

Comment installer WordPress avec Nginx et Let's Chiffrer SSL sur Ubuntu 22.04

WordPress CMS est un système de gestion de contenu gratuit et open source qui permet aux utilisateurs de créer et de gérer des sites Web. Il est connu pour sa flexibilité, son interface conviviale et sa vaste bibliothèque de plugins et de thèmes, ce qui le rend idéal à la fois débutants et professionnels.

WordPress facilite la création et la maintenance de sites Web et de blogs. En raison de sa popularité, plus d'un tiers des sites Web aujourd'hui sont propulsés par WordPress. Il est écrit en PHP et utilise MariaDB et MySQL comme backend de base de données.

Dans ce tutoriel, nous allons vous montrer comment installer WordPress avec Nginx et un certificat SSL Let's Encrypt gratuit sur Ubuntu 22.04.

Conditions préalables

- Un serveur exécutant Ubuntu 22.04.
- Un nom de domaine valide est pointé avec l'adresse IP de votre serveur.
- Un mot de passe root est configuré sur le serveur.

Installez Nginx, MariaDB et PHP

Avant de démarrer, le serveur LEMP doit être installé sur votre serveur. S'il n'est pas installé, vous pouvez l'installer en exécutant la commande suivante :

```
apt install nginx mariadb-server php php-fpm php-curl php-mysql php-gd php-mbstring php-xml php-imagick php-zip php-xmllrpc -y
```

Une fois le serveur LEMP installé, vérifiez la version PHP à l'aide de la commande suivante :

```
php -v
```

Vous obtiendrez la version PHP dans le résultat suivant :

```
PHP 8.1.2 (cli) (built: Apr 7 2022 17:46:26) (NTS) Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
with Zend OPcache v8.1.2, Copyright (c), by Zend Technologies
```

Ensuite, modifiez le fichier de configuration PHP et modifiez certains paramètres par défaut :

```
nano /etc/php/8.1/fpm/php.ini
```

Modifiez les lignes suivantes :

```
cgi.fix_pathinfo=0
upload_max_filesize = 128M
post_max_size = 128M
memory_limit = 512M
max_execution_time = 120
```

Enregistrez et fermez le fichier lorsque vous avez terminé.

Créer une base de données pour WordPress

WordPress utilise une base de données pour stocker son contenu. Vous devez donc créer une base de données et un utilisateur pour WordPress.

Tout d'abord, connectez-vous au shell MariaDB avec la commande suivante :

```
mysql
```

Une fois connecté, créez une base de données et un utilisateur avec la commande suivante :

```
MariaDB [(none)]> CREATE DATABASE wpdb;
MariaDB [(none)]> CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'securepassword';
```

Ensuite, accordez tous les privilèges à la base de données WordPress à l'aide de la commande suivante :

```
MariaDB [(none)]> GRANT ALL ON wpdb.* TO 'wpuser'@'localhost';
```

Ensuite, videz les privilèges et quittez MariaDB avec la commande suivante :

```
MariaDB [(none)]> FLUSH PRIVILEGES;
MariaDB [(none)]> EXIT;
```

Une fois que vous avez terminé, vous pouvez passer à l'étape suivante.

Installez WordPress sur Ubuntu 22.04

Tout d'abord, accédez au répertoire racine Web de Nginx et téléchargez la dernière version de WordPress à l'aide de la commande suivante :

```
cd /var/www/html
wget https://wordpress.org/latest.tar.gz
```

Une fois WordPress téléchargé, extrayez le fichier téléchargé avec la commande suivante :

```
tar -zxvf latest.tar.gz
```

Ensuite, renommez l'exemple de fichier de configuration WordPress.

```
mv /var/www/html/wordpress/wp-config-sample.php /var/www/html/wordpress/wp-config.php
```

Ensuite, éditez le fichier de configuration WordPress et définissez les paramètres de votre base de données :

```
nano /var/www/html/wordpress/wp-config.php
```

Définissez les paramètres de votre base de données comme indiqué ci-dessous :

```
define( 'DB_NAME', 'wpdb' );

/** Database username */
define( 'DB_USER', 'wpuser' );

/** Database password */
define( 'DB_PASSWORD', 'securepassword' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );
```

