

Deploiement projet - procedure

Objectif : Mettre en ligne une application existante Symfony sur une machine Debian9

1. Creation server
2. Installation serveur
 - 2.1 connexion ssh
 - 2.2 installer sudo (optionnel)
 - 2.3 Mettre a jour Debian
 - 2.4 Installer Apache
 - 2.5 Installer PHP 7.2
 - 2.6 Installer MySQL
 - 2.7 Installer composer
 - 2.8 Installer git
3. Mise en production d'un projet existant symfony
 - 3.1 Les droits Apache et admin Gandi
 - 3.2 Rapatriement du projet
 - 3.3 Installation projet
 - 3.4 .htaccess & VirtualHost

1. CREATION SERVER - Interface gandi 4 <https://v4.gandi.net>

Note : Seule les actions listées ci dessous sont nécessaires. Les autres valeurs sont à laisser par défaut

- Onglets : Services > Serveurs
- Bouton "Créé un serveur"
- Etape 1/2 - Page de création
 - RAM 512 (idealement 1024 sur un vrai environnement de production)
 - Valider
- Etape 2/2 - Page "Configurer votre serveur"
 - Système > Systeme d'exploitation
 - "Debian 9" (systeme d'exploitation le plus utilisé coté unix avec RedHat / Centos)
 - Paramètres de connexion
 - saisir "Identifiant administrateur"
 - saisir "Mot de passe"
 - ressaisir le mdp dans "Confirmation du mot de passe"

Fiche recap admin sys : <https://github.com/O-clock-Alumni/fiches-recap/blob/master/ldc/adminsys.md>

Pour plus d'informations :

Benchmarks : <https://symfony.fi/entry/symfony-benchmarks-scaling-php-by-adding-cpu-ram>

1. INSTALLATION SERVEUR

2.1 connexion ssh

- cliquer au niveau de l'interface gandi sur le serveur fraîchement créé et récupérer son ip
- taper dans le terminal `ssh votreIdentifiantServer@monIpServer`, acceptez le fingerprint puis entrer son mot de passe

2.2 installer sudo (optionnel)

Note : Si sudo n'est pas installé il faudra alors passer par su (super utilisateur pour installer vos paquets)

- `su`
- `apt-get install sudo`
- `nano /etc/sudoers`
- Rajouter la ligne `admin ALL=(ALL:ALL) ALL`

Note : si votre utilisateur server n'est pas la valeur par défaut (admin) , remplacer `admin` par votre utilisateur

- Quitter le mode `su` en tapant `exit`

Note : si `exit` ne fonctionne pas du premier coup, réitérer l'action.

2.3 Mettre à jour Debian

Note: Si sudo n'est pas installé il faudra alors passer par su (super utilisateur pour installer vos paquets)

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

2.4 Installer Apache

Note: A faire avant l'installation de PHP car cela permet de d'activer la librairie PHP / Apache par la suite

- `sudo apt-get install apache2 -y`
- Activer le module de réécriture d'url `sudo a2enmod rewrite`

- Restart du server pour activer la reecriture `sudo service apache2 restart`

2.5 Installer PHP 7.2

- `sudo apt-get install apt-transport-https lsb-release ca-certificates`
- `sudo wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg`
- `echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main" | sudo tee /etc/apt/sources.list.d/php.list`
- `sudo apt update`
- `sudo apt install php7.2`

Vérifiez votre installation : `php -v`

- Installer les librairies nécessaires à Apache / PHP et Symfony : `sudo apt install php7.2-cli php7.2-common php7.2-curl php7.2-gd php7.2-json php7.2-mbstring php7.2-mysql php7.2-xml libapache2-mod-php7.2 php7.2-zip php7.2-mysql`

2.6 Installer MySQL

- `sudo apt-get install mysql-server`
- Important pour la liaison entre le driver MySql / PHP `sudo service apache2 restart`
- Tester votre connexion en root avec `sudo mysql -u root` puis taper entrer
- Si c'est ok, une console MariaDb s'ouvrira. Rester dans la console afin de vous créer deux utilisateurs utilisables sans sudo dont un spécifique Symfony.

