

Wordpress

- [Installer WordPress sur Debian 11](#)
- [WordPress Désactiver wp_xmlrpc](#)
- [WordPress référencement naturel \(SEO\)](#)
- [WordPress Ajouter un Sitemap](#)
- [WordPress et Google Analytics](#)
- [WordPress Thème enfant](#)

Installer WordPress sur Debian 11

Dans cette procédure, je vais vous expliquer pas à pas comment installer WordPress sur une machine Debian 11. Après avoir suivi cette procédure vous aurez une machine Debian avec WordPress d'installer. En cas de soucis, je vous conseil de consulter le site officiel :

<https://wordpress.com/>



WORDPRESS

Prérequis :

- Une machine Debian 11
- [Un serveur Lamp](#)
- Un accès root sur la machine

Installation de WordPress sur Debian 11

WordPress est un CMS qui utilise une base de données pour stocker tous les articles, utilisateurs et autres... Pour notre utilisation nous allons installer Mariadb, qui est très similaire avec Mysql.

MariaDB :

Connexion sur MariaDb :

```
mysql -u root -p
```

Création d'une nouvelle base de données :

```
CREATE DATABASE wordpress_db;
```

Création d'un nouvel utilisateur MariaDB et attribution des droits sur la nouvelle base de données :

```
GRANT ALL ON wordpress_db.* TO 'wordpress_user'@'localhost' IDENTIFIED BY 'password';
```

Appliquer les privilèges :

```
FLUSH PRIVILEGES;
```

Quitter MariaDB :

```
EXIT;
```

Se diriger dans le répertoire /var/www/html :

```
cd /var/www/html/
```

Installer WordPress sur Debian 11 :

Télécharger WordPress :

```
curl -O https://wordpress.org/latest.tar.gz
```

Décompression de l'archive de WordPress

```
tar -xvf latest.tar.gz
```

Suppression de l'archive :

```
rm latest.tar.gz
```

Attribution des droits au serveur Web :

```
chown -R www-data:www-data /var/www/html/wordpress
```

Rendez-vous dans le dossier WordPress :

```
cd wordpress
```

Copie du fichier de configuration :

```
cp wp-config-sample.php wp-config.php
```

Édition du fichier de configuration :

```
// ** MySQL settings - You can get this info from your web host ** //  
/** The name of the database for WordPress */  
define( 'DB_NAME', 'wordpress_db' );  
  
/** MySQL database username */  
define( 'DB_USER', 'wordpress_user' );  
  
/** MySQL database password */  
define( 'DB_PASSWORD', 'WordPress@1234' );  
  
/** MySQL hostname */  
define( 'DB_HOST', 'localhost' );  
  
/** Database Charset to use in creating database tables. */  
define( 'DB_CHARSET', 'utf8' );  
  
/** The Database Collate type. Don't change this if in doubt. */  
define( 'DB_COLLATE', '' );
```

Suppression du fichier readme.html

```
rm /var/www/html/wordpress/readme.html
```

Installation de modules php pour WordPress :

```
apt install php-gd php-intl php-mbstring php-imagick
```

Aller modifier le fichier de config Apache et bouger la racine d'apache dans le dossier wordpress

```
GNU nano 7.2 /etc/apache2/sites-available/000-de
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/wordpress

# 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
</VirtualHost>
```

Puis redémarrer Apache :

```
systemctl restart apache2
```

Depuis votre navigateur vous devriez accéder à la configuration de WordPress :

```
http://votreip/wordpress
```

Avant la publication de WordPress sur le Web, il est conseillé de faire quelques opérations en plus d'un serveur Web. La première chose à faire est d'activer une extension qui empêchera les tentatives d'attaque par brute force. La seconde sera de modifier l'accès admin par un autre emplacement, des extensions permettent de le faire. La troisième sera de ne pas avoir de comptes qui contiennent « admin » ou des mots de passe faibles.

Je déconseille l'hébergement d'un site WordPress sur votre réseau, le moindre oubli de sécurité peut laisser la possibilité à un attaquant de contrôler votre machine / réseau.

