

# Déploiement d'un serveur web Nginx avec WordPress

Configuration de virtual hosts multiples pour l'hébergement de sites web

**Auteur :** Ahmed Koucha

**Organisation :** GSB

**Date :** Mai 2025

## Table des matières

1. Introduction
2. Architecture de la solution
3. Préparation de l'environnement
4. Installation et configuration de Nginx
5. Configuration du site principal (gsb.org)
6. Configuration du site secondaire (afrique.gsb.org)
7. Configuration SSL/TLS
8. Tests et validation
9. Maintenance et surveillance
10. Conclusion
11. Annexes

## 1. Introduction

### 1.1 Contexte du projet

Dans le cadre de la modernisation de l'infrastructure web de GSB, ce projet vise à mettre en place un serveur web public basé sur Nginx. L'objectif est d'héberger deux instances WordPress distinctes : le site principal de l'entreprise (gsb.org) et un site dédié à la zone Afrique (afrique.gsb.org). Cette infrastructure doit garantir la sécurité des communications via SSL/TLS, optimiser les performances et faciliter la maintenance des sites web.

## 1.2 Objectifs

- Déployer un serveur web Nginx dans un conteneur LXC
- Configurer deux virtual hosts distincts pour les domaines gsb.org et afrique.gsb.org
- Installer et configurer WordPress pour chaque site
- Mettre en place le chiffrement SSL/TLS pour sécuriser les communications
- Optimiser les performances et la sécurité du serveur web

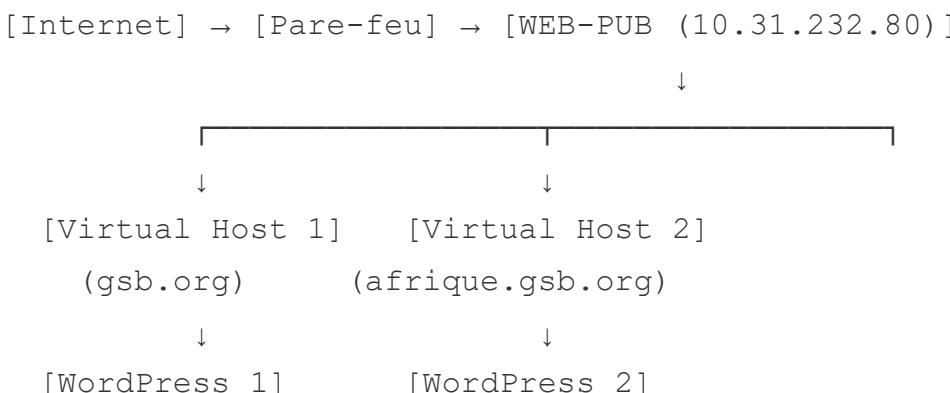
## 2. Architecture de la solution

---

### 2.1 Vue d'ensemble

La solution repose sur un conteneur LXC dédié au web public, hébergé sur l'infrastructure Proxmox existante. Ce conteneur exécute un serveur Nginx qui gère deux virtual hosts distincts, chacun hébergeant une instance WordPress. Cette approche permet d'isoler les configurations et les contenus tout en mutualisant les ressources serveur.

### 2.2 Schéma d'architecture



## 3. Préparation de l'environnement

---

### 3.1 Création du conteneur LXC

Le conteneur LXC est créé à partir d'un template Debian et configuré avec les paramètres réseau appropriés :

```
# Clonage du conteneur template
# Configuration réseau du conteneur WEB-PUB
IP : 10.31.232.80
bridge : Vmbr1
gateway : 10.31.235.254
```

### 3.2 Mise à jour du système

```
# Mise à jour des paquets
apt update
```

### 3.3 Installation des paquets nécessaires

```
# Installation de Nginx
apt install nginx
```

### 3.4 Vérification du service

```
# Vérification que le port 80 est en écoute
sudo netstat -tuln | grep :80
```

## 4. Installation et configuration de Nginx

---

### 4.1 Structure des fichiers de configuration Nginx

Nginx utilise une structure de configuration modulaire :

- /etc/nginx/nginx.conf : Fichier de configuration principal
- /etc/nginx/sites-available/ : Répertoire contenant les configurations des virtual hosts disponibles
- /etc/nginx/sites-enabled/ : Répertoire contenant les liens symboliques vers les configurations activées

- /etc/nginx/snippets/ : Répertoire contenant des fragments de configuration réutilisables

## 4.2 Configuration de base de Nginx

La configuration par défaut de Nginx est généralement suffisante pour commencer. Nous allons nous concentrer sur la création des virtual hosts pour nos sites WordPress.

