

**BTS Services informatiques aux organisations - SISR****Session 2022****E4 – Support et mise à disposition de services informatiques****Coefficient 4****DESCRIPTION DE LA REALISATION PROFESSIONNELLE****NOM et prénom du candidat : LEDUC Quentin****N° candidat : 02145639104****Contexte de la réalisation professionnelle**

*Pour le développement de Scani qui grossi en terme de client, nous refont une refonte totale du site internet et du serveur pour améliorer notre communication externe envers nos coopérateur.*

**Intitulé de la réalisation professionnelle**

*Création d'un serveur WEB  
LAMP*

**Période de réalisation : 17 mai 2021****Lieu : Joigny****Modalité : Individuelle****Principale(s) activité(s) concernée(s) :**

- Déployer un service
- Développer son projet professionnel
- Participer à la valorisation de l'image de l'organisation sur les médias numériques en tenant compte du cadre juridique et des enjeux économiques
- Référencer les services en ligne de l'organisation et mesurer leur visibilité.

**Conditions de réalisation**

- **Ressources présentes**            **Nous avons un site internet**
- **Résultats attendus**            **Avoir un site internet fonctionnel**
- **Durée de réalisation**            **2h**

**Modalités d'accès à cette réalisation professionnelle.****Site internet : [www.netwaze.fr](http://www.netwaze.fr)****Aller dans « Réalisations Professionnelles » Mot de passe : Mr.Robot****Partie 1 – Procédure de mise en œuvre.****PRÉREQUIS MATÉRIEL**

Mettre à jour sa distribution : \$ sudo dnf update -y

Installation de Apache avec ces outils : \$ sudo dnf install httpd httpd-tools

Installation de MariaDB : \$ sudo dnf install mariadb-server mariadb -y

Le référentiel EPEL : \$ sudo dnf install <https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm>Le dépôt remi (php) : \$ sudo dnf install dnf-utils <http://rpms.remirepo.net/enterprise/remi-release-8.rpm>

# PRÉREQUIS

A la fin de notre installation de Centos par l'interface anaconda nous arrivons sur notre serveur (vide). Pour plus d'efficacité, je le prendrais en main avec le protocole SSH.

```
[kiu@centos8-ecole ~]$ cat /etc/redhat-release  
CentOS Stream release 8  
[kiu@centos8-ecole ~]$
```

Date du jour 17/05/21

Avant de commencer nous devons installer les prérequis. Je ne vais pas montrer l'ensemble des commandes dans des screenshots afin de pouvoir approfondir d'autres sujets.

Je tiens à préciser que les commandes utilisant **dnf** fonctionnent aussi avec un **yum**, celui-ci n'étant bientôt plus maintenu par RedHat. Aussi, jamais on ne se mettra directement en root.

Nous pouvons commencer !

# HTTPD / APACHE

La première chose à faire est d'activer et démarrer apache sur notre serveur car par défaut tout est inactif.

```
$ sudo systemctl enable httpd
```

```
$ sudo systemctl start httpd
```

Pour vérifier que notre serveur apache est bien actif on lance la commande : `$ systemctl status httpd`

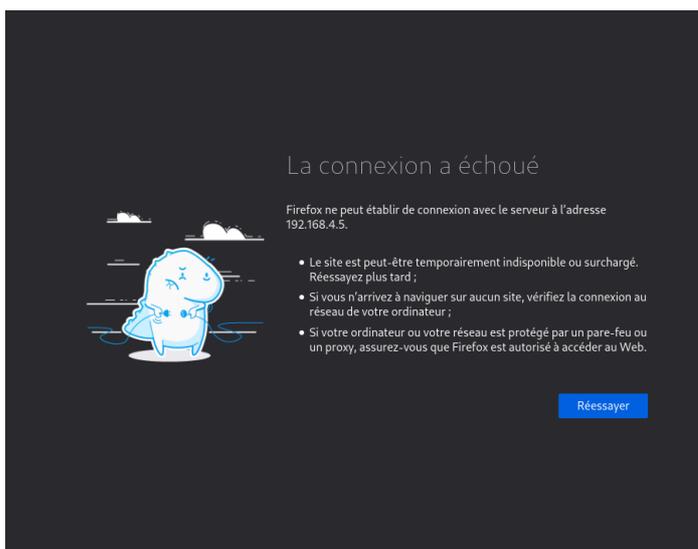
```
[kiu@centos8-ecole ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor prese
  Active: active (running) since Mon 2021-05-17 10:47:54 CEST; 49s ago
    Docs: man:httpd.service(8)
 Main PID: 49085 (httpd)
  Status: "Running, listening on: port 80"
   Tasks: 213 (limit: 4755)
  Memory: 29.2M
   CGroup: /system.slice/httpd.service
           └─49085 /usr/sbin/httpd -DFOREGROUND
             └─49086 /usr/sbin/httpd -DFOREGROUND
               └─49087 /usr/sbin/httpd -DFOREGROUND
                 └─49088 /usr/sbin/httpd -DFOREGROUND
                   └─49089 /usr/sbin/httpd -DFOREGROUND

mai 17 10:47:54 centos8-ecole systemd[1]: Starting The Apache HTTP Server...
mai 17 10:47:54 centos8-ecole httpd[49085]: AH00558: httpd: Could not reliably >
mai 17 10:47:54 centos8-ecole systemd[1]: Started The Apache HTTP Server.
mai 17 10:47:54 centos8-ecole httpd[49085]: Server configured, listening on: po>
[kiu@centos8-ecole ~]$
```

Pour finir, nous devons autoriser le trafic sur le port 80 et 443 (SSL), mais dans sa globalité nous allons directement autoriser le service.