Pour des raisons de sécurité, vous devrez également mettre à jour les clés de sécurité dans votre fichier wp-config. Tout d'abord, va [ici](#) pour les générer. Ensuite, ajoutez-le comme indiqué ci-dessous :

```
define( 'AUTH_KEY', 'Y$I, -gafVeR>Z-8qy&jQ62L}{R)e|LK/#RBh.Y#f+p-P*.8,,hP-iX[q3*tVP-fu' );
define( 'SECURE_AUTH_KEY', 'D)k6o`D G%<()-zXP5{T{v2}Zgo-c+8T-Un=+R3%n/X2=MLDb5$0]UHA%gK| .WR' );
define( 'LOGGED_IN_KEY', 'eL|5#`ul|;MrKm#q$KVL/ky(i)Jc;xrH(Eb|Hwzb/?- .RLSUSX2X[4HD:U:UOP:Y' );
define( 'NONCE_KEY', ' ]azQ+9f^#~l*r>uPMH5H>ck:?az4o{)*Txo:+MGjE5f&0kag309m85g3~VJ6YVWE' );
define( 'AUTH_SALT', 'fAM5&`m4X+{+wSsF. !}-/8Ce~u%>}la1bCC,@#+R*t]uYf?[hph/>!Bw>v#oaQ' );
define( 'SECURE_AUTH_SALT', '}|Z&dj_tFV2T$7y(0#0|bwwQ$sh6t!-zdE.MLOHLZ>4WDqG:_Qzn#Allm-U01#7P' );
define( 'LOGGED_IN_SALT', 'b9Uf~**E_xt@{KwksAL^9D7Ix3C0.+PpFF~btd)-pG~pXPQ,[c&WRE-NgLG9~)|' );
define( 'NONCE_SALT', ' }mTUi&. #i+YJT-TSrbIwqW0<|ut3K%CS-7g.} *NztVlgZDr`?>wxJ+_VW-D_zif' );
```

Enregistrez et fermez le fichier lorsque vous avez terminé. Ensuite, définissez l'autorisation et la propriété appropriées sur le répertoire WordPress :

```
chown -R www-data:www-data /var/www/html/wordpress
chmod -R 755 /var/www/html/wordpress
```

Créer un hôte virtuel Nginx pour WordPress

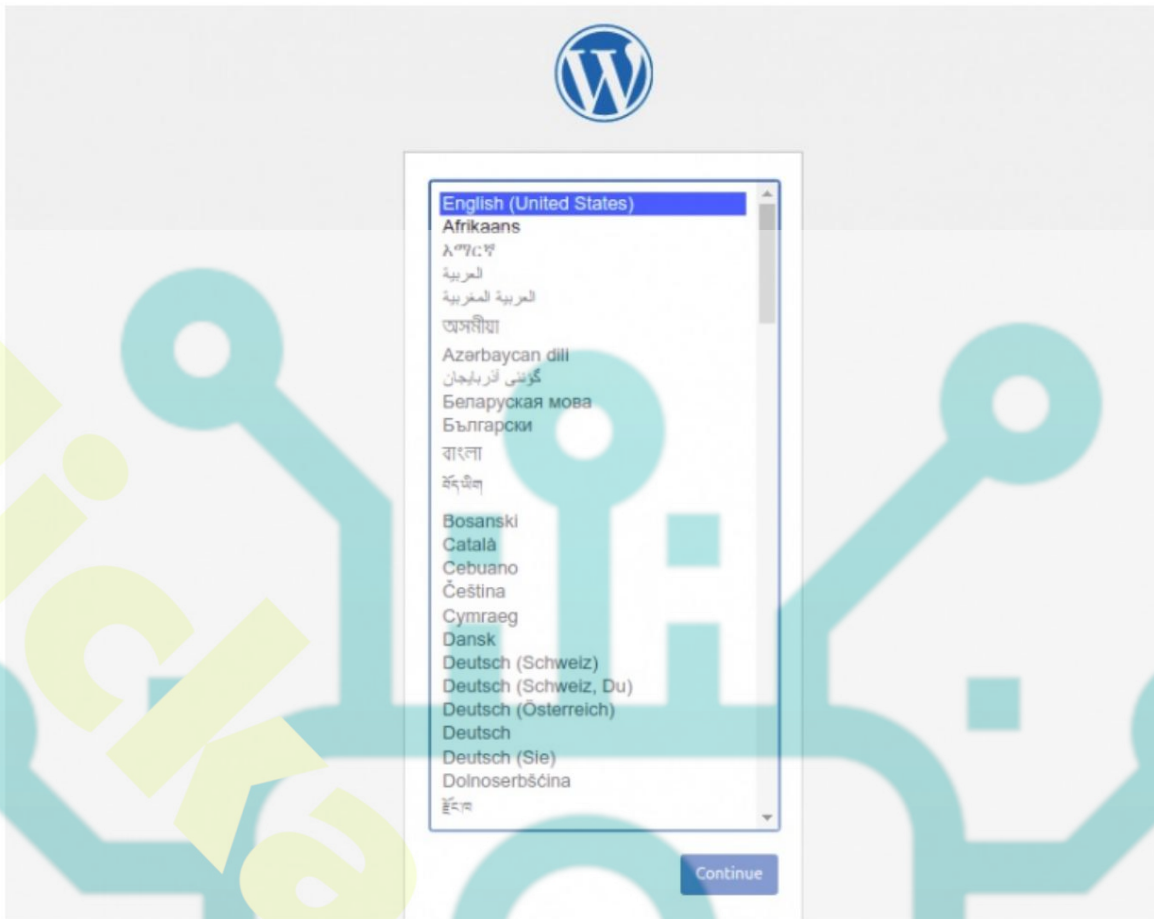
Ensuite, vous devrez créer un fichier de configuration d'hôte virtuel Nginx pour servir WordPress sur Internet.

```
nano /etc/nginx/conf.d/wordpress.conf
```

