

# Installation WordPress sur Debian et système de sauvegardes automatique

<b>Installation Wordpress.....</b>	<b>2</b>
Mise à jour de Debian.....	2
Installation de AMP, unzip et openssl.....	2
Vérifications du fonctionnement des logiciels installés.....	2
Pour éviter une erreur ajouter << ServerName 127.0.0.1>> à la fin du fichier /etc/apache2/apache2.conf.....	2
Protection de la base de données et certificat SSL auto-signé.....	3
<b>Création de la base de données.....</b>	<b>4</b>
<b>Installation de Wordpress et configuration.....</b>	<b>4</b>
Modification des permissions.....	4
Modifier le nom de la base, l'identifiant ainsi que le mot de passe.....	4
Création d'un virtual host.....	5
Activation des modules virtual host.....	5
Pour accéder à Wordpress il suffit de rentrer l'adresse ci-dessous dans un navigateur web : .....	5
<b>Création d'une sauvegarde du site.....</b>	<b>6</b>
Sauvegarde des données du site.....	6
Sauvegardes de la base de données.....	6
<b>Transfert du backup sur une autre machine.....</b>	<b>6</b>
Transfert des sauvegardes grâce au protocole sftp.....	6
<b>Automatisation des transferts.....</b>	<b>7</b>
Création clé/certificat entre les deux machines.....	7
Création d'un script bash qui crée les sauvegardes et les transfère sur la machine backup.....	7
Ajout d'une redondance du script.....	8
Création du script pour supprimer les sauvegardes de la machine backup.....	8
Ajout d'une redondance du script.....	8

# Installation Wordpress

## Mise à jour de Debian

*cd / && apt update && apt upgrade*

## Installation de AMP, unzip et openssl

*apt install apache2 mariadb-server php8.2 php8.2-cli php8.2-common php8.2-imagick php8.2-redis php8.2-snmp php8.2-xml php8.2-mysql php8.2-zip php8.2-mbstring php8.2-curl libapache2-mod-php unzip openssl*

```
sto@debian12:/$ sudo apt install apache2 mariadb-server php8.2 php8.2-cli php8.2-common php8.2-imagick php8.2-redis php8.2-snmp php8.2-xml php8.2-mysql php8.2-zip php8.2-mbstring php8.2-curl libapache2-mod-ph
p unzip openssl
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Note : sélection de « php8.2-mysql » au lieu de « php8.2-mysqli »
openssl est déjà la version la plus récente (3.0.15-1-debian12u1).
openssl passé en « installé manuellement ».
Les paquets supplémentaires suivants seront installés :
  apache2-bin apache2-data apache2-utils galera-4 gawk libapache2-mod-php8.2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libc-client2007e libcurl4 libdaxctl1 libdbd-mariadb-perl libdbi-perl libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi9ldbl libgpm2 libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblua5.3-0 liblwp-mediatypes-perl liblz02-2 libmariadb3 libmpfr6 libncurses6 libndctl6 libnuma1 libonig5 libpmem1
  libregexp-ipv6-perl libsensors-config libsensors5 libsigsegv2 libsnappy1v5 libsnmp-base libsnmp40 libsodium23 libterm-readkey-perl libtimedate-perl liburi-perl liburing2 libzip4 mariadb-client
  mariadb-client-core mariadb-common mariadb-plugin-provider-bzip2 mariadb-plugin-provider-lz4 mariadb-plugin-provider-lzma mariadb-plugin-provider-lzo mariadb-plugin-provider-snappy mariadb-server-core
  mlock mysql-common php-common php8.2-igbinary php8.2-opcache php8.2-readline psmisc pv rsync sgml-base snmp socat ssl-cert
Paquets suggérés :
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser gawk-doc php-pear uw-mailutils libltdb-perl libnet-daemon-perl libsql-statement-perl gpm libdata-dump-perl
  libipc-sharedcache-perl lm-sensors snmp-mibs-downloader libbusiness-isbn-perl libwww-perl mailx mariadb-test netcat-openbsd redis-server doc-base python3-braceexpand sgml-base-doc zip
Les NOUVEAUX paquets suivants seront installés :
  apache2 apache2-bin apache2-data apache2-utils galera-4 gawk libapache2-mod-php libapache2-mod-php8.2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libc-client2007e libcurl4 libdaxctl1 libdbd-mariadb-perl libdbi-perl libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi9ldbl libgpm2 libhtml-parser-perl
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblua5.3-0 liblwp-mediatypes-perl liblz02-2 libmariadb3 libmpfr6 libncurses6 libndctl6 libnuma1 libonig5
  libpmem1 libregexp-ipv6-perl libsensors-config libsensors5 libsigsegv2 libsnappy1v5 libsnmp-base libsnmp40 libsodium23 libterm-readkey-perl libtimedate-perl liburi-perl liburing2 libzip4 mariadb-client
  mariadb-client-core mariadb-common mariadb-plugin-provider-bzip2 mariadb-plugin-provider-lz4 mariadb-plugin-provider-lzma mariadb-plugin-provider-lzo mariadb-plugin-provider-snappy mariadb-server
  mariadb-server-core mlock mysql-common php-common php8.2 php8.2-cli php8.2-common php8.2-curl php8.2-igbinary php8.2-imagick php8.2-mbstring php8.2-mysql php8.2-opcache php8.2-readline php8.2-redis
  php8.2-snmp php8.2-xml php8.2-zip psmisc pv rsync sgml-base snmp socat ssl-cert unzip
0 mis à jour, 90 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 34,5 Mo dans les archives.
Après cette opération, 243 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] O
```