WordPress Désactiver wp_xmlrpc

Le XML-RPC permet à WordPress de pouvoir communiquer avec d'autres applications, le soucis de XML-RPC est qu'il est également utiliser pour les attaques sur les sites WordPress. Dans cette procédures, je vais vous montrer comment désactiver wp_xmlrpc dans WordPress. Si wp_xmlrpc n'est pas utilisé, il est impératif à désactiver car il représente une menace de sécurité.



WORDPRESS

Prérequis :

- Avoir un site WordPress
- Avoir un accès de modification sur les fichiers de WordPress

Désactiver wp_xmlrpc dans WordPress avec Apache :

Pour désactiver wp_xmlrpc dans WordPress, il faut interagir avec le fichier .htaccess qui se trouve à la racine de votre site, alors avant de modifier quoi que ce soit faites une copie de se fichier avant de le modifier. En cas de mauvaise manipulation vous aurez juste à restaurer le fichier de base pour revenir à la normale.

Commande pour copier le fichier :

```
cp .htaccess .htaccess.sauvegarde
```

Ensuite ouvrir le fichier .htaccess avec un éditeur de texte

```
nano .htaccess
```

Puis coller cette configuration à la fin de votre fichier :

```
# BEGIN Disable XML-RPC request
<Files xmlrpc.php>
  order allow,deny
  deny from all
</Files>
# END Disable XML-RPC
```

```
# BEGIN Disable XML-RPC request
<Files xmlrpc.php>
order allow,deny
deny from all
</Files>
# END Disable XML-RPC
```

Une fois que vous avez changé cette configuration, wp_xmlrpc de votre site Web sera désactiver.

“ Sources :

<https://ws.apache.org/xmlrpc/>

Après avoir fait ces modifications dans le fichier .htaccess, il se peut que des services que vous utiliser ne fonctionnent plus.

Si vous souhaitez revenir en arrière, soit vous devez retirer la configuration qui a été ajouter dans le fichier .htaccess ou sinon il faudra restauré le fichier .htaccess.sauvegarde.

Ou vous pouvez également modifier votre structure de la manière suivant :

```
# BEGIN Disable XML-RPC request
<Files xmlrpc.php>
  order allow,deny
  deny from all
  allow from xxx.xxx.xxx.xxx
</Files>
# END Disable XML-RPC
```

En remplaçant xxx.xxx.xxx.xxx par l'ip de votre service

WordPress référencement naturel (SEO)

Dans cette procédure, je vais vous expliquer comment améliorer son référencement naturel avec WordPress. Posséder un site WordPress n'est pas quelque chose de compliqué à obtenir. Mais par contre, avoir des visiteurs à l'aide de référencement naturel n'est aussi simple à mettre en œuvre. Le Référencement naturel est une pratique qui consiste à améliorer son positionnement dans les moteurs de recherche sans financer de la publicité (ex : Google Ads).



WORDPRESS

Prérequis pour améliorer son référencement naturel avec WordPress :

- Avoir un site WordPress qui est indexé sur plusieurs moteurs de recherche (ex: Google, Bing, etc.)

Optimiser son référencement naturel avec WordPress :

Le référencement naturel est aussi appelé SEO (**Search Engine Optimisation**), avoir un bon SEO nécessite quelques bonnes pratiques à mettre en œuvre.

Voici ma liste des pratiques à utiliser pour obtenir un meilleur positionnement :

- Un chargement rapide du site web
- Du contenu inédit et régulier
- Ciblé des mots-clés
- Effectuer des maillages interne et externe
- Utiliser correctement les Images

Un chargement rapide du site Web :

Afin d'avoir un site rapide avec WordPress, il faut un hébergeur de qualité, mais pas seulement et compresser son contenu à l'aide d'outils tel que [Gzip](#).

Il faut également se protéger contre les cyberattaques, je vous conseille d'utiliser les outils suivants : [Crowdsec](#), [Change wp-admin login](#) et [Limit Login Attempts Reloaded](#).

Ensuite, il faut utiliser des [CDN](#) pour délivrer son site plus rapidement.

Il est possible de mesurer la vitesse de son site à l'aide des outils suivants :

- [PageSpeed Insights](#)
- [GTmetrix](#)
- [Vérificateur de compression Gzip \(Websiteplanet\)](#)

Du contenu inédit et régulier :

Je vous conseille de publier au moins une fois par semaine un article, que vous avez écrit vous-même (sans faire de copier/coller).