En récupérant l'adresse IP de notre serveur et en se connectant avec sur un navigateur internet, nous pouvons voir que notre connexion est bloquée.

Commande pour récupérer notre adresse IP : `$ ip a`



Pour faire fonctionner le port 80 et le port 443 il faut le rentrer dans notre firewall-cmd.

```
$ sudo firewall-cmd --permanent --zone=public --add-service=http  
$ sudo firewall-cmd --permanent --zone=public --add-service=https
```

Pour que le firewall le prenne en compte, il faut le redémarrer

```
$ sudo firewall-cmd -reload
```

```
[kiu@centos8-ecole ~]$ sudo firewall-cmd --permanent --zone=public --add-service  
=http  
[sudo] Mot de passe de kiu :  
success  
[kiu@centos8-ecole ~]$ sudo firewall-cmd --permanent --zone=public --add-service  
=https  
success  
[kiu@centos8-ecole ~]$ sudo firewall-cmd --reload  
success  
[kiu@centos8-ecole ~]$
```

Si

Nos paramètres sont fonctionnels grâce au « success » que l'on peut lire.

nous rafraîchissons notre page internet avec notre adresse IP, nous pouvons constater que le trafic passe et que nous sommes dirigés vers la page HTML d'apache.



## Test Page

This page is used to test the proper operation of the [Apache HTTP server](#) after it has been installed. If you can read this page it means that this site is working properly. This server is powered by [CentOS](#).

### Just visiting?

The website you just visited is either experiencing problems or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting [www.example.com](#), you should send e-mail to "[webmaster@example.com](#)".

# MARIADB / MYSQL

De la même manière que pour Apache, nous devons activer et démarrer MariaDB.

```
$ sudo systemctl enable mariadb
```

```
$ sudo systemctl start mariadb
```

Pour savoir si mariadb est démarré : `sudo systemctl status mariadb`

Maintenant, nous devons sécuriser notre logiciel, nous pourrions appeler cela une post-install

```
$ mysql_secure_installation
```

Dans «current password for root» : il suffit d'appuyer sur entrer car ceci est une nouvelle installation.

Pour le reste des questions, nous mettons Y a chaque fois. (il faut mettre un mot de passe sûr et complexe).

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

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

Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

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!
[kiu@centos8-ecole ~]$
```

# PHP

Nous pouvons dès maintenant passer à l'installation de PHP. Nous reviendrons à Mariadb par la suite.

Tout d'abord, nous devons rechercher les versions de PHP disponibles dans les dépôts

```
$ sudo dnf module list php
```

```
[kiu@centos8-ecole ~]$ sudo dnf module list php
Dernière vérification de l'expiration des métadonnées effectuée il y a 0:01:15 le
lun. 17 mai 2021 11:19:01 CEST.
CentOS Stream 8 - AppStream
Name      Stream      Profiles      Summary
php       7.2 [d]     common [d], devel, minimal  PHP scripting language
php       7.3         common [d], devel, minimal  PHP scripting language
php       7.4         common [d], devel, minimal  PHP scripting language

Remi's Modular repository for Enterprise Linux 8 - x86_64
Name      Stream      Profiles      Summary
php       remi-7.2    common [d], devel, minimal  PHP scripting language
php       remi-7.3    common [d], devel, minimal  PHP scripting language
php       remi-7.4    common [d], devel, minimal  PHP scripting language
php       remi-8.0    common [d], devel, minimal  PHP scripting language

Aide : [d]éfaut, [e]activé, [x]désactivé, [i]nstallé
[kiu@centos8-ecole ~]$
```

Nous voyons que nous avons le dépôt par défaut de Centos, mais nous avons installer le dépôt remi. Pourquoi ? Le dépôt remi est mis à jour plus souvent que les dépôts officiels, d'ailleurs nous pouvons voir que PHP 8.0 est sorti seulement sur remi.

Pour l'exemple nous allons installer le PHP 7.4 de remi

Nous devons d'abord reset le module PHP

```
$ sudo dnf module reset php
```

Puis nous activons le module de remi

```
$ sudo dnf module enable php:remi-7.4
```

Nous pouvons maintenant installer php

```
$ sudo dnf install php php-opcache php-gd php-curl php-mysqlnd
```

On vérifie la version de notre php : `php -v`

```
[kiu@centos8-ecole ~]$ php -v
PHP 7.4.19 (cli) (built: May 4 2021 11:06:37) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
with Zend OPcache v7.4.19, Copyright (c), by Zend Technologies
[kiu@centos8-ecole ~]$
```

Pour que Wordpress fonctionne bien il lui faut un composant qui est PHP-fpm. On l'active et on le démarre.

```
$ sudo systemctl start php-fpm
```

```
$ sudo systemctl enable php-fpm
```

puis on vérifie son status : `sudo systemctl status php-fpm`

Pour que apache exécute du code php via PHP-fpm il faut lui faire comprendre.

```
$ sudo setsebool -P httpd_execmem 1
```

Pour que cette configuration fonctionne nous devons redémarrer apache.

