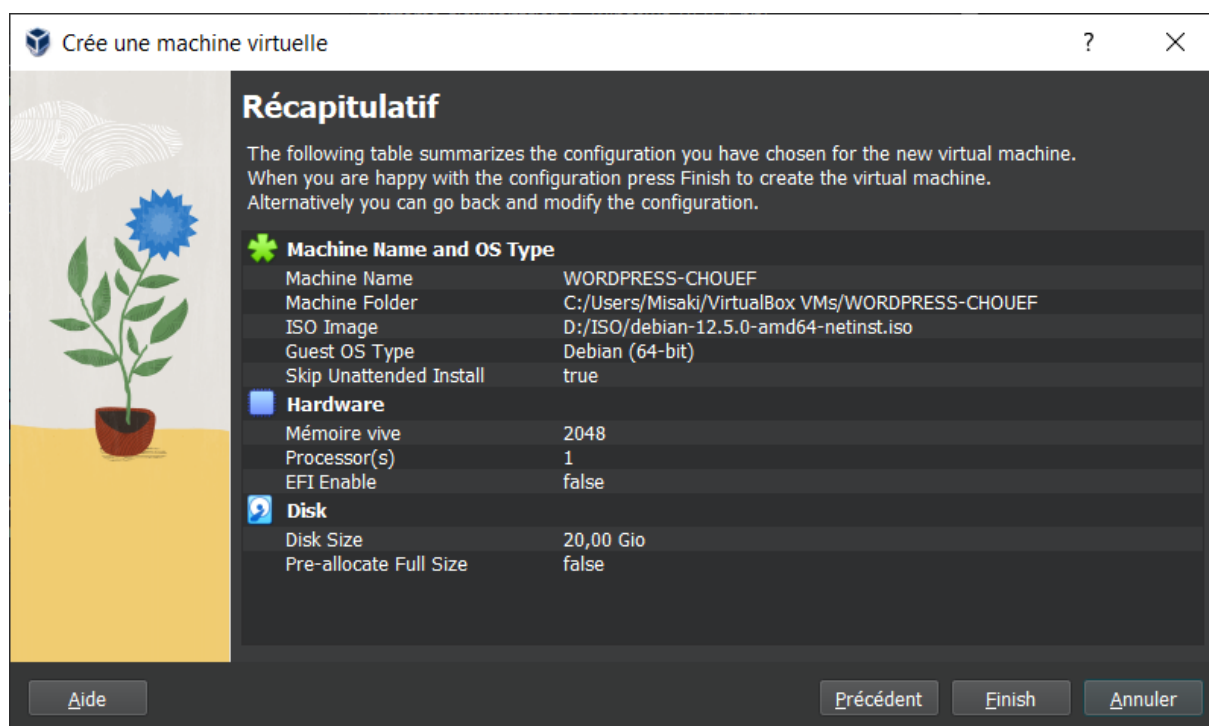


Dans le cadre de ma formation, nous avons mis en place un serveur WordPress sur une VM Debian 12 via VirtualBox. WordPress est un système de gestion de contenu (CMS) open-source utilisé pour créer et gérer des sites web. Ce serveur a été utilisé pour héberger un site web dans un environnement simulé, nous permettant de nous familiariser avec l'installation, la configuration et la gestion de WordPress.

Configuration de la VM :

- **Système d'exploitation** : Debian 12
- **Processeur** : 1 cœur
- **Mémoire vive (RAM)** : 2048 Mo
- **Espace disque** : 20 Go



Étape 1 : Préparation de l'environnement

Nous avons commencé par télécharger et installer Debian 12 sur VirtualBox. La création et la configuration de la VM, ainsi que l'installation du système d'exploitation, ont été réalisées avec succès.

