

Projet Fil Rouge
ASPTT Nantes Handball Club

Guide de déploiement

Tétiana Savitska
Vincent Canarelli
Clémentine Tournery
Romuald Tuffreau
Morgan Tréguier

Introduction

Ce projet destiné à l'association de handball Nantaise ASPTT doit permettre un déploiement facile à mettre en œuvre et à moindre coût.

L'équipe de développement ayant versionné le site sur Github, nous pourrons nous servir de Git afin de simplifier le téléchargement du code source sur le serveur de production.

Les moyens financiers de l'association étant limités, le serveur web et la base de données seront hébergés sur la même machine.

Choix de l'environnement de production

Système d'exploitation :

- Un environnement de production sous Debian 8 Jessie car il est libre, gratuit et stable.
Cette distribution utilisée la plupart du temps sur les serveurs offre un support à long terme* et possède une documentation fournie ainsi qu'une communauté active.

Solutions logicielles :

- Le serveur web apache2 proposé à l'installation de Debian (ou facile à installer via le gestionnaire de paquet apt dans le cas contraire)
- Le moteur de base de données MySQL offre la même facilité d'installation sous Debian, de plus une solution logicielle telle que LAMP ou XAMP permettent des facilités d'administrations.

* Long Term Support : les mises à jour de sécurité pour une distribution Debian 8 Jessie sont assurées jusqu'en 2020 (source : <https://wiki.debian.org/fr/LTS>)