```
$ sudo systemctl restart httpd
```

```
[kiu@centos8-ecole ~]$ sudo setsebool -P httpd_execmem 1
[kiu@centos8-ecole ~]$ sudo systemctl restart httpd
[kiu@centos8-ecole ~]$
```

Pour tester notre PHP, vérifier si il fonctionne parfaitement bien sur notre système, nous allons créer une simple page php

```
$ sudo nano /var/www/html/
info.php
```

On entre ce code :

```
<?php
phpinfo();
?>
```

Puis nous pouvons sauvegarder notre .php.

On peut aller sur notre adresse ip et rajouter /info.php a la fin de l'URL.

php	
<b>System</b>	Linux centos8-ecole 4.18.0-294.el8.x86_64 #1 SMP Mon Mar 15 22:38:42 UTC 2021 x86_64
<b>Build Date</b>	May 4 2021 11:06:37
<b>Build System</b>	Red Hat Enterprise Linux release 8.3 (Ootpa)
<b>Build Provider</b>	Remi's RPM repository <https://rpms.remirepo.net/>
<b>Server API</b>	FPM/FastCGI
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc
<b>Loaded Configuration File</b>	/etc/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php.d
<b>Additional .ini files parsed</b>	/etc/php.d/10-opcache.ini, /etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gd.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-json.ini, /etc/php.d/20-mbstring.ini, /etc/php.d/20-mysqli.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-simplexml.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sodium.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-xml.ini, /etc/php.d/20-xmlwriter.ini, /etc/php.d/20-xsl.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini, /etc/php.d/30-xmireader.ini
<b>PHP API</b>	20190902
<b>PHP Extension</b>	20190902
<b>Zend Extension</b>	320190902
<b>Zend Extension Build</b>	API320190902.NTS
<b>PHP Extension Build</b>	API20190902.NTS
<b>Debug Build</b>	no
<b>Thread Safety</b>	disabled
<b>Zend Signal Handling</b>	enabled
<b>Zend Memory Manager</b>	enabled
<b>Zend Multibyte Support</b>	provided by mbstring
<b>IPv6 Support</b>	enabled
<b>DTrace Support</b>	available, disabled
<b>Registered PHP Streams</b>	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar
<b>Registered Stream Socket Transports</b>	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
<b>Registered Stream Filters</b>	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, bzip2.*, convert.iconv.*

Notre php fonctionne bien. Pour des raisons de sécurité il vaut mieux supprimer le fichier info.php

```
$ rm /var/www/html/info.php
```

# ADMINER

Pour gérer notre base de données, nous avons 2 choix possibles. Soit passer par un logiciel dédié, PhpMyAdmin ou Adminer qui sont fortement utilisés chez les hébergeurs «habituels» ou bien le faire manuellement.

Personnellement je ne suis pas favorable à PhpMyAdmin car il utilise beaucoup trop de ressources physiques à cause des ajax dans tout les sens. Il est aussi une cible de choix pour les pirates informatiques. Donc je vais installer Adminer.

Pour installer Adminer on va devoir le télécharger et le mettre dans /usr/share/adminer

```
$ sudo wget https://github.com/vrana/adminer/releases/download/v4.8.1/adminer-4.8.1.php -P /usr/share/adminer
```

Une fois téléchargé, nous allons lui créer un alias, qui sera plus simple pour faire des mises à jour.

```
$ sudo ln -s /usr/share/adminer/adminer-4.8.1.php /usr/share/adminer/adminer.php
```

Passons à la configuration de Adminer. On crée un nouveau fichier adminer.conf

```
$ sudo nano /etc/httpd/conf.d/adminer.conf
```

et insérons la configuration par défaut :

```
Alias /adminer /usr/share/adminer/adminer.php
```

```
<Directory /usr/share/adminer>
```

```
Require all granted
```

```
</Directory>
```

The screenshot shows the Adminer 4.8.1 authentication page. At the top left, there is a language dropdown menu set to 'Français'. Below it, the version 'Adminer 4.8.1' is displayed. The main content area is titled 'Authentification' and contains a form with the following fields: 'Système' (MySQL), 'Serveur' (localhost), 'Utilisateur' (root), 'Mot de passe' (masked with dots), and 'Base de données'. At the bottom of the form, there are two checkboxes: 'Authentication' (checked) and 'Authentication permanente' (unchecked).

Pour se connecter, il suffit d'entrer notre utilisateur root et notre mot de passe.

On recharge alors notre apache pour qu'il prenne en compte notre nouvelle configuration

```
$ sudo systemctl reload httpd
```

Afin d'éviter toute erreur de notre SELinux nous devons autoriser Adminer

```
$ sudo restorecon -R /usr/share/adminer/
```

Nous pouvons nous connecter à Adminer en rajoutant /adminer dans notre URL.

Nous pouvons maintenant créer une base de donnée pour installer Wordpress.

Langue: Français MySQL » Serveur Déconnexion

**Adminer 4.8.1**

DB:  

[Requête SQL](#) [Importer](#)  
[Exporter](#)

Sélectionner la base de données

[Créer une base de données](#) [Privilèges](#) [Liste des processus](#) [Variables](#) [Statut](#)

Version de MySQL : **5.5.5-10.3.27-MariaDB** via l'extension PHP **MySQLi**

Authentifié en tant que : **root@localhost**

	Base de données - <a href="#">Rafraîchir</a>	Interclassement	Tables	Taille - <a href="#">Calcul</a>
<input type="checkbox"/>	<a href="#">information_schema</a>	utf8_general_ci	?	?
<input type="checkbox"/>	<a href="#">mysql</a>	latin1_swedish_ci	?	?
<input type="checkbox"/>	<a href="#">performance_schema</a>	utf8_general_ci	?	?

Sélectionnée(s) (0)

Supprimer

## INSTALLATION DE WORDPRESS

Notre Wordpress sera sur beta.scani.fr Nous allons télécharger Wordpress dans notre répertoire

/tmp

\$ cd /tmp

\$ wget <https://wordpress.org/latest.tar.gz>

