

Déploiement d'une application Symfony

Mathis BOURRELLY 30/03/2022

Table des matières

1.	Nom de domaine	1
2.	Installation et configuration de WAMP	2
3.	Redirection du DNS	2
4.	Redirection des ports	3
5.	Génération du certificat SSL	3
6.	Configuration du serveur apache	4
7.	Redirection du port SSL	5
8.	Activation du pare-feu	6
9.	Configuration de Symfony	6
10.	Script d'automatisation	6
11.	Déploiement continue	7
1.	Serveur FTP	7
2.	CI/CD gitlab	8
12.	Accès distant a la base de données	9
13.	Condition de déploiement	10

1. Nom de domaine

La première étape est de réserver un nom de domaine pour l'accès a l'application web de GSB. Pour cela il existe des « Registrar » (bureau d'enregistrement de nom de domaine) et notamment le site amen.fr qui fournit gratuitement des nom de domaine de type « .fr » pendant



1 an. Pour notre projet GSB nous avons choisi le nom de domaine pharma-gsb.fr . Celui-ci est associé à un DNS qui va se charger de rediriger les utilisateurs qui se rendent sur ce nom de domaine, vers notre application web.

2. Installation et configuration de WAMP

Une fois le nom de domaine enregistré, nous procédons à l'installation de la pile logiciel WAMP sur notre serveur, car nous travaillerons sur un serveur fonctionnant sous Windows 10 pro.

Dans les configurations de WAMP, il convient d'ajouter un Virtual Host afin de séparer les différents serveurs web qui pourraient être hébergés sur notre machine et qui permet d'attribuer une adresse web, l'adresse par défaut étant « localhost ».

Ajouter un VirtualHost - Retour à l'accueil

• Version 3.2.3 - 64bit french

Apache Virtual Hosts `c:/wamp64/bin/apache/apache2.4.46/conf/extra/httpd-vhosts.conf`

VirtualHost déjà définis :

ServerName : localhost - Directory : c:/wamp64/www
ServerName : pharma-gsb.fr - Directory : d:/gsb Formulaire de suppression de VirtualHost

Windows hosts `C:/windows/system32/drivers/etc/hosts`

Nom du **Virtual Host** Pas d'espace - Pas de tiret bas (_) **Requis**

Chemin complet absolu du **dossier** VirtualHost - Exemples : C:/wamp/www/projet/ ou E:/www/site1/ **Requis**

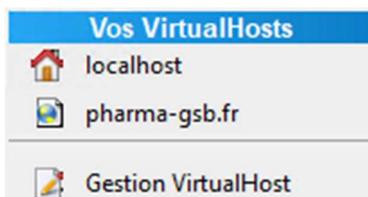
Si vous voulez utiliser un "Listen port" autre que celui par défaut, vous devez ajouter un Listen Port à Apache par Clic-Droit

Outils **Optionnel**

Si vous voulez utiliser les VirtualHost par IP : **IP locale** 127.x.y.z **Optionnel**

Démarrer la création du VirtualHost (Peut prendre un certain temps)

Pour cela on renseigne le nom du Virtual Host, ici notre nom de domaine, puis le dossier qui contient les fichiers du site web. Ensuite, il suffit de faire un redémarrage DNS grâce aux outils de WAMP.



3. Redirection du DNS

La troisième étape consiste à faire la redirection de notre nom de domaine vers notre serveur :

Nom	TTL	Type	Valeur
pharma-gsb.fr	900	A	82.65.254.14
Type / Valeur par défaut			

Pour cela, nous devons modifier la zone DNS pharma-gsb.fr de type A qui permet de relier une adresse IP à un nom de domaine. Ici la valeur de l'IP est celle de notre routeur internet fournie par le FAI Free. Cette box internet permet une redirection des ports que nous configurerons dans les prochaines étapes.