Diagramme de déploiement

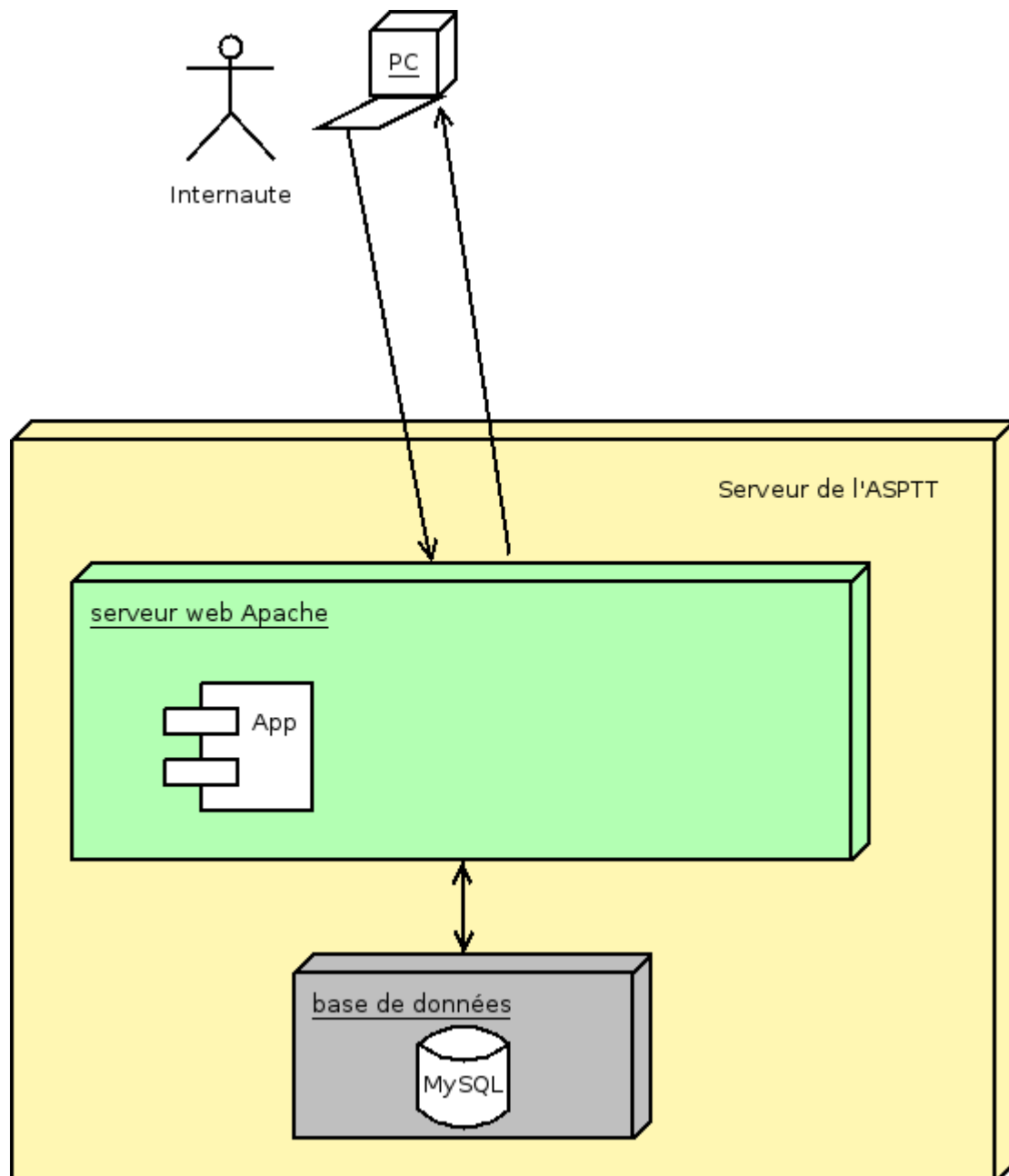
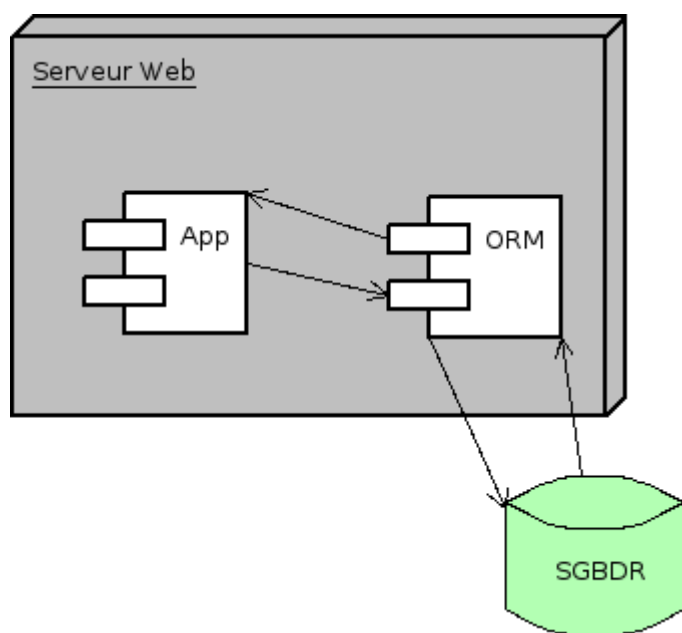


Diagramme de déploiement – Vue détaillée

Nous avons prévu une solution minimale nécessitant une seule machine afin de réduire les coûts.



Vue détaillée
dialogue entre les
composants

Préparation de l'environnement de production

Hébergement chez OVH

Créez votre serveur dédié sous Debian chez OVH. OVH propose plusieurs distributions linux et plusieurs versions de Debian, pour ce guide nous choisirons la version la plus récente : Jessie (version 8) stable.

La première chose à faire est de vous connecter par ssh sur le serveur.

Pour utiliser ssh sous linux

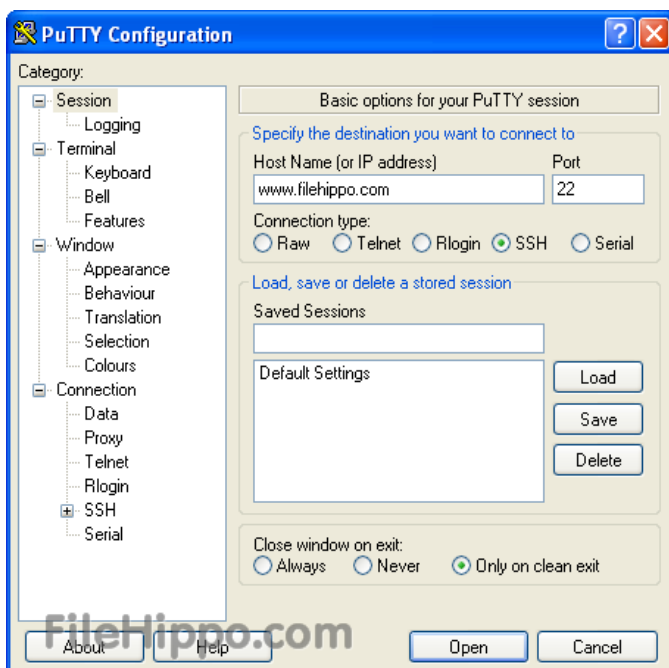
Ouvrez l'invite de commande et tapez la commande suivante :

```
ssh root@192.314.118.218 -p 22
```