Ajoutez la configuration suivante :

```
server {
    listen 80;
    root /var/www/html/wordpress;
    index index.php index.html index.htm;
    server_name wordpress.example.com;

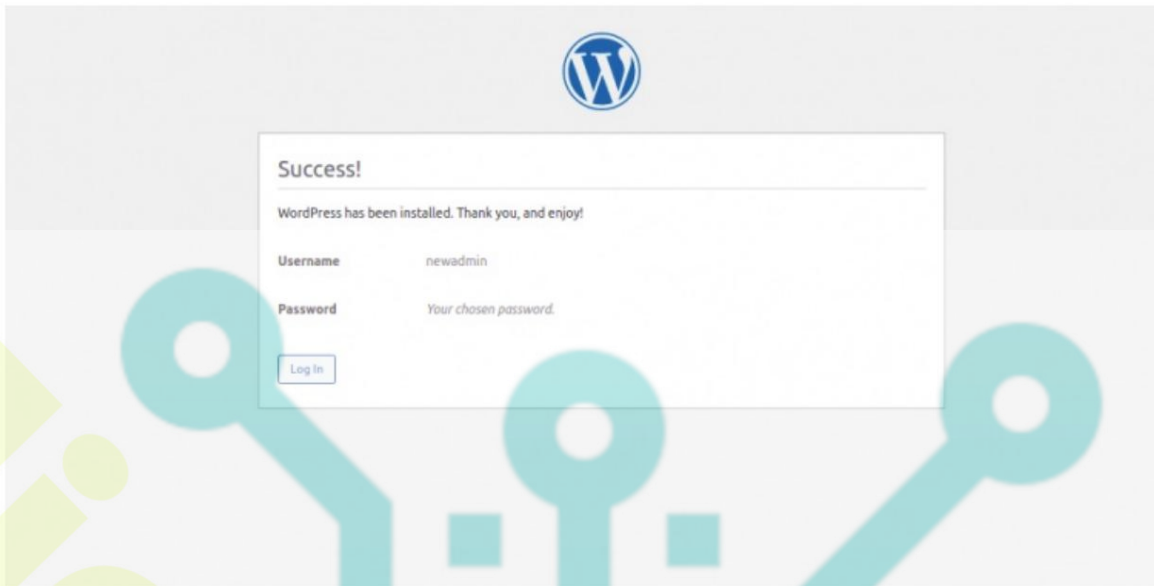
    client_max_body_size 500M;
```

Sélectionnez votre langue et cliquez sur le bouton Continuer . Vous devriez voir la page de configuration du site WordPress :



