

FICHE DE PROCÉDURE D'INSTALLATION

I. Présentation

Ce document fournit des instructions détaillées pour installer et déployer votre serveur web et application web. Il vise à guider chaque étape pour assurer une mise en place réussie.

II. Installation

I.I Installation des applications nécessaires

Pour séparer le service web et la base de données, deux machines virtuelles sont nécessaires :

- **Base de données sur Windows 10** : Configurez une VM Windows 10 pour héberger la base de données MySQL.
- **Serveur web sur Linux Debian 12** : Configurez une VM Debian 12 pour héberger votre serveur web Apache et l'application Django.

Guides d'installation recommandés :

- **Debian 12** : <https://www.debian.org/distrib/index.fr.html>
- **Windows 10** : <https://developer.microsoft.com/en-us/windows/downloads/virtual-machines/>

Installation de MySQL :

- Téléchargez et installez la dernière version stable de MySQL.
- Configurez un utilisateur et un mot de passe lors de l'installation.
- [Lien pour télécharger MySQL](#)

Projet Django :

- Clonez le projet « Gestion_film_version_final » depuis GitHub : <https://github.com/MacheG0ule/SAE23.git>

I.II Installation des paquets

Assurez-vous que toutes les fonctionnalités nécessaires sont installées. Voici les commandes à exécuter sur Linux dans cet ordre :

apt install python3 → *Installe Python requis pour faire fonctionner Django*

apt install python3-django → *Installe Django*

apt install python3-pip3 → *Installe pip, le gestionnaire de packages Python*

apt install python3-virtualenv → *Installe l'environnement virtuel pour Django*

apt install apache2 → *Installe Apache2 pour héberger le serveur web*

apt install apache2 libapache2-mod-wsgi-py3 → *Installe des modules supplémentaire Apache2*

apt install python3-dev → *Nécessaire pour l'installation de certains packages*

apt install pkg-config libmariadb-dev-compat libmariadb-dev → *Nécessaire pour utiliser MySQL avec Django*

Activer l'environnement virtuel avant d'exécuter la commande suivante : **pip3 install mysqlclient**

III. Virtualisation

III.I Virtualisation côté Linux

Pour configurer le serveur web sous Linux :

1. Accédez à la racine de la machine Linux : ``cd /``
2. Créez le dossier pour le projet Django : ``mkdir django-project``
3. Recréez l'arborescence suivante :

Faire l'arborescence pour le projet django à la racine / :

```
/django-project/  
|  
|--- site  
|   |-- logs  
|   |-- public  
|       |-- media  
|       |-- static  
|  
|--- src  
|   |-- ( tout les fichiers du django)  
|  
|--- venv
```

Commandes pour créer des dossiers :

- `mkdir` - Crée des dossiers.
- `cd` - Change de dossier (utiliser `cd ..` pour revenir en arrière). - `virtualenv venv -p python3` - Crée l'environnement virtuel.

Transférer les fichiers du projet Django :

- Utilisez FileZilla pour transférer les fichiers du GitHub vers la VM Linux. - Accordez les droits aux fichiers : `chmod 777 /django-project/`.

Activer l'environnement virtuel :

- `source venv/bin/activate`

Configuration d'Apache :

- Remplacez les fichiers de configuration `apache2.conf` et `000-default.conf` avec ceux du GitHub :

modifier le fichier `/etc/apache2/apache2.conf` :

Python

```
# access here, or in any related virtual host.  
<Directory />  
    Options FollowSymLinks  
    AllowOverride None  
    Require all granted <-- à modifier  
</Directory>
```

- `000-default.conf` se trouve dans `/etc/apache2/sites-available`.

Modifier le fichier `000-default.conf` dans le dossier `/etc/apache2/sites-available/` :

```
Python Copier

<VirtualHost *:80>

    ErrorLog /django-project/site/logs/error.log
    CustomLog /django-project/site/logs/access.log combine

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

    WSGIDaemonProcess film python-home=/django-project/venv python-path=/django-project/src/
    WSGIProcessGroup film
    WSGIScriptAlias / /django-project/src/Gestion_film/wsgi.py

</VirtualHost>
```

III.II Virtualisation côté MySQL

Pour configurer la base de données sous MySQL :

```
CREATE USER 'nom_user';
CREATE DATABASE filmographie;
GRANT ALL PRIVILEGES ON filmographie.* TO 'nom_user'@'%';
FLUSH PRIVILEGES;
```

Vérification du fichier settings.py :

- Vérifiez que le fichier `settings.py` correspond à la base de données créée : `nano /djangoproject/src/Gestion_film/settings.py`

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'filmographie',
        'USER': 'film',
        'PASSWORD': 'Filmfilm1!',
        'HOST': '192.168.115.136',
        'PORT': '3306',
    }
}
```

-Puis dans le dossier static il faut ajouter et modifier les URLs et transférer le section static :

Le dossier static pour le css :

Dans le fichier `nano /django-project/src/Gestion_film/settings.py`

```
# SECURITY WARNING: don't run with debug turned on in production!  
DEBUG = True <--- Le mettre en False
```

Modifier le lien static url

```
# Static files (CSS, JavaScript, Images)  
# https://docs.djangoproject.com/en/5.0/howto/static-files/  
  
STATIC_URL = 'static/'  
STATIC_ROOT = '/django-project/site/public/static' <--| à rajouter  
MEDIA_ROOT = '/django-project/site/public/media' <--| à rajouter
```

transférer les static :

```
(venv) root@debian:/django-project/src# python manage.py collectstatic
```

IV. Finalisation

Pour finaliser l'installation, exécutez les commandes suivantes dans le répertoire ``/django-project/src/myproject/`` :

python3 manage.py makemigrations → *Prépare les migrations pour votre base de données*

python3 manage.py migrate → *Applique les migrations à la base de données*

python3 manage.py runserver → *Lance le serveur web*

Vérifier les modifications dans la base de données :

USE filmographie;

SHOW TABLES;

Accédez enfin à votre site web via l'adresse locale : 127.0.0.1:8000.