Appuyez sur « y » pour autoriser la génération de clés.

Pour utiliser ssh sous windows

Téléchargez le client ssh putty, et entrez les paramètres suivants



HostName : 192.314.118.218

Port : 22

Cochez la case SSH

Appuyez sur open

Putty vous demandera confirmation, validez en appuyant sur ok et saisissez les identifiants suivants

Username : root

Password : <appuyez sur entrée>

Par mesure de sécurité il est fortement conseillé de vous créer un utilisateur sur le système pour la mise en production.

Créez un utilisateur et attribuez-lui un mot de passe.

```
useradd <username>  
passwd <username>
```

Ajoutez le au groupe SSH pour lui autoriser les connexions ssh ultérieures.

```
usermod -a -G ssh <username>
```

Revue des dépendances et pré-requis logiciels:

PHP	php 5.5.9 minimum
Symfony	symfony 3.2
Composer	permettra d'installer les dépendances nécessaires à Symfony grâce aux fichiers composer.json et composer.lock
MySQL	mysql-server
Apache	apache2
Git	

Pour une installation automatique de l'application

Tout d'abord vérifiez que wget est bien installé, dans le cas contraire installez-le et tapez la commande suivante dans le terminal de commande.

```
apt-get update && apt-get install wget
```

Vous pouvez télécharger le script de déploiement grâce à la commande suivante :

```
wget https://github.com/EquipeBravo/Projet-fil-rouge/livvable/blob/master/deploy.sh
```

Rendez le script exécutable

```
sudo chmod +x deploy.sh
```

Exécutez le script

```
./deploy.sh
```

Le script va se lancer et vous demander votre mot de passe, entrez le et appuyez sur <entrée>

Les mises à jour du système se lancent, appuyez sur O deux fois pour valider.

Composer vous demandera de paramétrer la base de données, les instructions à suivre sont les mêmes qu'à la page 11 du manuel.

Vous pouvez maintenant importer un script de sauvegarde de la base de données comme expliqué en dernière page du manuel.

Installation manuelle de l'environnement :

Vérifiez tout d'abord que le système d'exploitation et la liste des dépôts est à jour avec les commandes :

```
sudo apt-get update  
sudo apt-get upgrade  
sudo apt-get dist-upgrade
```

Lors de l'installation par le gestionnaire de paquets de debian nous conseillons d'installer également les dépendances associées au paquet téléchargé qui seront proposées.

Lors des installations en ligne de commande, vous pouvez vérifier que le paquet existe avec la commande `apt-cache search <nomdupaquet>` ou avoir des informations relatives à celui-ci avec la commande `apt-cache policy <nomdupaquet>`

Installation de l'environnement - suite

Vérifiez la version de php :

```
php -v
```

Si besoin est, installez php :

```
sudo apt-get install php5
```

Si besoin est, installez apache et MySQL :

```
sudo apt-get install apache2  
sudo apt-get install mysql-server
```

Installez Git :

```
sudo apt-get install git
```

Installez Symfony3* :

```
mkdir -p /usr/local/bin  
curl -Ls https://symfony.com/installer -o /usr/local/bin/symfony  
sudo chmod a+x /usr/local/bin/symfony
```

Créez un répertoire et placez vous dedans :

```
cd /var/www/html  
mkdir websiteAsptt  
cd websiteAsptt
```

* <https://symfony.com/doc/current/setup.html>

Installer Composer* :

