

Étapes requises pour configurer les éléments nécessaires pour un nouveau projet:

Le projet se trouve sur Github dans le répertoire : <https://github.com/linuq/PIAFS> .

L'intégration continue des tests se fait via Jenkins. Jenkins est hébergé via Rancher sur: <http://192.99.24.76:9998/> . Chaque fois qu'il y a un changement sur le code du projet Github, un build est lancé sur Jenkins pour faire exécuter les tests unitaires PHPunit et les tests d'acceptation Behat.

La façon la plus simple de configurer le répertoire Git est d'aller directement sur le site de Github pour créer notre répertoire et par la suite de donner les permissions à nos collaborateurs pour qu'ils puissent faire leurs pushes sur le projet.

Voici l'architecture de notre projet :

- build : fichiers de configuration pour Jenkins
- features : fichiers de tests pour Behat
- src : code source de l'application
- tests : fichiers de test pour Phpunit
- vendor : fichiers de configuration pour Composer

Une fois que notre répertoire est créé et que l'architecture de notre projet est faite, il faut lier celui-ci à notre installation de Jenkins. Pour faire cela, il faut aller sur Jenkins et se créer un nouveau projet Jenkins. Dans la configuration de ce projet, il faut sélectionner Git comme étant notre source de code et donner l'URL de notre répertoire Github.

Après avoir fait cette étape, il nous est possible de faire un build, mais les builds ne sont pas faits automatiquement. Pour activer cette option, il faut aller choisir dans la section «build trigger» de faire le lien avec les hooks Github. Il nous suffit maintenant d'aller activer le service sur Github dans les paramètres du projet. Il faut commencer par se rajouter un service Jenkins. Par la suite, dans la section «webhooks», il faut ajouter l'URL de notre installation Jenkins pour compléter la liaison entre les deux. Désormais chaque fois qu'un push est fait sur le projet Github, un build est lancé sur Jenkins.

Étapes requises nécessaires pour configurer un poste de développement:

Pour se configurer un poste de développement local, il faut commencer par installer un serveur web et tous les outils nécessaires pour le configurer. Dans cet exemple, nous décrivons les démarches pour installer le projet sur Ubuntu 16.04 avec Apache, PHP et MySQL.

Premièrement, il faut commencer par installer apache.

```
sudo apt-get update  
sudo apt-get install apache2
```

Normalement, après cette installation, si on va à l'adresse locale localhost:80 on devrait trouver une page web nous confirmant que l'installation d'apache s'est faite avec succès.

Par la suite, il faut installer notre interpréteur PHP. Dans le cadre de ce projet, nous utilisons PHP 7.0 avec une base de données MySQL, donc nous devons installer tous les outils qui permettront de faire fonctionner PHP avec MySQL.

```
sudo apt-get install php7.0 libapache2-mod-php7.0 php7.0-mcrypt php7.0-gd php7.0-dom php7.0-json
```

Ensuite, il faut configurer notre base de données. Durant cette installation, il nous sera demandé de mettre un mot de passe pour la base de données. Il est très important de se souvenir de ce mot de passe pour pouvoir utiliser notre base de données dans le futur.

```
sudo apt-get install mysql-server php7.0-mysql
```

Pour compléter l'installation de notre base de données, il faut créer la base de données que nous utiliserons pour le projet.

```
mysql -uroot -p
CREATE DATABASE piafs;
CREATE USER 'piafs'@'localhost' IDENTIFIED BY 'your-password';
GRANT ALL PRIVILEGES ON piafs . * TO 'piafs'@'localhost';
FLUSH PRIVILEGES;
quit
```

Finalement, il faut aller recopier le projet Piwigo dans notre dossier /var/www/HTML. Pour arriver à ce résultat, il faudra copier le projet venant de Github. En se plaçant dans le dossier /var/www/HTML, exécuter la commande :

```
sudo apt-get install git
git clone https://github.com/linuq/PIAFS.git
```

Normalement, la configuration devrait être terminée et complétée avec succès. Il suffit simplement d'aller à l'installation de PIAFS sur notre localhost et d'entrer les informations de notre base de données pour pouvoir terminer l'installation et il devient alors possible d'utiliser piafs localement. Il se peut qu'il doive être nécessaire de changer les permissions de notre dossier se trouvant à /var/www/html pour pouvoir accéder au site web. Changez-les si nécessaire.

Aussi, pour avoir un bon environnement de travail, nous suggérons d'utiliser un IDE tel que Atom ou PhpStorm.

Pour terminer la configuration de notre environnement de travail, il nous reste uniquement à installer nos cadriciels de test.

Voici les lignes de commande à exécuter pour l'installation de phpunit :

```
wget https://phar.phpunit.de/phpunit-6.0.phar
chmod +x phpunit.phar
sudo mv phpunit.phar /usr/local/bin/phpunit
```

L'installation de behat est un peu plus complexe. La façon la plus simple pour installer Behat est de l'installer grâce au gestionnaire de dépendances PHP Composer. Nous devons commencer par nous

créer un fichier composer.json qui sera utilisé par Composer pour qu'il sache quoi installer. Dans ce fichier nous allons écrire :

```
{
  "require-dev": {
    "behat/behat": "~2.5"
  },
  "config": {
    "bin-dir": "bin/"
  }
}
```

Par la suite, nous allons procéder à l'installation de Composer pour terminer la configuration de Behat.

```
curl -sS https://getcomposer.org/installer | php
php composer.phar install
```

Après cette étape, on devrait avoir un projet fonctionnel sur lequel on peut exécuter nos tests localement et la configuration devrait être complètement terminée.