# 5. Configuration du site principal (gsb.org)

---

## 5.1 Création du virtual host

```
# Création du fichier de configuration pour gsb.org
nano /etc/nginx/sites-available/wordpress_gsb
```

Contenu du fichier de configuration :

```
server {
    listen 80;
    server_name www.gsb.org gsb.org;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name www.gsb.org gsb.org;

    ssl_certificate /etc/ssl/mycert/server.crt;
    ssl_certificate_key /etc/ssl/mycert/server.key;

    root /var/www/wordpress_gsb_org;
    index index.php index.html index.htm;

    # Fusion des deux blocs location /
    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    # Configuration pour les fichiers PHP
    location ~ \.php$ {
        try_files $uri =404;
        include fastcgi_params;
    }
}
```

```
fastcgi_index index.php;
fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
}

location ~ /\.ht {
    deny all;
}

error_log /var/log/nginx/wordpress_gsb_error.log;
access_log /var/log/nginx/wordpress_gsb_access.log;
}
```

## 5.2 Activation du virtual host

```
# Création d'un lien symbolique pour activer le site
ln -s /etc/nginx/sites-available/wordpress_gsb /etc/nginx/sites-enabled/wordp
```

## 5.3 Installation de WordPress

```
# Création du répertoire pour WordPress
mkdir /var/www/wordpress_gsb_org

# Téléchargement de WordPress
cd /var/www/wordpress_gsb_org
wget https://wordpress.org/latest.tar.gz

# Extraction de l'archive
tar -xvzf latest.tar.gz

# Déplacement des fichiers (si nécessaire)
# Si WordPress a été extrait dans un sous-répertoire, déplacez les fichiers
mv wordpress/* .
rmdir wordpress

# Suppression de l'archive
rm latest.tar.gz
```

## 5.4 Configuration des permissions

```
# Attribution des droits appropriés
chown -R www-data:www-data /var/www/wordpress_gsb_org
chmod -R 755 /var/www/wordpress_gsb_org
```

## 6. Configuration du site secondaire (afrique.gsb.org)

### 6.1 Création du virtual host

```
# Création du fichier de configuration pour afrique.gsb.org
nano /etc/nginx/sites-available/wordpress_zone
```

Contenu du fichier de configuration :

```
server {
    listen 80;
    server_name www.afrique.gsb.org afrique.gsb.org;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name www.afrique.gsb.org afrique.gsb.org;

    ssl_certificate /etc/ssl/mycert/server.crt;
    ssl_certificate_key /etc/ssl/mycert/server.key;

    root /var/www/wordpress_zone_gsb_org;
    index index.php index.html index.htm;

    # Fusion des deux blocs location /
    location / {
        try_files $uri $uri/ /index.php$args;
    }

    # Configuration pour les fichiers PHP
    location ~ \.php$ {
        try_files $uri =404;
        include fastcgi_params;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }
}
```

```
location ~ /\.ht {  
    deny all;  
}  
  
error_log /var/log/nginx/wordpress_zone_error.log;  
access_log /var/log/nginx/wordpress_zone_access.log;  
}
```

## 6.2 Activation du virtual host

```
# Création d'un lien symbolique pour activer le site  
ln -s /etc/nginx/sites-available/wordpress_zone /etc/nginx/sites-enabled/word
```



## 6.3 Installation de WordPress

```
# Création du répertoire pour WordPress  
mkdir /var/www/wordpress_zone_gsb_org  
  
# Téléchargement de WordPress  
cd /var/www/wordpress_zone_gsb_org  
wget https://wordpress.org/latest.tar.gz  
  
# Extraction de l'archive  
tar -xvzf latest.tar.gz  
  
# Déplacement des fichiers (si nécessaire)  
# Si WordPress a été extrait dans un sous-répertoire, déplacez les fichiers  
mv wordpress/* .  
rmdir wordpress  
  
# Suppression de l'archive  
rm latest.tar.gz
```



## 6.4 Configuration des permissions

```
# Attribution des droits appropriés  
chown -R www-data:www-data /var/www/wordpress_zone_gsb_org  
chmod -R 755 /var/www/wordpress_zone_gsb_org
```

## 7. Configuration SSL/TLS

### 7.1 Création du répertoire pour les certificats

```
# Création du répertoire pour les certificats SSL
mkdir -p /etc/ssl/mycert
```

### 7.2 Génération ou installation des certificats

Pour un environnement de production, il est recommandé d'utiliser des certificats émis par une autorité de certification reconnue comme Let's Encrypt. Pour un environnement de test, vous pouvez générer des certificats auto-signés :

```
# Génération d'un certificat auto-signé (pour les tests uniquement)
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
-keyout /etc/ssl/mycert/server.key \
-out /etc/ssl/mycert/server.crt \
-subj "/C=FR/ST=Nouvelle-Aquitaine/L=Limoges/O=GSB/CN=gsb.org"
```