```
[kiu@centos8-ecole tmp]$ wget https://wordpress.org/latest.tar.gz
--2021-05-17 13:30:35-- https://wordpress.org/latest.tar.gz
Résolution de wordpress.org (wordpress.org)... 198.143.164.252
Connexion à wordpress.org (wordpress.org)[198.143.164.252]:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 15750424 (15M) [application/octet-stream]
Sauvegarde en : « latest.tar.gz »

latest.tar.gz 13%[====>] 1,98M 113KB/s tps 90s
```

Nous décompressons notre Wordpress toujours dans notre dossier /tmp

\$ tar xzvf latest.tar.gz

Pour aller plus vite, on va directement créer le fichier de configuration de Wordpress

\$ cp /tmp/wordpress/wp-config-sample.php /tmp/wordpress/wp-config.php

Puis nous copions tout le Wordpress pour le mettre dans /var/www/Scani

\$ sudo cp -a /tmp/wordpress/. /var/www/Scani

On accorde l'accès à apache

```
$ sudo chown -R $USER:$USER /var/www/Scani
```

On créer un dossier de log

```
$ sudo mkdir -p /var/www/Scani/log
```

Nous devons créer les dossiers dont apache à besoin pour nos hôtes virtuels

```
$ sudo mkdir /etc/httpd/sites-available
```

```
$ sudo mkdir /etc/httpd/sites-enabled
```

Nous modifions le fichier de configuration d'apache

```
$ sudo nano /etc/httpd/conf/httpd.conf
```

Puis nous lui indiquons où se situe nos hôtes virtuels a la toute fin de notre fichier de configuration

>

```
IncludeOptional sites-enabled/*.conf
```

On créer notre hôte virtuel

```
$ sudo nano /etc/httpd/sites-available/Scani.fr.conf et on lui insère la configuration
```

>

```
<VirtualHost *:80>
```

```
    ServerName www.beta.scani.fr
```

```
    ServerAlias beta.scani.fr
```

```
    DocumentRoot /var/www/Scani
```

```
    ErrorLog /var/www/Scani/log/error.log
```

```
    CustomLog /var/www/Scani/log/requests.log combined
```

```
</VirtualHost>
```

On active notre hôte virtuel en créant un lien symbolique

```
$ sudo ln -s /etc/httpd/sites-available/Scani.fr.conf /etc/httpd/sites-enabled/Scani.fr.conf
```

Afin d'éviter que SELinux pose des problèmes pour les log (car nous avons modifié le dossier) nous lui indiquons d'accepter cette configuration

```
$ sudo ls -dlZ /var/www/Scani/log
```

```
[kiu@centos8-ecole usr]$ sudo ls -dlZ /var/www/Scani/log/
drwxr-xr-x. 2 kiu kiu unconfined_u:object_r:user_tmp_t:s0 6 17 mai 14:02 /var/www/Scani/log/
[kiu@centos8-ecole usr]$
```

Nous devons avoir cette sortie

Maintenant, il faut signifier à apache d'écrire les log dans ce répertoire

```
$ sudo semanage fcontext -a -t httpd_log_t "/var/www/Scani/log(/.*)?"
```

Et nous rendons persistante cette configuration

```
$ sudo restorecon -R -v /var/www/Scani/log
```

Puis l'on confirme les modifications

```
$ sudo ls -dlZ /var/www/Scani/log/
```

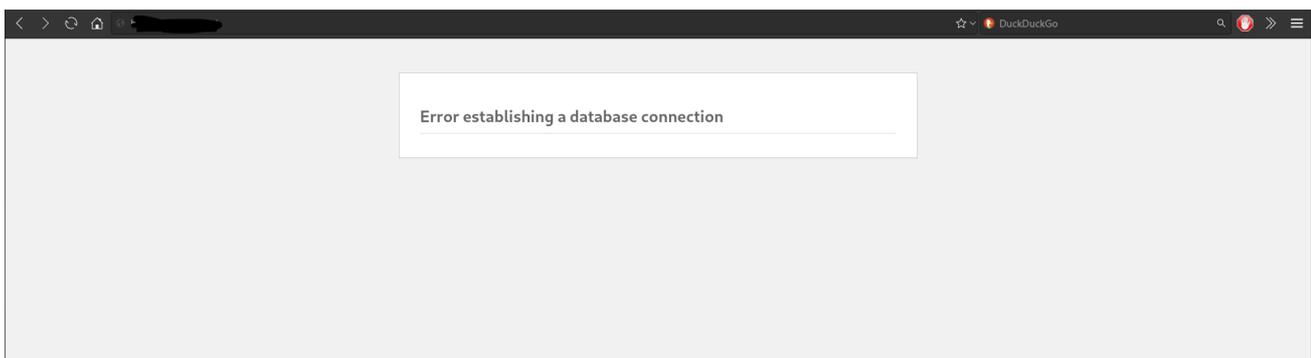
```
[kiu@centos8-ecole usr]$ sudo restorecon -R -v /var/www/...../log
[kiu@centos8-ecole usr]$ sudo ls -dlZ /var/www/...../log/
drwxr-xr-x. 2 kiu kiu unconfined_u:object_r:httpd_log_t:s0 6 17 mai 14:02 /var/www/...../log/
[kiu@centos8-ecole usr]$
```

On redémarre ensuite notre serveur apache

```
$ sudo systemctl restart httpd
```

Maintenant nous pouvons ouvrir les ports de notre box internet et faire pointer notre nom de domaine sur notre IP publique par notre registrar. (je ne vais pas montrer mes informations personnelles, mais j'ai ouvert ma box sur le port 80 pour le moment)

Une fois la configuration réalisée nous pouvons désormais mettre notre domaine : beta.scani.fr et on arrive sur Wordpress, avec une erreur de base de donnée



Cette erreur est normal car nous avons déjà un config.php mais il est vide car nous n'avons rien rentrer un peu avant.

On va donc le modifier et rentrer les informations nécessaires.