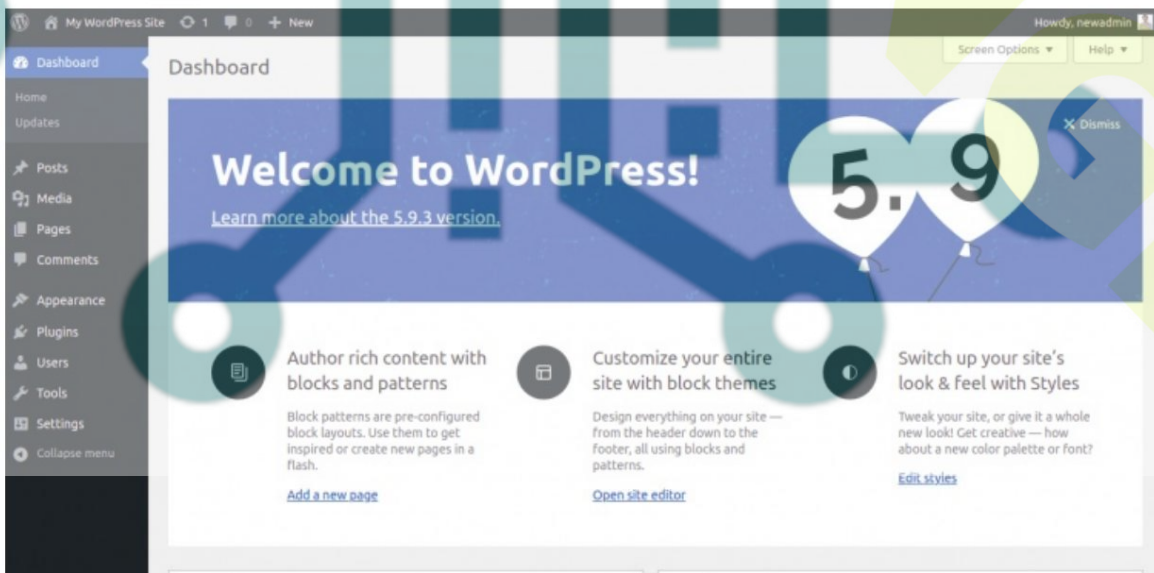
Fournissez le nom de votre site Web, votre nom d'utilisateur administrateur, votre mot de passe, votre adresse e-mail et cliquez sur le bouton Installer WordPress . Une fois WordPress installé, vous devriez voir la page suivante :



Cliquez sur le bouton Connexion . Vous devriez voir la page de connexion WordPress :



Fournissez votre nom d'utilisateur administrateur, votre mot de passe et cliquez sur le bouton Connexion . Vous devriez voir le tableau de bord WordPress sur la page suivante :



Activer HTTPS sur WordPress

Pour activer HTTPS sur votre site, vous devrez installer le client Certbot Let's Encrypt sur votre système. Vous pouvez l'installer en exécutant la commande suivante :

```
apt-get install python3-certbot-nginx -y
```

Une fois le client Certbot installé, exécutez la commande suivante pour activer le HTTPS sur votre site Web :

```
certbot --nginx -d wordpress.example.com
```

Il vous sera demandé de fournir une adresse e-mail valide et d'accepter les conditions d'utilisation comme indiqué ci-dessous :

```
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator nginx, Installer nginx
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): hitjethva@gmail.com
```

```
-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
-----
```

```
(A)gree/(C)ancel: A
```

```
-----
Would you be willing to share your email address with the Electronic Frontier
Foundation, a founding partner of the Let's Encrypt project and the non-profit
organization that develops Certbot? We'd like to send you email about our work
encrypting the web, EFF news, campaigns, and ways to support digital freedom.
-----
```

```
(Y)es/(N)o: Y
```

```
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for wordpress.example.com
Waiting for verification...
Cleaning up challenges
Deploying Certificate to VirtualHost /etc/nginx/conf.d/wordpress.conf
```

Ensuite, choisissez si vous souhaitez ou non rediriger le trafic HTTP vers HTTPS, comme indiqué ci-dessous :

- ```

1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for new sites, or if you're confident your
site works on HTTPS. You can undo this change by editing your web server's configuration.
```

Tapez 2 et appuyez sur Entrée pour terminer l'installation. Vous devriez voir le résultat suivant :

```
Redirecting all traffic on port 80 to ssl in /etc/nginx/conf.d/wordpress.conf
```

```

Congratulations! You have successfully enabled https://wordpress.example.com
```

```
You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=wordpress.example.com

```

#### IMPORTANT NOTES:

- Congratulations! Your certificate and chain have been saved at:  
/etc/letsencrypt/live/wordpress.example.com/fullchain.pem  
Your key file has been saved at:  
/etc/letsencrypt/live/wordpress.example.com/privkey.pem  
Your cert will expire on 2023-02-08. To obtain a new or tweaked version of this certificate in the future, simply run certbot again with the "certonly" option. To non-interactively renew \*all\* of your certificates, run "certbot renew"
- Your account credentials have been saved in your Certbot configuration directory at /etc/letsencrypt. You should make a secure backup of this folder now. This configuration directory will also contain certificates and private keys obtained by Certbot so making regular backups of this folder is ideal.
- If you like Certbot, please consider supporting our work by:

Donating to ISRG / Let's Encrypt: <https://letsencrypt.org/donate>  
Donating to EFF: <https://eff.org/donate-le>

- We were unable to subscribe you the EFF mailing list because your e-mail address appears to be invalid. You can try again later by visiting <https://act.eff.org>.

## Conclusion

Toutes nos félicitations! Vous avez installé avec succès WordPress avec Nginx et Let's Encrypt SSL sur Ubuntu 22.04. Vous pouvez maintenant installer vos thèmes et plugins préférés et commencer à créer votre propre site Web.

---