Pour un user root classique, dans la console MariaDB :

- `DROP USER 'root'@'localhost';`
- `CREATE USER 'root'@'localhost' IDENTIFIED BY '';`
- `GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION;`
- `FLUSH PRIVILEGES;`

Pour un utilisateur ayant l'identifiant `admin` et le mot de passe `adminpwd`:

- `CREATE USER 'admin'@'localhost' IDENTIFIED BY 'adminpwd';`
- `GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost' WITH GRANT OPTION;`
- `FLUSH PRIVILEGES;`

2.7 Installer composer

- `php -r "copy('https://getcomposer.org/installer', '/tmp/composer-setup.php');"`
- `sudo php /tmp/composer-setup.php --install-dir=/usr/local/bin --filename=composer`

2.8 Installer git

- `sudo apt-get install git-core`
- `git config --global user.name "Claire fortin"` (remplacer par votre nom)

- `git config --global user.email claire@oclock.io`

Note: Afin de simplifier la procédure nous clonerons nos projet en HTTPS. Pour l'instant cela n'est pas encore possible cf 3.1 Les droits Apache et admin Gandi.

Fiche recap

- git & SSH : <https://github.com/O-clock-Alumni/fiches-recap/blob/master/ldc/git.md>
- journée admin sys : <https://github.com/O-clock-Alumni/fiches-recap/blob/master/ldc/installation-complete.md>

Pour plus d'informations :

- Installation sudo : <https://www.geek17.com/fr/content/debian-9-stretch-installer-et-configurer-sudo-61>
- Installation server : <https://www.howtoforge.com/tutorial/how-to-setup-symfony-4-on-debian-9/>
- Installation php7.2 & debian9 : <https://www.chris-shaw.com/blog/installing-php-7.2-on-debian-8-jessie-and-debian-9-stretch>

3. Mise en production d'un projet existant symfony

3.1 Les droits Apache et admin Gandi

Pour pouvoir cloner et ecrire dans /var/www il nous faudra rajouter les droits suffisant aux users present sur notre machine

- `sudo apt-get install acl`
- `sudo setfacl -dR -m u:www-data:rwX -m u:admin:rwX /var`
- `sudo setfacl -R -m u:www-data:rwX -m u:admin:rwX /var`

Note: Il faut substituer l'utilisateur *admin* par le nom de votre utilisateur admin créé si différent.

3.2 Rapatriement du projet

- Se connecter sur github
- Se positionner sur le projet souhaité et cliquer sur le bouton `clone or download`
- Sur la pop-in , cliquer sur le lien `Use HTTPS` pour changer le lien récupérable via HTTPS
- Cloner le repository dans /var/www/html (ex `git clone https://github.com/O-clock-Invaders/Symfo-E19-tests.git`)
- A la demande du terminal saisir votre Github username / password (nécessaire au mode HTTPS)

Note: Pour utiliser le mode SSH il sera nécessaire de répéter la procédure issue de la fiche récap dédiée à Git (generation des clefs SSH & github account settings)

3.3 Installation projet

- `composer install`
- dupliquer le fichier `.env.dist` à la racine du projet pour creer le `.env` si inexistant :
 - `cd /var/www/html/monprojet`

- `mv .env.dist .env`
- Modifier le fichier `.env` (`nano .env`) une **premiere** fois pour :
 - Editer la ligne suivante pour la base de donnée de production
`DATABASE_URL=mysql://admin:admin@127.0.0.1:3306/moviedb`

Note : Adapter les credentials si besoin en fonction du user créé précédemment dans MySql

Ensuite, effectuer les actions suivantes **avant** le changement d'environnement. En effet, dans notre cas (movieDB), nous souhaitons utiliser nos fixtures et cette commande ne peut être exécutée qu'en environnement de développement (cf `require-dev` de `composer.json`).