```
$ sudo nano /var/www/netwaze.xyz/wp-config.php
```

```

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'root' );

/** MySQL database password */
define( 'DB_PASSWORD', 'toor89toor' );

/** 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', '' );

/**#@+

```

Les mots de passe sont faibles mais comme ceci est un exemple, l'on ne va pas en tenir compte et je m'arrête ici car les autres informations comme changer le wp\_ comme prefix n'est pas réellement utile.

Puis l'on recharge notre page internet et nous devons arriver sur la page de configuration de Wordpress.

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username   
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password    
Strong  
**Important:** You will need this password to log in. Please store it in a secure location.

Your Email   
Double-check your email address before continuing.

Search engine visibility  Discourage search engines from indexing this site  
It is up to search engines to honor this request.

Nous allons installer un certificat SSL. Pour le moment donc, je ne fait pas la configuration de Wordpress.

# INSTALLER UN CERTIFICAT SSL AVEC CERTBOT

Pour installer un certificat SSL nous allons installer un outil fort sympathique : Certbot

```
$ sudo dnf install mod_ssl openssl certbot python3-certbot-apache
```

Maintenant il faut dire à apache que notre site prend désormais le port 443.

Pour ce faire nous ouvrons notre fichier de conf de beta.scani.fr

```
$ sudo nano /etc/httpd/sites-available/scani.fr.conf
```

On rajoute notre configuration juste en dessous de notre conf actuelle

>

```
<VirtualHost *:443>
```

```
    ServerName beta.scani.fr
```

```
    ServerAlias www.beta.scani.fr
```

```
    DocumentRoot /var/www/Scani
```

```
<Directory /var/www/Scani>
```

```
    Options -Indexes +FollowSymLinks
```

```
    AllowOverride All
```

```
</Directory>
```

```
ErrorLog /var/log/httpd/www.beta.scani.fr-error.log
```

```
CustomLog /var/log/httpd/www.beta.scani.fr-access.log combined
```

```
</VirtualHost>
```

Puis on redémarre apache pour prendre en compte notre configuration

```
$ sudo systemctl restart httpd
```

Maintenant nous lançon certbot pour y installer notre certificat SSL et mettons notre configuration.

```
$ sudo certbot --apache -d beta.scani.fr
```

```

[kiu@centos8-ecole local]$ sudo certbot --apache -d [REDACTED]
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator apache, Installer apache
Enter email address (used for urgent renewal and security notices)
(Enter 'c' to cancel): [REDACTED]

-----
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. Do you agree?
-----
(Y)es/(N)o: y

-----
Would you be willing, once your first certificate is successfully issued, 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
Account registered.
Requesting a certificate for netwaze.xyz
Performing the following challenges:
http-01 challenge for netwaze.xyz
Waiting for verification...
Cleaning up challenges
Deploying Certificate to VirtualHost /etc/httpd/sites-enabled/[REDACTED]
Redirecting vhost in /etc/httpd/sites-enabled/netwaze.xyz.conf to ssl vhost in /etc/httpd/sites-enabled/[REDACTED]

-----
Congratulations! You have successfully enabled https://[REDACTED]
-----
Subscribe to the EFF mailing list (email: [REDACTED]).

IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/[REDACTED]/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/[REDACTED]/privkey.pem
  Your certificate will expire on 2021-08-15. 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"
- 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

[kiu@centos8-ecole local]$

```

Nous devons retourner dans notre fichier de configuration Scani.fr pour lui appliquer notre config SSL.

```
$ sudo nano /etc/httpd/sites-available/Scani.fr.conf
```

On rajoute : **SSLEngine On**