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"

php -r "if (hash_file('SHA384', 'composer-setup.php') ===
'669656bab3166a7aff8a7506b8cb2d1c292f042046c5a994c43155c0be6190fa
0355160742ab2e1c88d40d5be660b410') { echo 'Installer verified'; } else
{ echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"

php composer-setup.php

php -r "unlink('composer-setup.php');"
```

Si tout s'est bien déroulé vous devriez avoir un environnement propre pour installer le site.

Installation de l'application

Téléchargement du code source :

```
git clone https://github.com/EquipeBravo/Projet-fil-rouge.git
cp composer.phar Projet-fil-rouge
cd Projet-fil-rouge
```

Veillez à vider le cache de symfony avant le téléchargement du projet avec la commande, (en local) :

```
php bin/console cache:clear
php bin/console cache:clear --env=prod
```

Ajuster les droits :

Symfony doit pouvoir lire et écrire dans le répertoire var du projet pour la mise en cache et l'écriture des logs. A partir de la racine du site :

```
chmod -R 766 var/logs  
chmod -R 766 var/cache
```

Installation des dépendances avec Composer :

```
php composer.phar install
```

Exécutez cette commande dans le répertoire où sont les fichiers composer.json et composer.lock.

Elle aura pour effet de télécharger et d'installer les dépendances du projet et de recréer le dossier vendor, nécessaire à symfony mais trop gros pour être versionné.

```
Creating the "app/config/parameters.yml" file  
Some parameters are missing. Please provide them.  
database_host (127.0.0.1):  
database_port (null):  
database_name (symfony):  
database_user (root):  
database_password (null):  
mailer_transport (smtp):  
mailer_host (127.0.0.1):  
mailer_user (null):  
mailer_password (null):  
secret (ThisTokenIsNotSoSecretChangeIt):
```

Composer vous demandera de configurer la base de données

Entrez les paramètres suivants

```
database_host : 127.0.0.1  
database_port : 1337  
database_name : asptt_db  
database_user : asptt  
database_password : <entrez le mot de passe>  
mailer_transport : <appuyez sur entrée>  
mailer_host : <appuyez sur entrée>  
mailer_user : <appuyez sur entrée>  
mailer_password : <appuyez sur entrée>  
secret :
```

* <https://getcomposer.org/download/>

Mise en place de la base de données:

Si vous souhaitez modifier les paramètres de la base de données éditez le fichier app/config/parameters.yml du projet.

Par mesure de sécurité, il est primordial de créer un utilisateur « asptt » dans mysql et de mettre un mot de passe sécurisé pour l'accès à la base.

Connectez vous à MySQL :

```
mysql -u root
```

Exécutez la requête suivante pour créer un utilisateur asptt et lui attribuer des droits sur asptt_db

```
CREATE USER 'asptt'@'%' IDENTIFIED WITH mysql_native_password AS  
<votre mot de passe>;  
GRANT USAGE ON *.* TO 'asptt_db'@'%' REQUIRE NONE WITH  
MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0  
MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;
```

Une fois que tout est prêt, créez la base de données avec la commande :

```
php bin/console doctrine:database:create
```

Puis, mettez à jour les tables de la base de données grâce à la commande :

```
php bin/console doctrine:schema:update --force
```

Importez le script install.sql dans le dossier livrable afin de créer le premier administrateur.

```
mysql -h <localhost> -u asptt -p <password>  
asptt_db < dumpasptt.sql
```

Lancement du serveur:

```
php bin/console server:run
```

Si vous souhaitez changer la configuration d'apache2 pour créer un virtualhost pour le site, vous pouvez créer les fichiers de configuration dans : /etc/apache2/sites-available/

Pour importer une sauvegarde SQL d'une base pré-existante:

A partir de la racine du projet déplacez vous dans le répertoire livrable

```
cd livrable
```

Dans le répertoire « livrable » avec le fichier au format .sql

```
mysql -h <localhost> -u <asptt> -p <password>  
asptt_db < dumpasptt.sql
```

Avec entre chevrons les données modifiables :

<localhost> : hôte

<asptt> : username

<password> : mot de passe