



Suivi des modifications

<i>Version</i>	<i>Référence</i>	<i>Auteur</i>	<i>Date</i>	<i>Commentaires</i>
A	TP_Abo	Alexandre	30/08/2024	Création

Objet :

Mise en place d'un WordPress et sauvegarde de la base de données

Diffusion :

BTS SIO – Étudiants BTS SIO.

Développement :

Table des matières

INSTALLATION DE WORDPRESS	2
SAUVEGARDE DE LA BASE DE DONNÉES ET DU SITE.....	4

INSTALLATION DE WORDPRESS

En premier lieu nous devons installer WordPress, dans notre cas nous installerons WordPress sur un serveur LAMP sous Debian 12.

Pour installer WordPress, nous devons installer au préalable un serveur LAMP, pour ce faire nous allons taper ces ligne de commandes.

```
apt install apache2 mariadb-server php8.2 php8.2-cli php8.2-common php8.2-imap php8.2-redis php8.2-snmp php8.2-xml php8.2-mysqli php8.2-zip php8.2-mbstring php8.2-curl libapache2-mod-php unzip openssl
```

Une fois notre serveur LAMP installé, nous allons le paramétrer de sorte à ce qu'il sache qu'il est lui-même le serveur.

```
nano /etc/apache2/apache2.conf  
ServerName 127.0.0.1
```

À présent nous allons sécuriser la base de données.

```
mysql_secure_installation
```

Nous allons créer un certificat SSL auto-signé.

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

Bien, maintenant créons la base de données qui va accueillir notre WordPress.

```
mysql
```

```
CREATE USER 'wordpress'@'localhost' IDENTIFIED BY 'Sio2019+';  
CREATE DATABASE wordpress;  
GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpress'@'localhost';  
FLUSH PRIVILEGES;  
EXIT;
```

Bien sur le mot de passe est à changer, pour le TP le changement du mot de passe n'était pas nécessaire.

Nous allons nous déplacer dans le dossier html pour installer notre WordPress, télécharger le fichier zip et le dézipper.

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

Nous allons faire en que l'utilisateur d'apache www-data puisse avoir les droits pour notre WordPress.

```
chown -R www-data:www-data wordpress  
cd wordpress
```

```
find . -type d -exec chmod 755 {} \;  
find . -type f -exec chmod 644 {} \;
```

À présent nous allons paramétrer notre WordPress.

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

```
<?php  
/**  
 * The base configuration for WordPress  
 *  
 * The wp-config.php creation script uses this file during the installation.  
 * You don't have to use the website, you can copy this file to "wp-config.php"  
 * and fill in the values.  
 *  
 * This file contains the following configurations:  
 *  
 * * Database settings  
 * * Secret keys  
 * * Database table prefix  
 * * ABSPATH  
 *  
 * @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/  
 *  
 * @package WordPress  
 */  
  
// ** 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', 'Sio2019+' );  
  
/** Database 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', '' );
```

Maintenant nous allons activer le site WordPress.

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

```
GNU nano 7.2
<VirtualHost *:80>
  ServerName 192.168.66.128
  Redirect "/" "https://ip-de-votre-serveur/"
</VirtualHost>

<VirtualHost *:443>
  ServerName 192.168.66.128
  DocumentRoot /var/www/html/wordpress

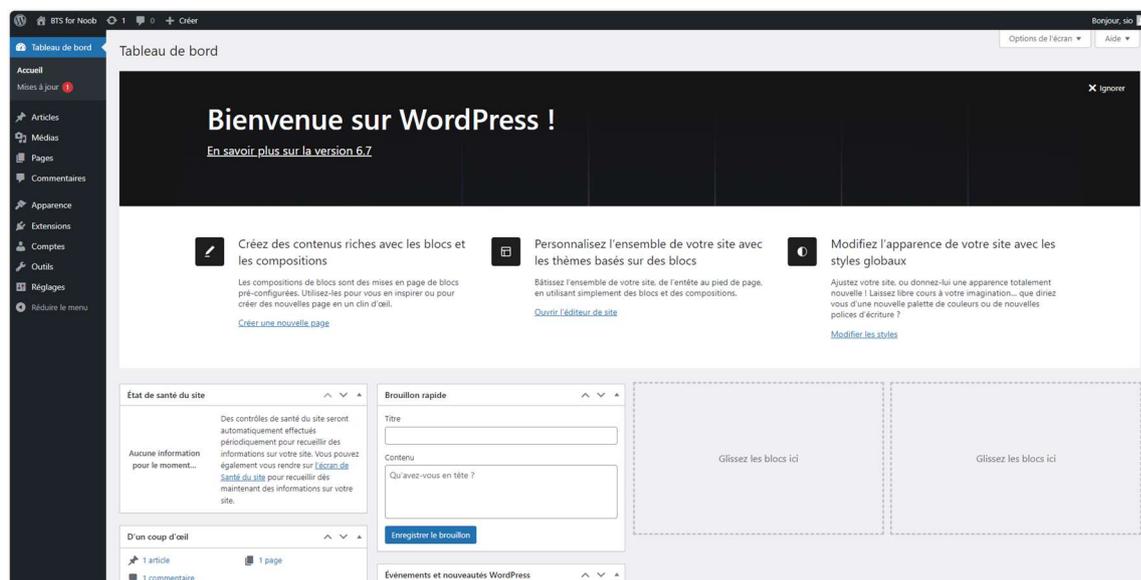
<Directory /var/www/html/wordpress>
  AllowOverride All
</Directory>

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>
```

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

Et voilà notre WordPress est installé.