```

<VirtualHost *:443>
  ServerName beta.scani.fr
  ServerAlias www.beta.scani.fr
  DocumentRoot /var/www/Scani
<Directory /var/www/beta.scani.fr>
  Options -Indexes +FollowSymLinks

```

AllowOverride All

</Directory>

ErrorLog /var/log/httpd/www.beta.scani.fr-error.log

CustomLog /var/log/httpd/www.netwaze.xyz-access.log combined

SSLEngine On

SSLCertificateFile /etc/letsencrypt/live/beta.scani.fr/fullchain.pem

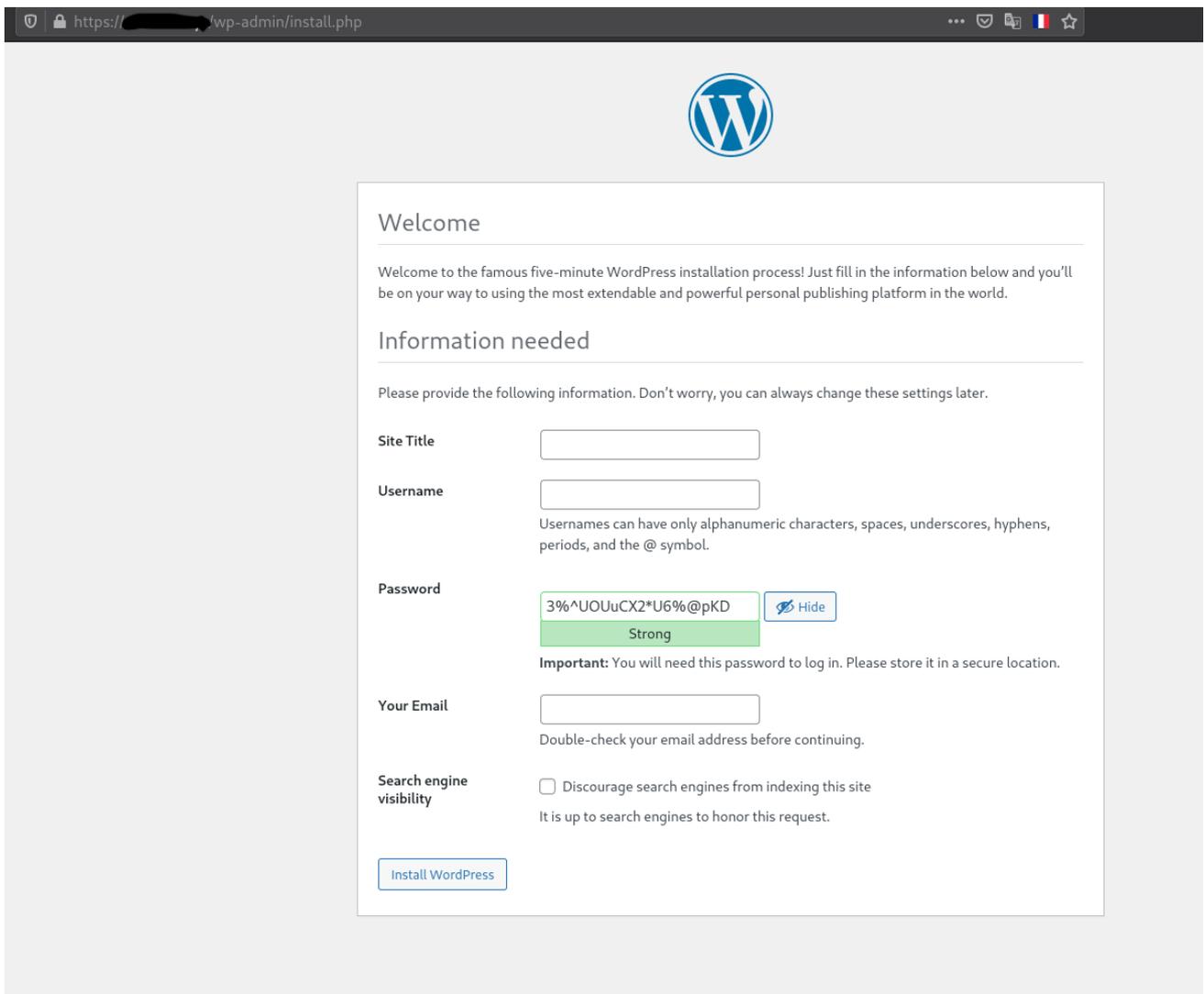
SSLCertificateKeyFile /etc/letsencrypt/live/beta.scani.fr/privkey.pem

Include /etc/letsencrypt/options-ssl-apache.conf

On redémarre notre serveur apache

\$ sudo systemctl restart httpd

On ouvre notre port 443 sur notre box et nous pouvons relancer notre page internet et nous constatons que notre http est passer automatiquement en https.



# RENOUVELER LE CERTIFICAT SSL

The image shows a browser's security tab for the website netwaze.xyz. The tab is titled 'Certificat' and displays the following information:

- netwaze.xyz** (R3) issued by **DST Root CA X3**
- Nom du sujet**: Nom courant [redacted]
- Nom de l'émetteur**: Pays US, Organisation Let's Encrypt, Nom courant R3
- Validité**: Pas avant Mon, 17 May 2021 13:00:29 GMT, Pas après Sun, 15 Aug 2021 13:00:29 GMT
- Noms alternatifs du sujet**: Nom DNS [redacted]
- Informations sur la clé publique**: Algorithme RSA, Taille de la clé 2048, Exposant 65537, Module D7:A1:9D:B8:D7:F3:B6:CD:4C:ED:38:ED:DB:6F:55:4C:90:DB:E6:A3:BC:15:44:...

Sachant que le script n'est valable que 90 jours nous allons lancer une commande automatique tout les jours et 2 fois par jour (certbot le recommande).

Pour se faire nous allons mettre une tâche cron

