Partie configuration du serveur Web :

Nous avons opté pour une implémentation via Apache avec l’utilisation du mode wsgi sur une debian10.

Pour se faire il nous faut d’abord installer les paquets des applicatifs qui vont bien à savoir la fonction pip de python libapache muni du mod wsgi via la commande suivante :

sudo apt-get update

sudo apt-get install python3-pip apache2 libapache2-mod-wsgi-py3

on peut maintenant configurer l’environnement virtuel pour cela il nous suffit de suivre les étapes suivante :

***sudo pip3 install virtualenv***

Puis on crée un répertoire pour notre projet

mkdir ~/myproject

cd ~/myproject

une fois dans le répertoire on crée notre environnement virtuel

virtualenv myprojectenv

puis on l’active

source ~/myproject/myprojectenv/bin/activate

on la sait activer dès que le début de notre prompt débute par

(myprojectenv)

On installe donc django

(myprojectenv) $ pip install django

On créer notre projet

django-admin.py startproject myproject ~/myproject

dans le dossier settings on configure les root par lesquels django doit passer pour accéder aux différent éléments de la base

on y configure les adresse ip du serveur

pour cela il faut compléter les lignes suivante dans notre fichier settings

nano ~/myproject/myproject/settings.py

ALLOWED\_HOSTS = ["10.129.14.16", "127.0.0.1", "127.0.1.1"]

STATIC\_URL = '/static/'

STATIC\_ROOT = os.path.join(BASE\_DIR, 'static/')

On n’oublie pas de migrer tous cela afin que les modifications apporter soient prises en compte

(myprojectenv) $cd ~/myproject

(myprojectenv) $manage.py makemigrations

(myprojectenv) $manage.py migrate

On crée un utilisateur administratif

(myprojectenv) $./manage.py createsuperuser

On demande à django de collecter toutes les informations static via le root implémenter précédemment dans le fichier settings

(myprojectenv) $./manage.py collectstatic

On implémente une règle autorisant les trafique tcp via le port 8000 de la vm

(myprojectenv) $sudo iptables -I INPUT -p tcp --dport 8000 -j ACCEPT

Il reste plus qu’à lancer le serveur

(myprojectenv) $./manage.py runserver 0.0.0.0:8000

Puis à rentrer l’adresse ou le nom du domaine dans votre navigateur web ainsi que le port

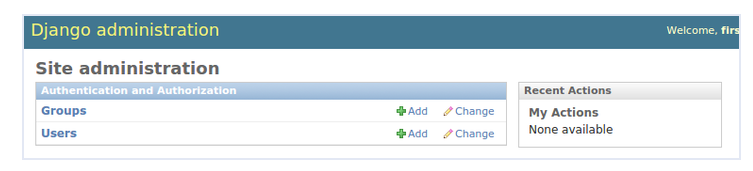
<http://10.129.14.16:8000>

Vous atterrissez sur une fenêtre de ce type :

Une image contenant texte

Description générée automatiquement

Une les informations de l’utilisateur administratif créer plutôt vous a une fenêtre de ce type qui apparait



Nous pouvons à présent éteindre le serveur via un contrôle c dans la terminale et désactiver l’environnement virtuel

(myprojectenv) $ deactivate

On peut maintenant configurer apache

sudo nano /etc/apache2/sites-available/000-default.conf

<VirtualHost \*:80>

. . .

Alias /static /home/sammy/myproject/static

<Directory /home/sammy/myproject/static>

Require all granted

</Directory>

<Directory /home/sammy/myproject/myproject>

<Files wsgi.py>

Require all granted

</Files>

</Directory>

WSGIDaemonProcess myproject python-home=/home/toto /myproject/myprojectenv python-path=/home/toto/myproject

WSGIProcessGroup myproject

WSGIScriptAlias / /home/toto/myproject/myproject/wsgi.py

</VirtualHost>

Il nous faut ensuite modifier les droits d’accès à la base de données pour que le groupe des propriétaires puisse avoir les droits de lecture et d’écriture

chmod 664 ~/myproject/db.sqlite3

chmod 775 ~/myproject

on a aussi besoin de mettre la base de données dans le groupe de propriétaires www-data

sudo chown :www-data ~/myproject/db.sqlite3

sudo chown :www-data ~/myproject

on doit ensuite modifier le port de transite des données en annulant celle précédemment injecter et la remplaçant par la nouvelle

sudo iptables -D INPUT -p tcp --dport 8000 -j ACCEPT

sudo iptables -I INPUT -p tcp --dport 80 -j ACCEPT

il faut tester le fichier apache afin d’être sur qu’il n’y a pas d’erreur de syntaxe

sudo apache2ctl configtest

Output

. . .

Syntax OK

Si vous avez se retour c’est que tout est bon

Afin d’appliquer les modifications on doit redémarer le service apache

sudo systemctl restart apache2.