### 7.3 Configuration des paramètres SSL dans Nginx

Pour renforcer la sécurité SSL, vous pouvez créer un fichier de configuration SSL réutilisable :

```
# Création d'un snippet SSL
nano /etc/nginx/snippets/ssl-params.conf
```

Contenu du fichier ssl-params.conf :

```
# Paramètres SSL optimisés
ssl_protocols TLSv1.2 TLSv1.3;
ssl_prefer_server_ciphers on;
ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-I
ssl_session_timeout 1d;
ssl_session_cache shared:SSL:10m;
ssl_session_tickets off;

# HSTS (15768000 secondes = 6 mois)
add_header Strict-Transport-Security "max-age=15768000; includeSubDomains" a:
# OCSP Stapling
```

```
ssl_stapling on;
ssl_stapling_verify on;
```

## 7.4 Inclusion des paramètres SSL dans les virtual hosts

Modifiez les fichiers de configuration des virtual hosts pour inclure les paramètres SSL :

```
# Modification des fichiers de configuration
nano /etc/nginx/sites-available/wordpress_gsb
nano /etc/nginx/sites-available/wordpress_zone
```

Ajoutez la ligne suivante dans le bloc server qui écoute sur le port 443 :

```
include snippets/ssl-params.conf;
```

## 8. Tests et validation

---

### 8.1 Vérification de la configuration Nginx

```
# Vérification de la syntaxe de la configuration
nginx -t
```

### 8.2 Redémarrage de Nginx

```
# Redémarrage du service Nginx
systemctl restart nginx
```

### 8.3 Vérification des sites WordPress

Accédez aux sites via un navigateur pour vérifier qu'ils sont accessibles :

- <https://gsb.org>
- <https://afrique.gsb.org>

### 8.4 Configuration initiale de WordPress

Pour chaque site, suivez l'assistant d'installation de WordPress :

1. Sélectionnez la langue
2. Créez un utilisateur administrateur
3. Configurez le titre du site et les informations de base
4. Connectez-vous au tableau de bord WordPress

## 9. Maintenance et surveillance

### 9.1 Surveillance des logs

```
# Surveillance des logs d'accès
tail -f /var/log/nginx/wordpress_gsb_access.log
tail -f /var/log/nginx/wordpress_zone_access.log

# Surveillance des logs d'erreurs
tail -f /var/log/nginx/wordpress_gsb_error.log
tail -f /var/log/nginx/wordpress_zone_error.log
```

### 9.2 Mise à jour de WordPress

Il est important de maintenir WordPress à jour pour des raisons de sécurité. Vous pouvez effectuer les mises à jour via l'interface d'administration de WordPress ou en ligne de commande avec WP-CLI.

### 9.3 Sauvegarde des sites

```
# Sauvegarde des fichiers WordPress
tar -czvf /backup/wordpress_gsb_files_$(date +%Y%m%d).tar.gz /var/www/wordpress_gsb
tar -czvf /backup/wordpress_zone_files_$(date +%Y%m%d).tar.gz /var/www/wordpress_zone

# Sauvegarde de la configuration Nginx
tar -czvf /backup/nginx_config_$(date +%Y%m%d).tar.gz /etc/nginx/sites-available
```

### 9.4 Optimisation des performances

Pour améliorer les performances des sites WordPress, vous pouvez :

- Installer un plugin de cache comme W3 Total Cache ou WP Super Cache
- Configurer la mise en cache au niveau de Nginx

- Optimiser les images et les ressources statiques
- Utiliser un CDN pour distribuer le contenu

## 10. Conclusion

---

Cette solution d'hébergement web basée sur Nginx et WordPress offre une plateforme robuste et sécurisée pour les sites web de GSB. L'utilisation de virtual hosts distincts permet d'isoler les différents sites tout en mutualisant les ressources serveur. La mise en place du chiffrement SSL/TLS garantit la confidentialité et l'intégrité des communications.

L'architecture mise en place est évolutive et pourra facilement être adaptée pour héberger d'autres sites web à l'avenir. La séparation claire des configurations et des contenus facilite la maintenance et la gestion des sites.

## 11. Annexes

---

### 11.1 Références

- Documentation officielle de Nginx : <https://nginx.org/en/docs/>
- Documentation officielle de WordPress : <https://wordpress.org/support/>
- Guide d'optimisation de WordPress avec Nginx : <https://www.nginx.com/blog/nginx-wordpress-performance/>

### 11.2 Glossaire

- **Nginx** : Serveur web haute performance, également utilisé comme proxy inverse et équilibrer de charge
- **WordPress** : Système de gestion de contenu (CMS) open source
- **Virtual Host** : Configuration permettant d'héberger plusieurs sites web sur un même serveur
- **SSL/TLS** : Protocoles cryptographiques assurant la sécurité des communications sur Internet
- **LXC** : Linux Containers, technologie de virtualisation au niveau du système d'exploitation