Il est préférable de rien poster plutôt que de faire un copier/coller d'un article de quelqu'un d'autre.

Cibler des mots-clés :

Utiliser les bons mots-clés permet d'obtenir plus d'impression/clics sur son article.

Certains outils mesurent les performances comme [Ubersuggest](#) ou [Google Analytics](#) / [Matomo](#).

Effectuer des maillages interne et externe :

Dans vos pages et articles, n'hésitez pas à renvoyer le visiteur sur d'autres pages de votre site.

Vous pouvez par ailleurs renvoyer des visiteurs vers les sites de vos contacts afin de rediriger vos visiteurs.

Utiliser correctement les Images :

Pour utiliser correctement les images, il faut utiliser un nom de fichier qui correspond à l'image et au mot-clé de l'article.

Renseigner un Texte alternatif qui, comme le nom de l'image, correspond à l'image et au mot-clé de l'article.

“ Sources :

<https://www.seo.fr/definition/seo-definition>

WordPress Ajouter un Sitemap

Un Sitemap est une page qui permet de trouver l'ensemble des pages d'un site web. Il existe les Sitemap Xml utilisés pour le référencement et les Sitemap Html utilisés par les utilisateurs du site web. Dans cette procédure nous allons voir comment créer ces deux types de Sitemap sur un site WordPress à l'aide de plugins. Ajouter un Sitemap sur son site WordPress est utile pour le référencement SEO et faciliter l'utilisation du site pour les utilisateurs. Le Sitemap est aussi important pour indexer toutes les pages de son site web sans en oublier.



WORDPRESS

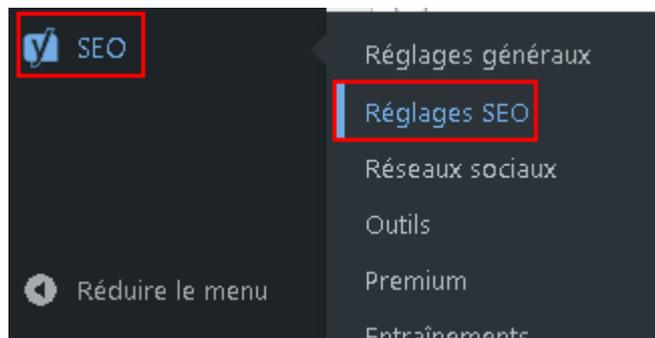
Prérequis :

- Une site WordPress

WordPress Ajouter un Sitemap XML :

Pour ajouter un Sitemap XML sur son site, il faut installer et activer l'extension **Yoast SEO**

Ensuite, il faudra se rendre dans la catégorie **SEO -> Réglages SEO -> Fonctionnalités**



Réglages généraux - Yoast SEO

Tableau de bord

Fonctionnalités

Intégrations

Outils pour les webmasters

✓ Problèmes (0)

Beau travail ! Nous n'avons détecté aucun problème sérieux de référencement.

✓ Notifications (0)

Aucune nouvelle notification.

Crédits

[Voir qui a contribué à Yoast SEO.](#)

Puis ensuite activé la fonctionnalité : **Plans de site XML**

Plans de site XML ?

Activé

Désactivé

Ensuite le sitemap sera présent à l'emplacement : https://votresite.com/sitemap_index.xml

XML Sitemap

Generated by **Yoast SEO**, this is an XML Sitemap, meant for consumption by search engines.

You can find more information about XML sitemaps on sitemaps.org.

This XML Sitemap Index file contains 4 sitemaps.