```
$ sudo nano crontab -e
```

et on y rentre

```
echo "0 0,12 * * * root python3 -c 'import random; import time;
```

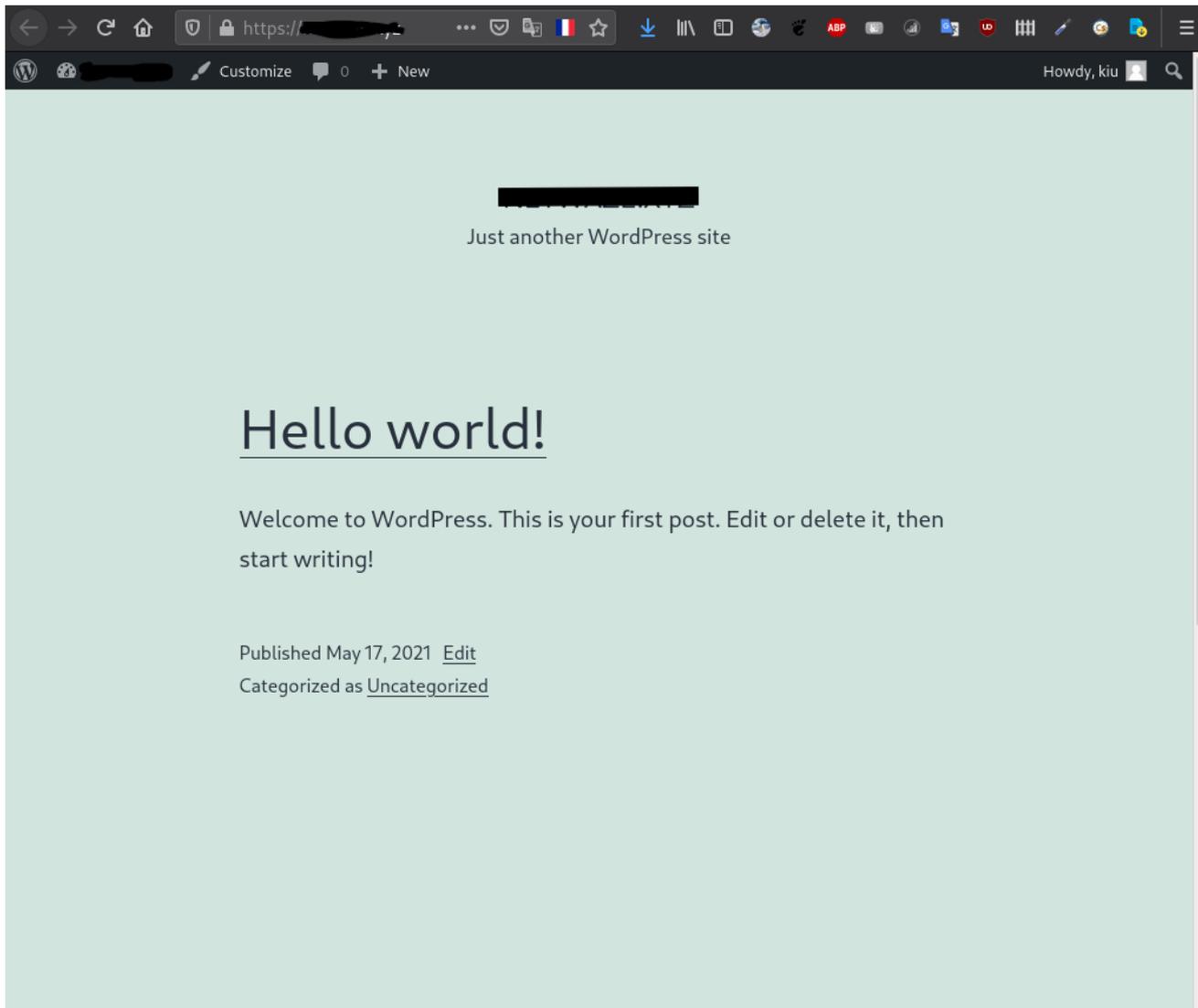
```
time.sleep(random.random() * 3600)' && certbot renew -q" | sudo tee -a /etc/crontab > /dev/null
```

```
echo "0 0,12 * * * root python3 -c 'import random; import time; time.sleep(random.random() * 3600)' && certbot renew -q" | sudo tee -a /etc/crontab > /dev/null
```

Ceci aura pour effet de demander 2 fois par jour, a minuit et a midi un certificat. Ceci évite de faire réduire la charge de let's encrypt. 2 fois par jour peut sembler beaucoup mais ceci évite d'avoir une mauvaise surprise comme une révocation de let's encrypt.

## CONFIGURATION DE WORDPRESS

Nous pouvons maintenant configurer notre Wordpress et y accéder.



Seulement notre langue actuelle est l'anglais, on ne peut pas télécharger de thème ou de plugin depuis wordpress.com. Ceci peut être gênant pour certaine personne. Mais nous allons remédier à ce problème.

Add Plugins [Upload Plugin](#) Help ▾

Featured Popular Recommended Favorites

Plugins extend and expand the functionality of WordPress. You may automatically install plugins from the [WordPress Plugin Directory](#) or upload a plugin in .zip format by clicking the button at the top of this page.

An unexpected error occurred. Something may be wrong with WordPress.org or this server's configuration. If you continue to have problems, please try the [support forums](#).

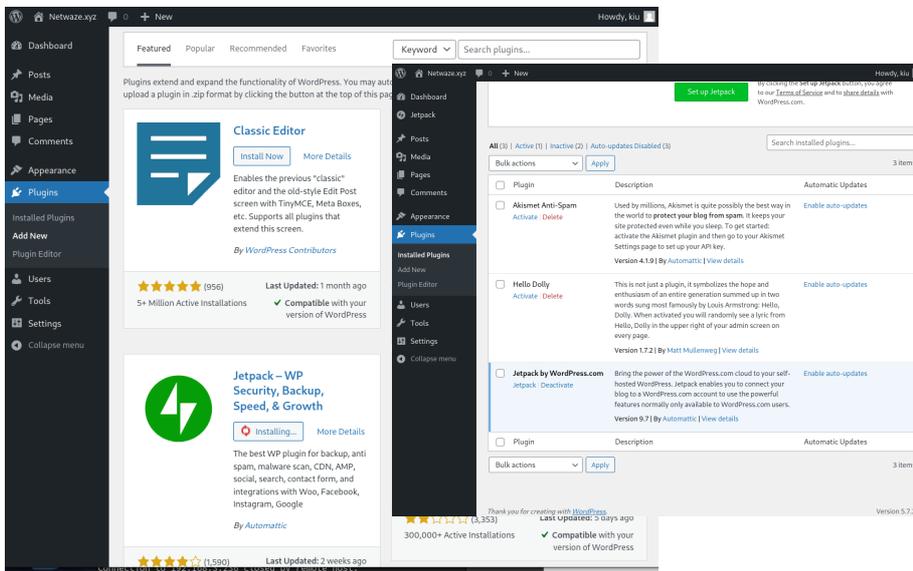
[Try Again](#)

**Popular tags**

You may also browse based on the most popular tags in the Plugin Directory:

Pour se faire nous devons dire à SELinux d'être un peu plus agréable avec notre Wordpress\$  
sudo nano /etc/selinux/config  
Puis nous changeont SELINUX=enforcing en SELINUX=permissive

```
GNU nano 2.9.8 /etc/selir
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```



Puis on redémarre notre serveur

\$ sudo shutdown -r now

Une fois redémarrer nous pouvons rafraichir notre page internet, on peut constater que les plugins sont apparus et que nous pouvons les télécharger et activer.