4. Redirection des ports

Pour pouvoir accéder à notre serveur depuis l'IP de la box, nous devons faire une redirection de port grâce à l'application web de celle-ci :

Redirection de port ✕

IP Destination : UTILISA-7J5B2FP (192.168.1.172) ⓘ

Redirection active :

IP source : Toutes

Protocole : TCP ⓘ

Port de début : 80 ⓘ

Port de fin : 80 ⓘ

Port de destination : 80 ⓘ

Commentaire : http

Ici nous inscrivons en destination l'IP local de notre serveur, toutes les IP pour la source, TCP pour le protocole pour assurer un bon transfert et 80 pour le port de destination.

5. Génération du certificat SSL

Pour pouvoir accéder aux serveurs via le protocole sécurisé https, nous devons générer un certificat SSL. Pour cela nous allons utiliser le programme appelé certbot. Celui-ci nous fournit une liste d'instructions à réaliser :

- Télécharger et installer le programme certbot
- Lancer la commande `certbot certonly --standalone -d pharma-gsb.fr -d www.pharma-gsb.fr` en vérifiant que le chemin pour la commande certbot soit valide et que le serveur wamp soit éteint

Puis nous devons générer un certificat SSL pour le serveur lui-même en utilisant openssl que l'on peut télécharger sur le site officiel, puis lancer ces commandes :

- `CD C:\Program Files\OpenSSL-Win64\bin`
- `mkdir website`
- `openssl genrsa -out website\server.key 2048`

- `openssl req -new -key website\server.key -out website\server.csr`
- cette commande nécessite un mot de passe robuste pour pouvoir générer la clé
- Puis répondre au questionnaire qui suit

```
Example question and answers:
Country Name (2 letter code) [AU]:GB
State or Province Name (full name) [Some-State]: Hampshire
Locality Name (eg, city) []: Portsmouth
Organization Name (eg, company) [Internet Widgits Pty Ltd]: Wamp Helpers Ltd
Organizational Unit Name (eg, section) []: Information Technology
Common Name (e.g. server FQDN or YOUR name) []: wamphelpers.dev
Email Address []: me@wamphelpers.dev
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: ( leave blank just hit the enter key )
An optional company name []: ( leave blank just hit the enter key )
```

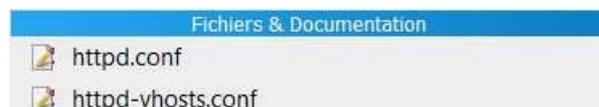
- `openssl x509 -req -days 365 -in website\server.csr -signkey website\server.key -out website\server.crt`
- Puis on déplace les fichier créer dans le dossier du serveur apache :

`C:\wamp64\bin\apache\apache2.4.46\conf`

Nom	Modifié le	Type	Taille
extra	23/10/2021 13:34	Dossier de fichiers	
original	23/10/2021 14:22	Dossier de fichiers	
charset.conv	02/08/2020 10:25	Fichier CONV	2 Ko
httpd.conf	09/01/2022 00:12	Fichier CONF	22 Ko
magic	02/08/2020 10:25	Fichier	14 Ko
mime.types	02/08/2020 10:25	Fichier TYPES	62 Ko
openssl.cnf	21/04/2020 14:22	Fichier CNF	11 Ko
server.crt	08/01/2022 20:54	Certificat de sécuri...	2 Ko
server.csr	08/01/2022 20:53	Fichier CSR	2 Ko
server.key	08/01/2022 19:37	Fichier KEY	2 Ko

6. Configuration du serveur apache

Pour configurer le serveur apache, il faut modifier le fichier `httpd.conf` :



Il faut décommenter les lignes :

```
LoadModule ssl_module modules/mod_ssl.so
Include conf/extra/httpd-vhosts.conf
Include conf/extra/httpd-ssl.conf
LoadModule ssl_module modules/mod_ssl.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
```

Ces lignes nous servent à activer les modules SSL du serveur apache.

