ROBIN
Mathias
RT122

FICHE DE PROCEDURE D'INSTALLATION

I. Présentation

Dans cette fiche de procédure d'installation, vous trouverez toutes les clés vous permettant de réaliser la mise en place de votre serveur web ainsi que le déploiement de votre application web. Ce document a été conçu pour vous guider pas à pas à travers chaque étape du processus. L'objectif principal de cette procédure est de fournir une solution complète et cohérente pour installer et déployer votre application web.

II. Installation

I.I Installation d'application

Commençons tout d'abord par l'installation, pour réaliser un niveau de déploiement service web et base de donnée séparé, il nous faudra l'installation de deux machine virtuelle, en l'occurrence ici des machines virtuelle VMware, une machine **windows 10** pour la base de données et une linux **debian 12** pour le server web. Guide d'installation Debian 12 , Guide d'installation Windows 10.

Par la suite nous allons effectuer la base de données sur mysql et par conséquent vous allez devoir installer mysql (la dernière version stable), ne pas oublier de **configurer un user et un mdp** à l'installation de mysql, vous pouvez trouver un lien vers l'installation de mysql ici. N'oublier pas d'installer le projet django dans le github, c'est le dossier contenant le site web dynamique, nous aurons besoin de ce dernier pour pouvoir le mettre sur votre serveur de production.

I.II Installation de package

Avant de commencer à configurer quoi que ce soit il va falloir veiller à ce que toutes les fonctionnalités dont on a besoin sont installées et prêt à l'emploi pour pouvoir travailler dans de bonnes conditions. Vous allez trouver ci-dessous une liste des commandes à effectuer sur linux **dans l'ORDRE impérativement**.

Commandes d'installation sous linux (`lorem ipsum` = commandes linux) :

- `apt install python3` ⇒ Si python n'est pas installer ce paquet est nécessaire pour faire fonctionner django.

- `apt install python3-django` ⇒ Sert à installer django.

- `apt install python3-pip` ⇒ Sert à installer l'installeur de librairie python, nécessaire pour l'installation d'autre paquet.

- `apt install python3-virtualenv` ⇒ Sert à l'installation d'un environnement virtuel nécessaire pour faire fonctionner le projet django.

- `apt install apache2` ⇒ Sert à installer le service apache2, nécessaire pour l'hébergement de notre serveur web.

- `apt install apache2 libapache2-mod-wsgi-py3` ⇒ Ces paquets sont à installer pour des fonctionnalités supplémentaires à apache 2.

- `apt install python3-dev` ⇒ Paquet nécessaire pour l'installation des paquets ci-dessous.

- `apt install pkg-config libmariadb-dev-compat libmariadb-dev` ⇒ Paquets nécessaires pour pouvoir utiliser la base de données et django.

- `pip3 install mysqlclient` ⇒ !! Avant d'effectuer cette commande veuillez vous mettre dans un environnement virtuel, nous verrons cet aspect un peu plus tard, veuillez donc suivre la suite de ce tuto et à l'instant où vous avez activé l'environnement virtuel vous pourrez effectuer cette commande.

III. Virtualisation

III.I Virtualisation côté linux

Dans cette partie nous allons nous pencher sur la mise en place de votre serveur web sur linux. Dans un premier temps vous allez devoir vous rendre à la racine de la machine linux : `cd /` par la suite nous allons créer le dossier contenant tout ce qui concerne le site web : `mkdir django-project.` Vous allez devoir par la suite recréer l'arborescence suivante :

```
/django-project/
|
|--- site
|    |-- logs
|
|
|--- src
|   |-- (Concerne tout le projet django)
|
|--- venv
```

à l'aide des commandes suivantes :
`mkdir` ⇒ permet de créer des dossier (dans le dossier ou l'on se situe)
`cd` ⇒ permet de changer de dossier (cd .. pour revenir en arrière).
Pour le dossier 'venv' vous devez faire cette commande : `virtualenv venv -p python3`

Une fois l'arborescence créée il va falloir y déposer le site web (projet django) disponible dans le github, vous allez devoir par conséquent transférer les fichier du github vers votre vm linux, pour ce faire vous pouvez utiliser FileZilla. Ne pas oublier d'accorder les droits au fichiers django-project : `chmod 777 /django-project/`. Une fois les fichiers concernant le projet déposés dans le dossier src, nous allons activer l'environnement virtuel : `source venv/bin/activate`.