Sitemap	Last Modified
https://aymeric-cucherousset.fr/post-sitemap.xml	2021-12-18 12:30 +00:00
https://aymeric-cucherousset.fr/page-sitemap.xml	2021-12-11 11:06 +00:00
https://aymeric-cucherousset.fr/category-sitemap.xml	2021-12-18 12:30 +00:00
https://aymeric-cucherousset.fr/author-sitemap.xml	2021-10-24 08:20 +00:00

WordPress Ajouter un Sitemap HTML :

Pour ajouter un Sitemap HTML, il faudra installer et activer l'extension : **WP Sitemap Page**

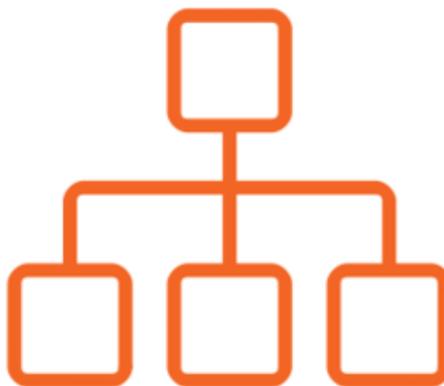
Ensuite il faudra créer une nouvelle page dans cette page il faudra ajouter sur cette page le short code suivant :

```
[wp_sitemap_page]
```

Sitemap

Je mets à disposition de chaque visiteur le **Sitemap** (Plan du site) afin de favoriser vos recherches et de trouver facilement chaque ressource. Le plan du site vous permet d'avoir accès à toutes les pages du site web. Le Siten est aussi utilisé dans le but de favoriser le référencement SEO du site Web dans le but d'indexer toutes les pages de son site Web. De plus il existe une version XML de cette page [ici](#).

```
[wp_sitemap_page]
```



Pour chaque question vous avez la possibilité de me contacter via la page de [contact](#) ou à l'aide de mes réseaux sociaux comme [Instagram](#) / [LinkedIn](#) / [GitHub](#).

Ensuite le sitemap va être automatiquement générer sur la nouvelle page. Puis vous avez la possibilité de mettre dans votre footer un lien vers votre Sitemap.

Avoir un SiteMap Xml ou Html pour son site est important car il permet moteur de recherches d'indexer toutes les pages du site.

Un fois vos Sitemaps générés vous pouvez les envoyer aux moteurs de recherches pour qu'ils puissent indexer toutes les pages de votre site web.

Il est possible de mettre un lien du Sitemap Html dans votre footer pour le rendre accessible à proximité des CGU.

Sources :

<https://fr.wordpress.org/>

WordPress et Google Analytics

Dans cette procédure, je vais vous montrer comment lier son site web **WordPress** et **Google Analytics**. La réalisation de cette procédure sera utiliser sans utiliser de plugins WordPress. Installer Google Analytics permet de suivre les actions des utilisateurs de son site Web.



Google Analytics

Prérequis :

- Un site WordPress
- Un thème enfant
- Les accès administrateurs du tableau de bord

Installation de Google Analytics sur WordPress

Pour commencer il faut se connecter un compte avec un compte Google sur le site de [Google Analytics](#).

Une fois connecter sur Google Analytics il faut aller dans les paramètres en bas à gauche de la fenêtre.

Puis il faudra créer une propriété et répondre aux questions que Google Analytics va vous demander.

Une fois la propriété créer, rendez-vous sur l'assistant de configuration -> installation de la balise -> Ajouter un flux -> Web

Saisissez la configuration de votre site Web :

Tagging Instructions
Utilisez l'une des méthodes suivantes pour commencer à collecter des données.

[Ajouter une nouvelle balise sur la page](#) [Utiliser la balise sur la page existante](#)

Global site tag (gtag.js) Choisissez cette balise si vous utilisez un outil de création de sites Web ou un site hébergé par un CMS. Ajoutez la balise Analytics à votre site Web et commencez à visualiser des données dans votre propriété.

Copy the global site tag into the **<head>** section of your HTML. Or, if you use a website builder (e.g. GoDaddy, Shopify, etc), [copy the global site tag into your website builder's custom HTML field](#).

```
<!-- Global site tag (gtag.js) - Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/js?id=G-KKDDPVZPM3"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());

  gtag('config', 'G-KKDDPVZPM3');
</script>
```

Google Tag Manager
Ajoutez et gérez des balises via une interface Web afin d'envoyer des données à Google Analytics ainsi qu'à d'autres outils Google ou tiers.

Récupérer le contenu du Global site tag

Tagging Instructions
Utilisez l'une des méthodes suivantes pour commencer à collecter des données.