Puis il nous faut configurer le fichier `httpd-vhost.conf` dans lequel on ajoute ce code :

```
<VirtualHost *:80>
    ServerName pharma-gsb.fr
    ServerAlias www.pharma-gsb.fr
    DocumentRoot "d:/gsb/public"
    <Directory "d:/gsb/public/">
        Options +Indexes +Includes +FollowSymLinks +MultiViews
        AllowOverride All
        Require all granted
    </Directory>
# Redirect to the HTTPS site
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^/?(.*)$ https://www.pharma-gsb.fr/$1 [NE,L,R=301]
</VirtualHost>

<VirtualHost *:443>
    ServerName pharma-gsb.fr
    ServerAlias www.pharma-gsb.fr
    DocumentRoot "d:/gsb/public"
    <Directory "d:/gsb/public/">
        Options -Indexes +Includes +FollowSymLinks +MultiViews
        AllowOverride All
        Require all granted
    </Directory>
RewriteEngine On
# Redirect to the correct domain name
RewriteCond %{HTTP_HOST} !^www.pharma-gsb.fr$ [NC]
RewriteRule ^/?(.*)$ https://www.pharma-gsb.fr/$1 [NE,L,R=301]
SSLEngine on
SSLCertificateFile C:\Certbot\live\www.pharma-gsb.fr\fullchain.pem
SSLCertificateKeyFile C:\Certbot\live\www.pharma-gsb.fr\privkey.pem
SSLCertificateChainFile C:\Certbot\live\www.pharma-gsb.fr\chain.pem
</VirtualHost>
```

7. Redirection du port SSL

De même que le port http, nous devons rediriger le port 443 qui est utilisé pour le SSL.

Redirection de port
✕

IP Destination :	UTILISA-7J5B2FP (192.168.1.172) ?
Redirection active :	<input checked="" type="checkbox"/>
IP source :	Toutes v
Protocole :	TCP ?
Port de début :	443 ? ^ v
Port de fin :	443 ? ^ v
Port de destination :	443 ? ^ v
Commentaire :	ssl

✕ Annuler
💾 Sauvegarder

8. Activation du pare-feu

Il ne faut pas oublier d'ouvrir les ports HTTP et SSL du pare-feu Windows si nécessaire. Pour cela il suffit de se rendre dans les paramètres de sécurité et d'autoriser les ports 80 et 443.

9. Configuration de Symfony

Pour passer Symfony en mode production, il suffit de remplacer ces deux lignes de code dans le fichier « .env »

```
APP_ENV=prod
APP_DEBUG=0
```

Puis installer le package apache pour symfony avec cette commande :

```
composer require symfony/apache-pack
```

ce qui a pour but de créer un fichier .htaccess dans le dossier build qui permet au navigateur d'interpréter correctement le dossier et ainsi exécuter les pages au lieu de faire une liste des fichiers et dossiers présents.

10. Script d'automatisation

Pour simplifier la mise à jour ou le redémarrage de notre application, on peut écrire un script en .bat pour pouvoir automatiser certaines commandes :

```
start C:\wamp64\wampmanager.exe
D:
cd gsb
```

```
php bin/console cache:clear
php bin/console doctrine:schema:update -f
start symfony server:start
yarn run build
exit
```

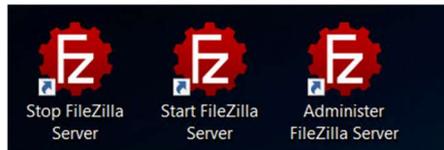
celui-ci va tout d'abord exécuter WAMP , puis aller dans le répertoire dans lequel est située l'application GSB, effectuer un nettoyage du cache, effectuer la synchronisation de la base de données, lancer le serveur symfony dans un autre terminal puis créer un build de nos pages.