Nous allons passer à la configuration apache, vous devez d'abord installer les deux fichiers de configuration concernant apache, '**apache2.conf**' , '**000-default.conf**' que vous pouvez retrouver dans le github, le fichier apache2.conf se situe dans le dossier **/etc/apache2,** il vous suffit de le supprimer (`rm`) et de le remplacer (`mv` pour déplacer des fichier) vous allez faire de-même avec le fichier 000-default.conf qui lui se situe dans le dossier **/etc/apache2/sites-available** il vous suffira de faire la manipulation que pour l'autre fichier.

Petite clarification, vous êtes supposez retrouver toutes ces lignes dans le fichier '**000-default.conf**':

```
<VirtualHost *:80>
        ErrorLog /django-project/site/logs/error.log
        CustomLog /django-project/site/logs/access.log combined

        alias /static /django-project/site/public/static
        <Directory /django-project/site/public/static>
                Require all granted
        </Directory>

        <Directory /django-project/src/myproject>
        <Files wsgi.py>
                Require all granted
        </Files>
        </Directory>

        WSGIDaemonProcess toto python-home=/django-project/venv python-path=/django-pr
        WSGIProcessGroup toto
        WSGIScriptAlias / /django-project/src/myproject/myproject/wsgi.py
</VirtualHost>
```

Pour '**apache2.conf**' ces lignes non commenté doivent être présentent:

```
<Directory />
        Options FollowSymLinks
        AllowOverride None
        Require all denied
</Directory>

<Directory /usr/share>
        AllowOverride None
        Require all granted
</Directory>

<Directory /var/www/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>

<Directory /srv/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>
```

III.II Virtualisation côté mysql

Dans cette partie nous allons nous occuper de la base de données sur mysql. Rendez-vous donc dans la vm windows 10 avec **mysql commande line client** d'ouvert vous allez devoir effectuer les commandes suivantes (`LOREM IPSUM` = commandes mysql) :

- `CREATE USER nom_user;` ⇒ ici vous choisissez le nom d'utilisateur que vous souhaitez.
- `CREATE DATABASE ludotheque;` ⇒ ici le nom de la base de donnée est ludotheque.
- `GRANT ALL PRIVILEGES ON ludotheque.* TO 'nom_user@'%';` ⇒ ici on ajoute tous les droits à l'utilisateur 'nom_user' à la base de donnée 'ludotheque'.
- `FLUSH PRIVILEGES;` ⇒ applique les privilèges précédemment établis.

Petites précisions veuillez vérifier impérativement le fichier settings.py dans projet pour qu'il corresponde à la base de donnée précédemment créé, pour ce faire :
`nano /django-project/src/myproject/myprject/settings.py`

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'ludotheque',
        'USER': 'toto',
        'PASSWORD': 'toto',
        'HOST': '192.168.5.136',
        'PORT': '3306',
    }
}
```

veillez à ce que le **user** créer auparavant **corresponde** à celui dans le settings.py, de même pour le **nom de la base de donnée**. En ce qui concerne la ligne **'HOST'** il faut l'adresse **ip de la machine windows**.

IV. Finalisation

Pour finir il vous suffit de faire quelque lignes de commandes en vente dans le répertoire /django-project/src/myproject/ :

- `Python3 manage.py makemigration`
- `python3 manage.py migrate` ⇒ permet de migrer le models.py vers votre base de donnée.
- `python3 manage.py runserver` ⇒ permet de lancer le serveur.

Ainsi vous pouvez observer les modifications sur la base de donnée avec ces commandes :

- `USE ludotheque;`
- `SHOW TABLES;` ⇒ si vous observez les tables qui se sont actualisées c'est que votre migration c'est bien passé.

Vous pouvez donc enfin accéder à votre site web grâce à l'adresse local 127.0.0.1:800