Mr OBERLAENDER

14/03/2025

Compte rendu TP

Installation d'une architecture 3-tier Wordpress / Mariadb / Reverse Proxy sous Hyper V

TEWES Arnaud
BTS SIO SISR 2ÈME ANNÉE

Introduction

Ce document décrit le processus de déploiement d'une infrastructure web pour l'hébergement d'un site WordPress, en utilisant la distribution Alpine Linux pour sa légèreté et son efficacité. L'architecture mise en œuvre repose sur une séparation des rôles serveur afin d'optimiser la performance et la maintenabilité.

L'infrastructure comprend les composants suivants :

- **Serveur Apache2 :** Dédié à l'exécution de l'application WordPress.
- **Serveur MariaDB :** Hébergeant la base de données WordPress.
- **Serveur Nginx (Reverse Proxy) :** Configuré pour la gestion du trafic entrant depuis la DMZ et le routage vers le serveur Apache2 au sein du VLAN serveur.

L'utilisation d'Alpine Linux pour chaque serveur contribue à une empreinte système minimale et à une gestion simplifiée. Le serveur Nginx, agissant comme reverse proxy en point d'entrée depuis la DMZ, facilite l'accès au site WordPress hébergé sur le réseau interne.

Ce document détaillera les étapes de configuration de chaque serveur, en se concentrant sur la mise en place d'un environnement WordPress fonctionnel. L'objectif est de fournir une compréhension claire de l'architecture déployée et des considérations essentielles pour son administration.

Configuration d'Apache2, PHP et installation de WordPress

Mise à jour et installation des paquets nécessaires

```
apk update
apk upgrade
apk add apache2
apk add php8.2 php8.2-apache2
apk add nano
apk add sudo
```

Configuration des dépôts et installation supplémentaire

```
Nano /etc/apk/repositories
apk add sudo
apk add php82 php82-apache2
apk add php82
rc-service apache2 restart
apk add php82-apache2
apk add apache2-libs
nano /etc/apk/repositories
apk add libapache2-mod-php82
apk add libapache2-mod-apparmor
apk add apache2-libs
rc-service apache2 restart
```

Configuration du répertoire web

```
cd /var/www/localhost/htdocs/
rm index.html
apk add mariadb-client
```

Test de connexion à la base de données

Utilisateur `test` avec le mot de passe `test` créer sur la base de données mariaDB

```
mariadb -h 192.168.10.10 -p -u test
```

Installation de WordPress

```
cd /tmp
apk add wget
wget https://wordpress.org/latest.zip
apk add zip
unzip latest.zip -d /var/www/localhost/htdocs/
cd /var/www/localhost/htdocs/
mv wordpress/* /var/www/localhost//htdocs
rm wordpress/ -Rf
chown -R apache:apache /var/www/localhost/htdocs/
apk add php82-mysqli
rc-service apache2 restart
rc-update add apache2
```

Installation et configuration de MariaDB

1. Mise à jour et préparation initiale

```
apk update
apk upgrade
apk add nano
nano /etc/apk/repositories
apk update
apk upgrade
```

2. Installation de MariaDB

```
apk add sudo
apk add mariadb-server
apk add mariadb-client
apk add mariadb
/etc/init.d/mariadb setup
rc-service mariadb start
```

3. Initialisation de MariaDB

```
mysql_install_db --user=mysql --datadir=/var/lib/mysql
rc-service mariadb status
rc-update add mariadb
```

4. Connexion à MariaDB

```
mariadb
sudo mysql -u root -p
```

5. Création de la base de données et de l'utilisateur

```
CREATE DATABASE WPDB;
CREATE USER 'root'@'%' IDENTIFIED BY 'toor';
GRANT ALL PRIVILEGES ON WPDB.* TO 'root'@'%';
FLUSH PRIVILEGES;
```

6. Configuration du fichier MariaDB

Chemin : `/etc/my.cnf.d/mariadb-server.cnf`

```
skip-networking #Commentez cette ligne dans le fichier de conf
bind-address = 0.0.0.0 #Mettre 0.0.0.0 pour autoriser les connexions de
toutes les IP sur la base de données
```

7. Redémarrage de MariaDB

```
rc-service mariadb restart
```

Installation et configuration du ReverseProxy avec Nginx

1. Installation de Nginx et Nano

```
apk add nginx
apk add nano
```

2. Modification du fichier de conf de Nginx pour le http

```
nano /etc/nginx/nginx.conf

-----------------------------------------------------

# /etc/nginx/nginx.conf

user nginx;

events {
worker_connections 1024;
}

http {
    server {
        listen 80;
        server_name test.miami.local;

        location / {
            proxy_pass http://192.168.10.15;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;
        }
    }
}
-----------------------------------------------------
```