11. Déploiement continu

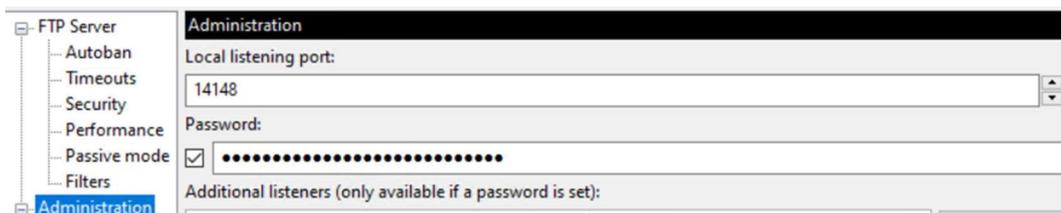
Pour ajouter le déploiement continu à notre projet, nous allons envoyer les fichiers à notre serveur dès que la branche master de notre dépôt est modifiée. Pour cela nous devons configurer un serveur FTP et utiliser le pipeline CI/CD de gitlab.

1. Serveur FTP

Le logiciel que nous allons utiliser pour faire tourner un serveur FTP sur notre machine est FileZilla Server. Il suffit de télécharger la dernière version sur le site officiel, puis de l'installer. On obtient 3 icônes :



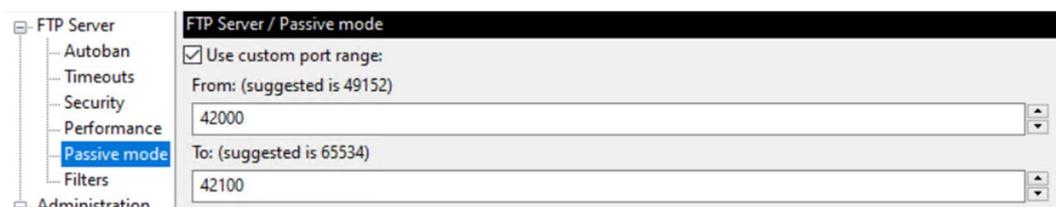
Par défaut il n'y a pas de mot de passe pour l'accès à la console, nous allons donc en configurer un :



Puis suffit de configurer l'ip d'écoute sur celle de notre serveur :



Puis les ports du mode passif :



Pour cela il suffit de définir une plage de 100 numéros de ports entre 1024 et 65535. Le mode passif permet d'envoyer une donnée via un port différent, changeant de façon

aléatoire dans cette plage pour chaque transfert, ce qui réduit les possibilités d'interception des données par une personne malveillante.

Puis nous allons avoir besoin d'un compte utilisateur pour pouvoir effectuer le transfert :

Virtual path	Native path	Writable
/pharma-gsb	D:\gsb	<input checked="" type="checkbox"/>

Pour cela on définit un nom d'utilisateur, ici « gsbGitDeploy », puis un mot de passe et enfin un point d'entrée constitué d'une adresse virtuelle, d'une adresse locale et de le définir en « inscriptible » .

Ensuite dans l'interface d'administration de notre routeur, il faut ajouter plusieurs redirections de ports :

Paramètre	Valeur
IP Destination	UTILISA-7J5B2FP (192.168.1.172)
Redirection active	<input checked="" type="checkbox"/>
IP source	Toutes
Protocole	TCP
Port de début	42000
Port de fin	42100
Port de destination	42000
Commentaire	ftp

Paramètre	Valeur
IP Destination	UTILISA-7J5B2FP (192.168.1.172)
Redirection active	<input checked="" type="checkbox"/>
IP source	Toutes
Protocole	TCP
Port de début	42421
Port de fin	42421
Port de destination	21
Commentaire	ftp 2

Sur la première, nous ouvrons les ports utilisés par le mode passif qui permet de transférer les données, et sur la seconde, nous ouvrons le port qui permet d'établir la première connexion au serveur. Sur cette deuxième redirection, nous redirigeons le port 42421 vers le port 21 en local afin d'éviter le Port Spoofing (surveillance des ports les plus utilisés pour intercepter le trafic).