Étape 2 : Mise à jour des paquets

Ensuite, nous avons mis à jour les paquets sur la machine Debian 12 en utilisant les commandes suivantes :

- `apt-get update && apt-get upgrade`

Étape 3 : Installation du socle LAMP

Pour installer Apache2, MariaDB et PHP 8.2 avec les extensions nécessaires, nous avons exécuté les commandes suivantes :

- apt-get install apache2 php mariadb-server
- apt-get install php-xml php-common php-json php-mysql php-mbstring php-curl php-gd php-intl php-zip php-bz2 php-imap php-apcu php-ldap

Étape 4 : Configuration de MariaDB

La sécurisation de MariaDB et la création de la base de données pour WordPress ont été effectuées comme suit :

- mysql_secure_installation
- mysql -u root -p

Ensuite, dans l'interface MySQL :

- CREATE DATABASE db23_wordpress;
- GRANT ALL PRIVILEGES ON db23_wordpress.* TO wp_adm@localhost IDENTIFIED BY "MonMotDePasse";
- FLUSH PRIVILEGES;
- EXIT;

Étape 5 : Téléchargement et installation de WordPress

Le téléchargement et la décompression de l'archive WordPress dans le répertoire web d'Apache ont été réalisés avec les commandes suivantes :

- cd /tmp
- wget https://wordpress.org/latest.tar.gz
- tar -xzf latest.tar.gz -C /var/www/
- chown www-data /var/www/wordpress/ -R

Étape 6 : Configuration des répertoires de WordPress

Nous avons créé des répertoires sécurisés pour la configuration, les fichiers et les logs de WordPress avec les commandes suivantes :

- mkdir /etc/wordpress
- chown www-data /etc/wordpress/
- mv /var/www/wordpress/wp-config.php /etc/wordpress/
- mkdir /var/lib/wordpress
- chown www-data /var/lib/wordpress/
- mv /var/www/wordpress/wp-content /var/lib/wordpress
- mkdir /var/log/wordpress
- chown www-data /var/log/wordpress

Étape 7 : Configuration Apache2

Nous avons configuré le VirtualHost WordPress en créant et modifiant le fichier suivant :

- `nano /etc/apache2/sites-available/wordpress-mohamed.lan.conf`

Contenu :

```
<VirtualHost *:80>
    ServerName wordpress-mohamed.lan
    DocumentRoot /var/www/wordpress
    <Directory /var/www/wordpress>
        Require all granted
        RewriteEngine On
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteRule ^(.*)$ index.php [QSA,L]
    </Directory>
</VirtualHost>
```

Ensuite, nous avons activé le site et les modules nécessaires, puis redémarré Apache2 :

- `a2ensite wordpress-mohamed.lan.conf`
- `a2dissite 000-default.conf`
- `a2enmod rewrite`
- `systemctl restart apache2`

Étape 8 : Utilisation de PHP-FPM avec Apache2

Pour améliorer les performances, nous avons installé et configuré PHP-FPM :

- `apt-get install php8.2-fpm`

- `a2enmod proxy_fcgi setenvif`
- `a2enconf php8.2-fpm`
- `systemctl reload apache2`
- `nano /etc/php/8.2/fpm/php.ini`

Modification de `session.cookie_httponly` pour `on`.

- `systemctl restart php8.2-fpm.service`
- `nano /etc/apache2/sites-available/wordpress-mohamed.lan.conf`

Ajout du handler PHP-FPM :

```
<FilesMatch \.php$>  
    SetHandler "proxy:unix:/run/php/php8.2-fpm.sock|fcgi://localhost/"  
</FilesMatch>
```