[Ajouter une nouvelle balise sur la page](#) [Utiliser la balise sur la page existante](#)

Global site tag (gtag.js) Choisissez cette balise si vous utilisez un outil de création de sites Web ou un site hébergé par un CMS. Ajoutez la balise Analytics à votre site Web et commencez à visualiser des données dans votre propriété.

Copy the global site tag into the **<head>** section of your HTML. Or, if you use a website builder (e.g. GoDaddy, Shopify, etc), [copy the global site tag into your website builder's custom HTML field](#).

```
<!-- Global site tag (gtag.js) - Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/js?id=G-KKDDPVZPM3"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());

  gtag('config', 'G-KKDDPVZPM3');
</script>
```

Google Tag Manager
Ajoutez et gérez des balises via une interface Web afin d'envoyer des données à Google Analytics ainsi qu'à d'autres outils Google ou tiers.

Puis ensuite il faudra ajouter une fonction dans votre thème enfant :

```
/**
 * Ajout du code de suivi de Google Analytics dans le Header avant la fermeture de la balise Head
 */
add_action('wp_head', 'add_google_analytics');
function add_google_analytics() {
    <?>
    <!-- Global site tag (gtag.js) - Google Analytics -->
    <script async src="https://www.googletagmanager.com/gtag/js?id=G-5P696FEXTB"></script>
    <script>
```

```
window.dataLayer = window.dataLayer || [];  
function gtag(){dataLayer.push(arguments);}  
gtag('js', new Date());  
  
gtag('config', 'G-XXXXXXXXXX');  
</script>  
<?php }  
/**  
 * Fin Google Analytics  
 */
```

Puis remplacer ce script gtag.js par celui que Google Analytics vous a donné.

Les modifications seront effectives quelques minutes après la modification.

“ Sources :

<https://fr.wordpress.org/>

WordPress Thème enfant

Dans cette procédure je vais vous expliquer comment personnaliser un thème **WordPress** à l'aide d'un **thème enfant**. Après avoir suivi la procédure vous serez en mesure de personnaliser votre thème principale **WordPress** grâce a un **thème enfant**. L'utilisation d'un thème enfant permet de réaliser des modifications et de les conserver après les mises à jour thème.



Prérequis :

- Un site web WordPress
- Les accès SSH ou FTP de l'hébergeur
- Les accès Administrateur de WordPress

Création du Thème Enfant WordPress :

A l'aide des accès SSH ou FTP connectez vous à l'emplacement de votre site sur la machine qui héberge le site.

Puis rendez-vous dans l'emplacement : `wp-content/themes`

Ensuite créer un dossier qui s'appellera `le-nom-de-votre-theme-child`

Après la création de ce dossier, il faut créer deux fichiers dans ce dossier :

- `style.css`
- `functions.php`

Puis dans le fichier « `style.css` », il faudra l'adapter à votre thème avec :

- **Theme Name** : Le nom de votre thème enfant
- **Description** : La description du thème
- **Author** : Le nom de l'auteur du thème enfant
- **Author URI** : Un lien de site du thème enfant
- **Template** : Le thème parent
- **Version** : La version du thème enfant

```
/*
Theme Name: astra-child
Description: Modification du theme parent.
Author: Cucherousset Aymeric
Author URI: https://aymeric-cucherousset.fr
Template: astra
Version: 1.0
*/
```

Après ce commentaire vous pouvez y ajouter vos modifications de CSS.

Ensuite il faut modifier le fichier `functions.php` pour activer le thème enfant :

Le code pour activer le thème peut être différent d'un thème parent à un autre.

```
<?php

/**
** activation theme
**/

add_action( 'wp_enqueue_scripts', 'theme_enqueue_styles' );
function theme_enqueue_styles() {
    wp_enqueue_style( 'parent-style', get_template_directory_uri() . '/style.css' );
    wp_enqueue_style( 'astra-child', get_stylesheet_uri() );
}

?>
```

Pour rendre effectif le thème il ne reste plus qu'à se rendre dans le panel admin de WordPress.

Il faut ensuite aller dans Apparences -> Thèmes.

Puis activer le thème enfant. Désormais les modifications faites sur le fichier (Accessible depuis Apparences -> Editeur de thèmes) du thème enfant seront prises en compte sur le thème et seront conservés au mises à jour du thème.

📖 Sources :

<https://fr.wordpress.org/>