3. Création du repertoire SSL + génération des certificats

```
mkdir ssl
cd ssl/
```

Génération de la clé privée et du certificat CA

```
openssl genrsa -des3 -out elao-ca.key 2048 -rand /var/log/messages
openssl req -new -x509 -days 3650 -key elao-ca.key -out elao-ca.crt
```

Vérification du certificat CA :

```
openssl x509 -in elao-ca.crt -text -noout
```

Génération de la clé privée et CSR (Certificate Signing Request) pour le serveur :

```
openssl genrsa -des3 -out elao-server.key 1024
openssl req -new -key elao-server.key -out elao-server.csr
```

Signature et création du certificat serveur par le CA :

```
openssl x509 -req -in elao-server.csr -out elao-server.crt -sha1 -CA elao-
ca.crt -CAkey elao-ca.key -CAcreateserial -days 3650
openssl x509 -in elao-server.crt -text -noout
```

Création d'une clé serveur sans mot de passe (clé "insecure") :

```
openssl rsa -in elao-server.key -out elao-server.key.insecure
```

Génération du certificat serveur avec SHA-256 :

```
openssl req -new -x509 -sha256 -days 365 -key elao-server.key -out elao-
server.crt
```

Modification et redémarrage de Nginx :

```
rc-service nginx restart #erreur
```

Génération d'une nouvelle clé privée plus longue (2048 bits) pour le serveur :

```
openssl genrsa -out /etc/nginx/ssl/new-elao-server.key 2048

openssl req -new -key /etc/nginx/ssl/new-elao-server.key -out
/etc/nginx/ssl/new-elao-server.csr

openssl x509 -req -days 365 -in /etc/nginx/ssl/new-elao-server.csr -signkey
/etc/nginx/ssl/new-elao-server.key -out /etc/nginx/ssl/new-elao-server.crt
```

4. Mise à jour de la configuration de Nginx avec les bons certificats sécurisé (https)

```
nano nginx.conf
```

```
# /etc/nginx/nginx.conf

user nginx;

events {
worker_connections 1024;

}

http {
    server {
        listen 80;
        server_name test.miami.local;

        # Redirection HTTP vers HTTPS
        location / {
            return 301 https://$host$request_uri;
        }
    }

    server {
        listen 443 ssl;
        server_name test.miami.local;

        # Chemins des certificats SSL
        ssl_certificate /etc/nginx/ssl/new-elao-server.crt;
        ssl_certificate_key /etc/nginx/ssl/new-elao-server.key;

        # Configuration SSL supplémentaire (meilleures pratiques)
        ssl_protocols TLSv1.2 TLSv1.3;
        ssl_prefer_server_ciphers on;
        ssl_ciphers HIGH:!aNULL:!MD5;

        location / {
            proxy_pass http://192.168.10.15;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;
        }
    }
}
```