- `php bin/console cache:clear`
- `php bin/console doctrine:database:create`
- `php bin/console doctrine:migration:migrate`
- `php bin/console doctrine:fixtures:load`
- La commande custom d'import de poster réalisée lors du projet `app:...`

Puis

- Modifier le fichier `.env` (`nano .env`) une **deuxieme** fois pour :
 - Remplacer `APP_ENV=dev` en `APP_ENV=prod`. Ceci permettra alors à votre application d'utiliser les configurations disponibles dans `config/packages/prod` et d'en plus afficher la debug toolbar.

En condition réelle de production les fixtures ne sont généralement pas utilisées. L'ordre des actions (modifications de `.env`) évoqué est relatif à la mise en production du projet MovieDB qui contient des données générées par les fixtures.

Astuce : Le fichier `.env` étant ignoré du projet, votre configuration de développement ne pourra pas écraser vos modifications effectuées dans le `.env` de l'environnement de production.

3.4 .htaccess & VirtualHost

Installer le pack apache pour Symfony 4 qui permettra la création du `.htaccess` dans le dossier public de votre projet. Cela permettra vis à vis de votre configuration projet avec votre virtualHost (cf 2.4 Installer Apache) de prendre en compte les bonnes redirections grâce au mode `rewrite` déjà installé.

- `composer require symfony/apache-pack`

Pour activer la réécriture pour nos sites dans `/var/www/html`

- `cd /etc/apache2/sites-available/`
- ouvrir le fichier de configuration avec `sudo vim 000-default.conf`
- rajouter ce bloc à la fin à l'intérieur et à la fin du bloc `VirtualHost *:80`

```
<Directory "/var/www/html">
```

```
Options FollowSymLinks MultiViews
```

```
    AllowOverride All

    Order allow,deny

    Allow from All

</Directory>
```

Pour donner ceci

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

    DocumentRoot /var/www/html/symfo-movieDB/public

    <Directory /var/www/html/symfo-movieDB/public>
        AllowOverride All
        Order Allow,Deny
        Allow from All
    </Directory>

</VirtualHost>
```

Il est aussi courant de placer son projet en dehors du dossier `html` soit directement dans `/var/www` ; Si tel est votre cas, modifier les urls de votre virtual host en conséquence.

Note: le lien du projet `<Directory /var/www/html/symfo-movieDB/public>` est celui utilisé pour le projet de demonstration qui s'apelle `symfo-movieDB`. Ce path sera à remplacer par le vrai path de votre projet cloné.

- Puis restart apache `sudo service apache2 restart` (prise en compte des modifications)
- Vérifier que le site et ses urls sont fonctionels : Aller sur l'url de votre site à partir de son IP ex: `http://xxx.xx.xxx.xx/symfo-movieDB`
- Si il y a une erreur 500, verifier avec `cat` ou `tail -f` dans `/var/log/apache2/error.log` ce qui s'y passe puis regler les problèmes remontés.

Note : Pour fonctionner avec Apache votre version de PHP doit impérativement être supérieure ou égale à 7.1.3

Pour plus d'informations:

- Symfony & apache : https://symfony.com/doc/current/setup/web_server_configuration.html
- Gérer son environnement de production (KNP) : <https://knpuniversity.com/screencast/symfony-fundamentals/environment-tweaks>
- En complement deploiement symfony: <https://symfony.com/doc/current/deployment.html>

Fiche recap :

deploiement SF : <https://github.com/O-clock-Alumni/fiches-recap/blob/master/symfony/themes/deploiement.md>

Note : Certaines manipulations de ce fichiers sont propre à l'installation avec Debian. De plus, toutes les manipulations d'installation hors symfony doivent être effectuées avec su ou sudo. Cette fiche récap vient en complément de cette procédure