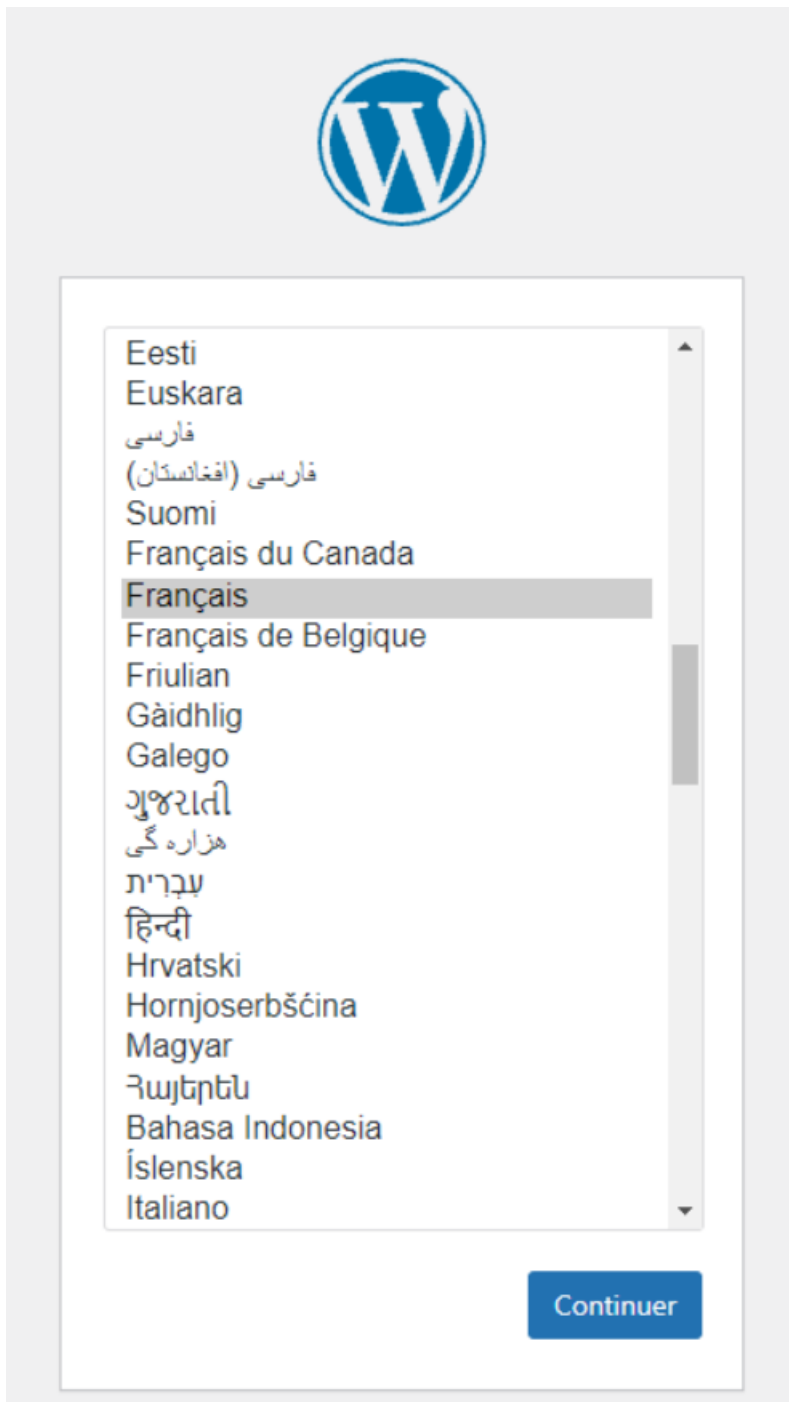
Redémarrage d'Apache2 :

- `systemctl restart apache2`

Étape 9 : Installation de WordPress via l'interface web

L'installation de WordPress via l'interface web a été réalisée en accédant à l'URL du serveur et en suivant les étapes de configuration :

- Choix de la langue
- Installation
- Vérification des prérequis
- Configuration de la base de données
- Finalisation de l'installation



Étape 10 : Sécurisation et finalisation

Pour sécuriser l'installation, nous avons changé les mots de passe par défaut et configuré les paramètres de sécurité recommandés dans l'interface de gestion de WordPress. Nous avons également supprimé le fichier d'installation :

- `rm /var/www/wordpress/wp-admin/install.php`

Conclusion :

En conclusion, cette mission m'a permis de développer et de valider mes compétences en administration de systèmes et réseaux, conformément aux exigences du module [Mettre à disposition des utilisateurs un service informatique]. J'ai démontré ma capacité à installer, configurer et sécuriser un serveur web avec un CMS comme WordPress. Ce projet m'a également sensibilisé aux enjeux de la gestion de contenu web et de la sécurité des sites internet en entreprise.