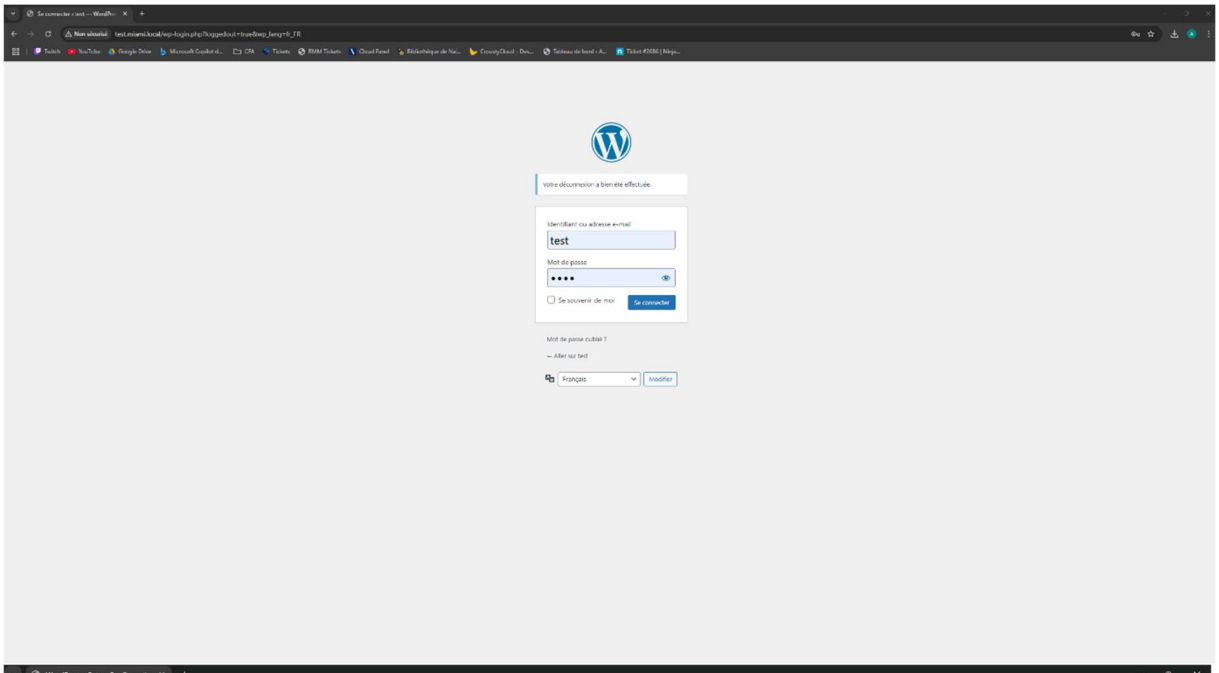
---

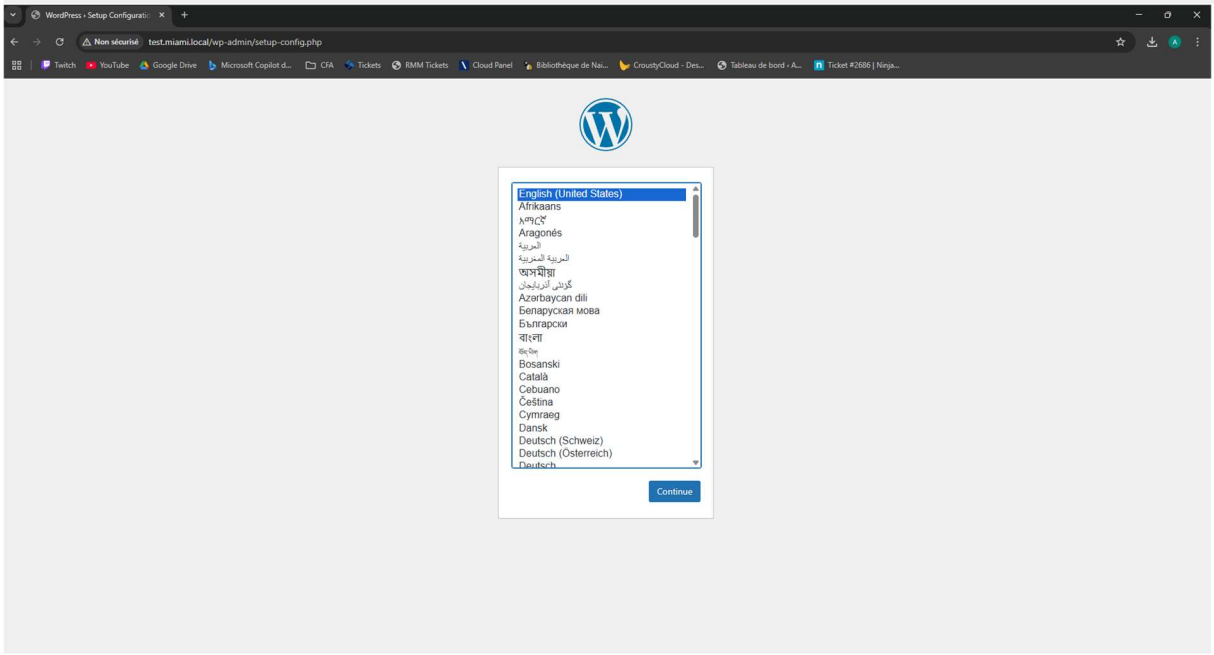
5. Redémarrage de Nginx

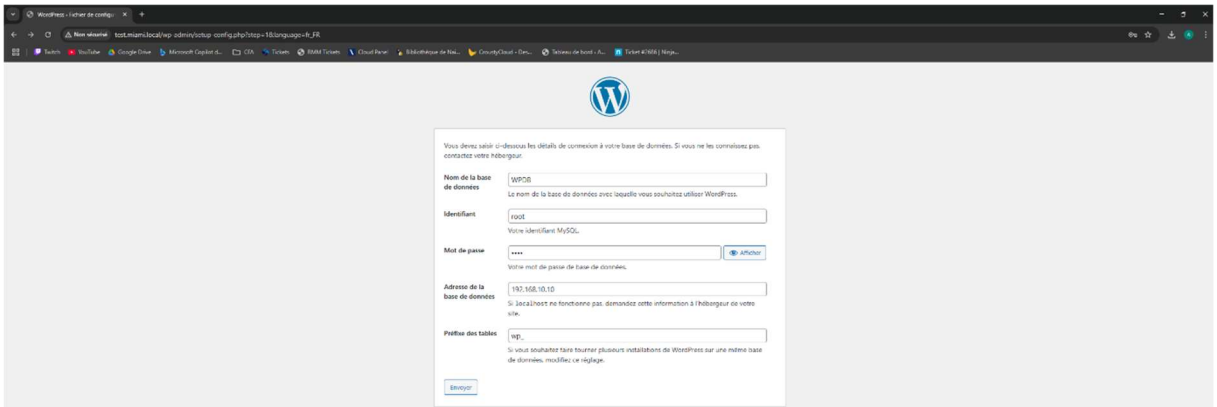
```
rc-service nginx restart
```

Test de connexion au site internet depuis la redirection DNS sur l'ip du reverse proxy créer sur notre machine de test

Installation de Word Press :







test