## GOOGLE ANALYTICS

On veut maintenant mesurer l'audience du site internet, pour ce faire sous wordpress nous allons utiliser Google Analytics.

On va télécharger le plugin GA Google Analytics et on va l'installer.

1036 elements
Sur 29



### GA Google Analytics

Ajoute votre code de suivi Google Analytics à votre site WordPress.

*Par Jeff Starr*

Actif

[Plus de détails](#)

★★★★★ (140)

600 000+ installations actives

Dernière mise à jour : il y a 3 mois

✓ Compatible avec votre version de WordPress

Dans le plugin, Réglages - Google Analytics on va dans l'onglet Réglages de l'extension et on met notre ID de Suivi GA et on laisse la suite par défaut.

## Réglages de l'extension

ID de suivi GA

UA-17

Méthode de Suivi

- Statistiques universelles / `analytics.js` (par défaut)
- Balise globale du site / `gtag.js` (nouvelle méthode)
- Héritage / `ga.js` (obsolète)

Affichage publicitaire

Activer la prise en charge pour [Affichage publicitaire](#)

Attribution de lien

Activer la prise en charge pour [Améliorer les liens d'attributions](#)

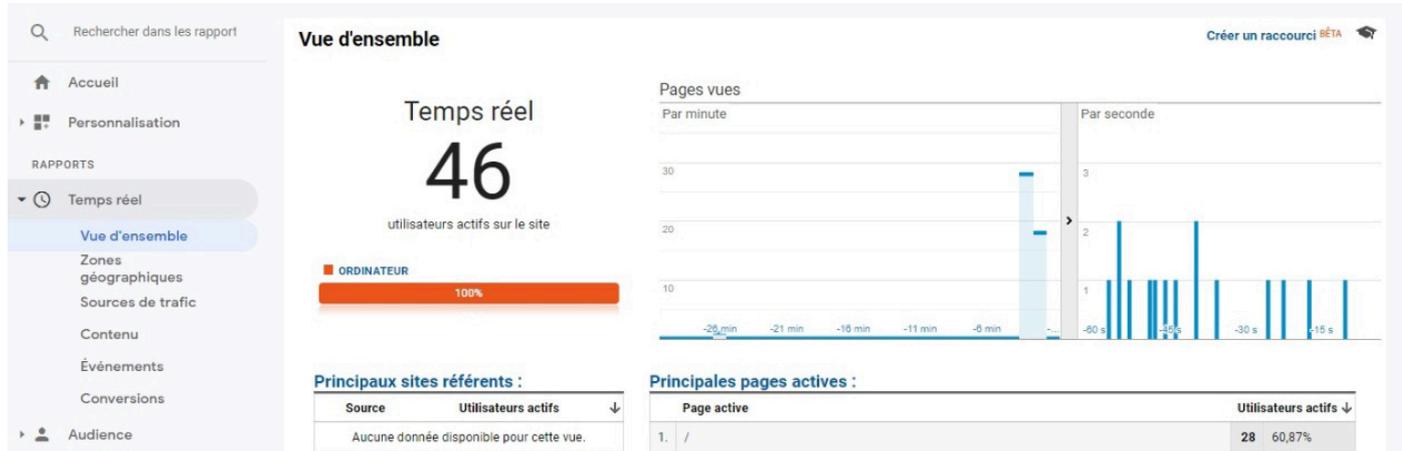
Anonymisation IP

Activer la prise en charge pour [Anonymisation IP](#)

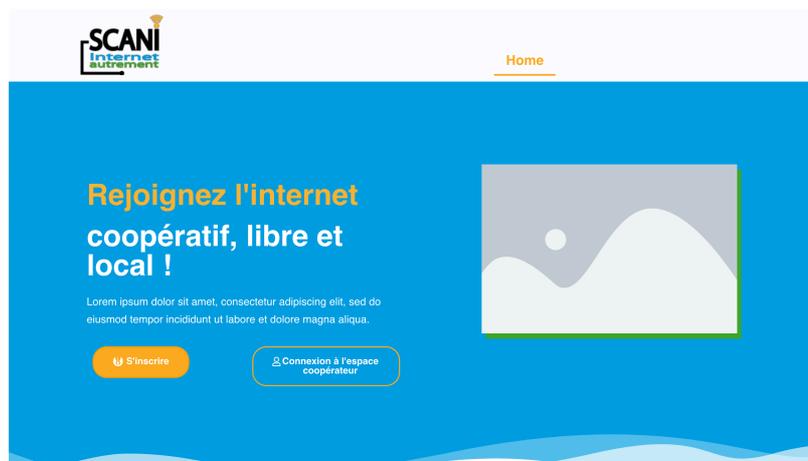
Forcer le SSL

Activer la prise en charge pour [Forcer le SSL](#)

En demandant à tout Scani de ce connecter sur le site on arrive à voir en temps réel le nombre de personne qui sont actuellement connecté à notre site internet depuis google analytics.



## Partie 2 – Validation.



Après avoir customiser notre Wordpress, nous pouvons alors voir notre site internet et trouver nos informations.  
(Site Toujours en Construction)

## Partie 3 – Veille technologique.

On a fait notre site internet avec Apache mais on aurait pu le faire avec NGINX ou IIS. Pour le CMS utiliser on aurait pu utiliser Joomla ou Drupal mais on aurait pu alors coder notre site entièrement avec des technologies HTML, PHP et CSS.