SAUVEGARDE DE LA BASE DE DONNÉES ET DU SITE

À présent nous allons sauvegarder notre site et sa base de données sur un autre serveur.

En premier nous allons créer une clés de chiffrement pour pouvoir se connecter en sftp sans mot de passe.

```
sio@debian12:~$ ssh ssh-keygen
ssh: Could not resolve hostname ssh-keygen: Temporary failure in name resolution
sio@debian12:~$ sudo ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:kg1bh6vHML+PoULeuuh+RSKEDJ8cNqnnr+cySZwureA root@debian12
The key's randomart image is:
+---[RSA 3072]-----+
|+.+.
|..=0+
|o+ . o .
|. o . .* o
|+ o o* S
|= . .B
|. + = o. =
|+ *.* .o +
|.E=B++ . o..
+---[SHA256]-----+
sio@debian12:~$ sudo ssh-copy-id sio@192.168.66.129
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
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
/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.

sio@debian12:~$ sftp sio@192.168.66.129
sio@192.168.66.129's password:
sio@debian12:~$ sudo sftp sio@192.168.66.129
Connected to 192.168.66.129.
sftp>
```

Nous pouvons voir que la connexion se fait bien.

Maintenant sur le serveur sur lequel est installé WordPress, nous allons faire un script "backup.sh" pour automatiser la sauvegarde de la base de données et du site, de plus ce script enverra la sauvegarde sur notre deuxième serveur.

```
GNU nano 7.2
#!/bin/bash
#definition variable date
date=$(date +%Y)-$(date +%m)-$(date +%d)-$(date +%H)-$(date +%M)
#creation de l'archive
tar czvf /home/sio/backup-$date.tar.gz /var/www/html/wordpress/
# envoi sftp de l'archive
sftp sio@192.168.66.129 <<< '$put /home/sio/backup-'$date'.tar.gz'
#suppression de l'archive
rm -r /home/sio/backup-$date.tar.gz
#sql dump bdd wordpress
sudo mysqldump -u root wordpress > /home/sio/bddwordpress-$date.sql
#envoi bdd
sftp sio@192.168.66.129 <<< '$put /home/sio/bddwordpress-'$date'.sql'
#suppression backup bdd
rm -r /home/sio/bddwordpress-$date.sql
#log sauvegarde
echo "sauvegarde effectuée le $date" >> /home/sio/backup.log
```

Dans ce script nous utilisons une variable date pour connaître la date de notre sauvegarde, cela nous sera utile également pour le script de suppression des sauvegardes trop vieilles.

Regardons si sur notre deuxième serveur nous recevons bien les sauvegardes.

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
/usr/bin/xauth: file /home/sio/.Xauthority does not exist
sio@debian12:~$ ls
datawordpress-20241115.tar.gz
sio@debian12:~$ ls
datawordpress-20241115.tar.gz  wordpress-20241115.sql
sio@debian12:~$ █
```

C'est bon nous recevons bien les sauvegardes maintenant automatisons la sauvegarde et l'envoi car pour l'instant nous devons lancer notre script manuellement.

Pour l'automatiser nous allons aller dans la crontab qui permet de réaliser des tâches toutes les x minutes ou tous les 15 du mois par exemple. Dans notre cas pour le TP nous allons lancer le script toutes les minutes.

crontab -e

- * * * * * /home/sio/backup.sh

Grâce à cette ligne dans la crontab le script backup.sh s'exécutera toutes les minutes.

Il y a un problème comme nous allons recevoir des sauvegardes toutes les minutes, le dossier sur lequel nous recevons les sauvegardes va vite se remplir.

C'est pourquoi, nous allons faire un script qui supprime toutes les sauvegardes plus vieilles que 5 minutes

```
GNU nano 7.2
#!/bin/bash
find /home/sio/b* -type f -mmin +5 -exec rm {} \;
```

Et ce script, nous allons également le mettre dans la crontab mais du serveur de sauvegarde.

Ce script s'exécutera toutes les 5 minutes.

```
GNU nano 7.2
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
*/5 * * * * /home/sio/purge.sh
```

Et voila nous sauvegardons le site et la base de données de notre WordPress et cela de façons automatique.