Page d'exemple

Blog

Bonjour tout le monde !

Bienvenue sur WordPress. Ceci est votre premier article. Modifiez-le ou supprimez-le, puis commencez à écrire !

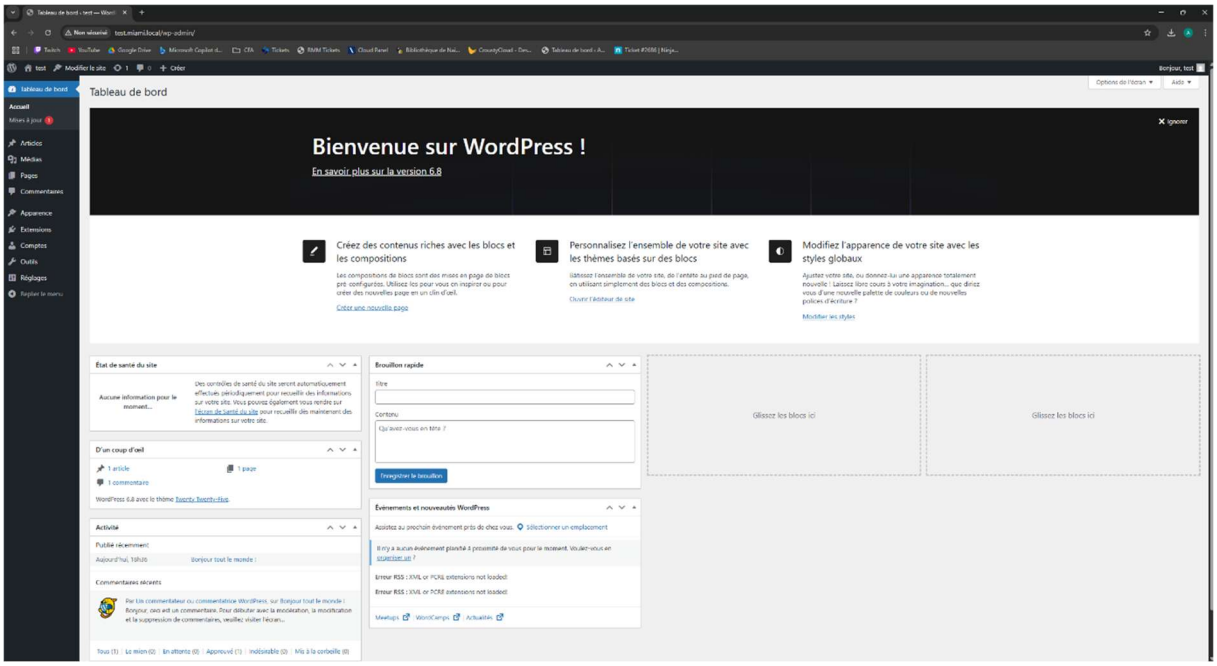
18 avril 2025

test

Blog
About
FAQs
Authors
Events
Shop
Patterns
Themes

Twenty Twenty-Five

Designed with WordPress



Check du certificat SSL auto signé