2. CI/CD gitlab

Pour finir, il suffit d'ajouter un fichier .gitlab-ci.yml et d'y inscrire les lignes suivantes :

```
image: jakzal/phpqa:php7.4
before_script:
  - composer install --no-scripts
  - composer update
stages:
  - Deploy
deploy:
  stage: Deploy
  script:
    - apt-get update -qq && apt-get install -y -qq lftp
```

```

- lftp -u gsbGitDeploy,GalaxySBdeploy13 ftp.pharma-gsb.fr -p
42421 -e "set ftp:ssl-allow no;debug;mirror --parallel=10 -Rev ./
/pharmagsb --ignore-time --exclude-glob .git* --exclude .git/ --
exclude vendor/ --exclude node_modules/ --exclude public/ --exclude
var/" || true
- lftp -u gsbGitDeploy,GalaxySBdeploy13 ftp.pharma-gsb.fr -p
42421 -e "quit"
- echo "transfert terminé !"

```

Pour effectuer le transfert, nous utilisons l'outil lftp qui permet d'envoyer des données via le protocole FTP. Nous commençons par nous connecter au serveur avec le client créé précédemment :

- les paramètres `mirror` et `-rev` signifient que l'on copie les fichiers de la machine vers le serveur distant
- le paramètre `--parallel` permet d'envoyer plusieurs fichiers en parallèle via l'intervalle des ports défini sur le serveur FTP,
- le paramètre `./` permet de renseigner l'adresse locale de nos fichiers à transférer (ici la racine directement)
- le paramètre `/pharmagsb` est l'adresse virtuelle du dossier dans lequel les fichiers sont reçus
- les paramètres de type `exclude` définissent les fichiers exclus du transfert comme les `.git` et autres dossiers qui sont générés automatiquement par l'application comme le `/build`

Une deuxième commande lftp permet de fermer la connexion pour éviter qu'elle ne reste ouverte.

12. Accès distant a la base de données

Pour héberger la base de données, nous allons utiliser un serveur mysql fourni par notre hébergeur de nom de domaine amen.fr qui dispose de micro-hébergements gratuits. Pour cela il suffit de créer une base de données et un utilisateur disposant des droits sur cette base via l'interface Cpanel :



Puis il nous suffit d'ajouter l'ip de notre serveur a la liste des autorisations d'accès a la base de données :

- MySQL Remote Host: **lhcp3232.webapps.net**
- MySQL Remote Port: **3306**

Ajouter un hôte d'accès

Hôte (le caractère générique % est autorisé)

Ajouter un hôte

Tout ceci nous donne finalement l'adresse suivante pour accéder notre BDD distante :

```
DATABASE_URL="mysql://kw4o92rb_Btssioslam:Bsioslam!13@lhcp3232.webapps.net:3306/kw4o92rb_symfony_pharma?serverVersion=5.7"
```

13. Condition de déploiement

Lors du développement de l'application, nous ne voulons pas systématiquement déployer celle-ci lorsqu'une modification est faite dans le repository. Nous devons donc conditionner le déploiement à partir du fichier `.gitlab-ci.yml` :

```
image: jakzal/phpqa:php7.4
before_script:
  - composer install --no-scripts
  - composer update
.only-master:
  only:
    - master
stages:
  - Deploy
deploy:
  stage: Deploy
  extends: .only-master
  script:
    - apt-get update -qq && apt-get install -y -qq lftp
    - lftp -u gsbGitDeploy, GalaxySBdeploy13 ftp.pharma-gsb.fr -p
      42421 -e "set ftp:ssl-allow no;debug;mirror --parallel=10 -Rev ./
      /pharmagsb --ignore-time --exclude-glob .git* --exclude .git/ --
      exclude vendor/ --exclude node_modules/ --exclude public/ --exclude
      var/" || true
    - lftp -u gsbGitDeploy, GalaxySBdeploy13 ftp.pharma-gsb.fr -p
      42421 -e "quit"
    - echo "transfert terminé !"
```

On définit donc au début de ce script les branches du dépôt concernées par le déploiement pour exécuter les commandes uniquement lors que ces branches reçoivent un push ou un merge. Ici, la variable `.only-master` est associée à la branche master du repository.

Puis on étend le stage de deployment avec la variable `.only-master` pour finir l'association.