## Vérifications du fonctionnement des logiciels installés

*systemctl status apache2 && systemctl status mariadb && php -v*

Pour éviter une erreur ajouter << ServerName 127.0.0.1>> à la fin du fichier /etc/apache2/apache2.conf

## Protection de la base de données et certificat SSL auto-signé

### *mysql\_secure\_installation*

```
In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n
... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] Y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] Y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
sio@debian12:/$
```

```
openssl req -x509 -nodes -days 10000 -newkey rsa:2048 -keyout
/etc/ssl/private/wp-selfsigned.key -out /etc/ssl/certs/wp-selfsigned.crt
```

# Création de la base de données

```
mysql
CREATE USER 'wordpress'@'localhost' IDENTIFIED BY 'mot-de-passe';
CREATE DATABASE wordpress;
GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpress'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

## Installation de Wordpress et configuration

```
cd var/www/html
wget https://wordpress.org/latest.zip && unzip latest.zip
```

### Modification des permissions

```
chown -R www-data:www-data wordpress
cd wordpress
find . -type d -exec chmod 755 {} \;
find . -type f -exec chmod 644 {} \;
```

```
mv wp-config-sample.php wp-config.php
```

### Modifier le nom de la base, l'identifiant ainsi que le mot de passe

```
nano wp-config.php
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

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

/** Database password */
define( 'DB_PASSWORD', 'mot-de-passe' );
```

## Création d'un virtual host

```
cd / && cd etc/apache2/sites-available  
nano wordpress.conf
```

```
<VirtualHost *:80>  
  ServerName ip-du-serveur  
  Redirect "/" "https://ip-du-serveur"  
</VirtualHost>  
  
<VirtualHost *:443>  
  ServerName ip-du-serveur  
  DocumentRoot /var/www/html/wordpress  
  
<Directory /var/www/html/wordpress>  
  AllowOverride All  
</Directory>  
  
ErrorLog ${APACHE_LOG_DIR}/error.log  
CustomLog ${APACHE_LOG_DIR}/access.log combined  
  
SSLEngine on  
  SSLCertificateFile /etc/ssl/certs/wp-selfsigned.crt  
  SSLCertificateKeyFile /etc/ssl/private/wp-selfsigned.key  
</VirtualHost>
```

## Activation des modules virtual host

```
a2enmod rewrite && a2enmod ssl && a2ensite wordpress.conf && reboot
```

Pour accéder à Wordpress il suffit de rentrer l'adresse ci-dessous dans un navigateur web :

<https://adresse-du-serveur>

# Création d'une sauvegarde du site

## Sauvegarde des données du site

