuré notre site en WordPress on y accède avec l'adresse IP de la VM dans l'URL

**<https://192.168.228.50>**

(Pour pouvoir y accéder il faut être sur la Wifi BTSSIO-M2L)

## Installation du serveur DNS :

- On commence par installer le paquet nécessaire au service DNS

```
apt install bind9 bind9-doc bind9utils dnsutils
```

Commande : -apt install

- On peut vérifier le fonctionnement On peut utiliser des outils de diagnostic apporté par ce service bind9 comme :

```
dig @8.8.8.8 google.com
```

Commande : -dig

- On ajoute la zone dans le fichier **named.conf.local**

```
zone "m21.local"{
    type master;
    file "db.m21.local";
};
```

- On crée le fichier db.m21.local

```
db.m21.local
```

Commande : -nano

- On ajoute ceci

```
$TTL 86400
@ IN SOA ns1.m21.local. hostmaster.m21.local. (
    2008113001 ;serial
    86400 ;refresh
    21600 ;retry
    3600000 ;expire
    3600 ;negative caching ttl
)

@ IN NS ns1.m21.local.
ns1 IN A 192.168.228.50

serveurdns IN CNAME ns1
```

- On vérifie le bon fonctionnement

```
named-checkconf -z ou systemctl restart bind9
```

Commande : [-checkconf](#) et [-systemctl](#)

En conclusion nous avons donc mis en place un environnement LAMP pour héberger des sites internet comme une appli de gestion de stage et un site de présentation fait avec wordpress.