```
tar czvf /home/sio/backup-$(date +%Y%m%d).tar.gz /var/www/html/wordpress/
```

## Sauvegardes de la base de données

```
mysqldump -u root wordpress > /home/sio/bddwordpress-$(date +%Y%m%d).sql
```

# Transfert du backup sur une autre machine

## Transfert des sauvegardes grâce au protocole sftp

```
sftp nom-de-session@adresse-de-autre-machine
```

```
sio@debian12:~$ sudo sftp sio@192.168.66.129
The authenticity of host '192.168.66.129 (192.168.66.129)' can't be established.
ED25519 key fingerprint is SHA256:4kGgsGeDEapDIPyUKe6N2vhl6o9ogMdBE3AWx03nIPk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.66.129' (ED25519) to the list of known hosts.
sio@192.168.66.129's password:
Connected to 192.168.66.129.
sftp> put /home/sio/dataw
datawordpress-20241115.tar.gz      datawordpress.tar.gz
sftp> put /home/sio/datawordpress-20241115.tar.gz
Uploading /home/sio/datawordpress-20241115.tar.gz to /home/sio/datawordpress-20241115.tar.gz
datawordpress-20241115.tar.gz
sftp> put /home/sio/wordpress-2024-11-15.sql
Uploading /home/sio/wordpress-2024-11-15.sql to /home/sio/wordpress-2024-11-15.sql
wordpress-2024-11-15.sql
```

# Automatisation des transferts

## Création clé/certificat entre les deux machines

`ssh-keygen`

`ssh-copy-id nom-de-session@adresse-de-autre-machine`

```
sio@debian12:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/sio/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/sio/.ssh/id_rsa
Your public key has been saved in /home/sio/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:72Xq0ReTDIG9HkVTOHTv05kufAYmsRmx1tnwkwDCKrA sio@debian12
The key's randomart image is:
+----[RSA 3072]-----+
|      ..+.      |
|      ..0+.     |
|      .+.X.     |
|   o  .  =Xo*   |
|   E .  =Xo*   |
|      .S.B*+    |
|      .+B.o     |
|      .+*+      |
|      .B.B+     |
|      .+o.      |
+----[SHA256]-----+
sio@debian12:~$ ssh-copy-id
Usage: /usr/bin/ssh-copy-id [-h-?|-f|-n|-s] [-i [identity_file]] [-p port] [-F alternative_ssh_config_file] [[-o <ssh -o options>] ...] [user@]hostname
-f: force mode -- copy keys without trying to check if they are already installed
-n: dry run -- no keys are actually copied
-s: use sftp -- use sftp instead of executing remote-commands. Can be useful if the remote only allows sftp
-h|-?: print this help
sio@debian12:~$ ssh-copy-id sio@192.168.66.129
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/sio/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
sio@192.168.66.129's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'sio@192.168.66.129'"
and check to make sure that only the key(s) you wanted were added.
```

## Création d'un script bash qui crée les sauvegardes et les transfère sur la machine backup

```
#!/bin/bash
#variable date
date=$(date +%Y)-$(date +%m)-$(date +%d)-$(date +%H)-$(date +%M)

#archive du dossier wordpress
tar czvf /home/sio/backup-$date.tar.gz /var/www/html/wordpress/

#connexion sftp avec la machine backup et envoi de l'archive
sftp sio@192.168.66.129 <<<$'put /home/sio/backup-'$date'.tar.gz'

#suppression du backup wordpress
rm -r /home/sio/backup-$date.tar.gz

#mysqldump bdd wordpress
sudo mysqldump -u root wordpress > /home/sio/bddwordpress-$date.sql

#envoi bdd
sftp sio@192.168.66.129 <<<$'put /home/sio/bddwordpress-'$date'.sql'
```

```
#suppression du backup bdd
rm -r /home/sio/bddwordpress-$date.sql

#log sauvegarde
echo "sauvegarde effectuée le $date" >> /home/sio/backup.log
```

## Ajout d'une redondance du script

*crontab -e*

## Création du script pour supprimer les sauvegardes de la machine backup

```
#!/bin/bash
find /home/sio/b* -type f -mmin +5 -exec rm {} \;
```

## Ajout d'une redondance du script

*crontab -e*