



Abdel <siabdel25@gmail.com>

installer une plateforme dev fullstack sous Debian 12

1 message

Abdel <siabdel25@gmail.com>

Thu, Dec 11, 2025 at 9:20 AM

To: siabdel25@gmail.com

Voici une **procédure complète, claire et optimisée** pour transformer votre **PC HP Pavilion 17"** sous **Debian 12** en **plateforme de développement full-stack** :

Django + DRF + Celery + PostgreSQL + Vue.js + Docker + Node/npm + Python venv + IA (PyTorch/Transformers)

avec les **meilleurs outils recommandés en 2025**.

✓ 1. Mise à jour du système

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y build-essential curl wget git unzip software-properties-common
```

✓ 2. Installer Python 3.11+ et outils dev

Debian 12 a Python 3.11 par défaut, mais installez les outils nécessaires :

```
sudo apt install -y python3 python3-dev python3-venv python3-pip
```

Vérification :

```
python3 --version
pip3 --version
```

✓ 3. Installer PostgreSQL + outils

```
sudo apt install -y postgresql postgresql-contrib libpq-dev
```

Créer un utilisateur PostgreSQL pour Django :

```
sudo -u postgres createuser --interactive
# Name: votre_user
# Superuser: yes
```

Créer une base de données :

```
sudo -u postgres createdb projet_db -O votre_user
```

✓ 4. Installer Node.js 20 LTS + npm + pnpm (optionnel mais conseillé)

Ajouter le dépôt officiel NodeSource :

```
curl -fsSL https://deb.nodesource.com/setup\_20.x | sudo -E bash -
sudo apt install -y nodejs
```

Installer PNPM (meilleure performance que npm/yarn)

```
sudo npm install -g pnpm
```

Vérifier :

```
node -v
```

```
npm -v
```

```
pnpm -v
```



5. Installer Docker + Docker Compose

Docker simplifie énormément vos services (Postgres, Redis, Celery Worker, Vue frontend...)

Installer Docker Engine

```
sudo apt install -y ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/debian $(lsb_release -cs) stable" \
| sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-
plugin
```

⚠ Permettre à Docker de fonctionner sans sudo :

```
sudo usermod -aG docker $USER
newgrp docker
```



6. Installer Redis (pour Celery)

```
sudo apt install -y redis-server
```

Activer le service :

```
sudo systemctl enable redis --now
```



7. Installer les dépendances IA : PyTorch + Transformers

⚠ PyTorch officiel s'installe mieux via pip.

Dans votre environnement virtuel Python :

```
pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cpu
pip install transformers accelerate
pip install sentencepiece
```



8. Créer un environnement de développement Django

Créer un dossier :

```
mkdir ~/dev && cd ~/dev  
mkdir projet_fullstack && cd projet_fullstack
```

Créer un venv :

```
python3 -m venv venv  
source venv/bin/activate
```

Installer Django + DRF + Celery + psycopg2 + outils utiles :

```
pip install django djangorestframework celery redis psycopg2-binary  
pip install django-cors-headers python-dotenv  
pip install gunicorn
```

✓ 9. Créer le projet Django

```
django-admin startproject backend  
cd backend
```

Créer une app API :

```
python manage.py startapp api
```

Appliquer les migrations :

```
python manage.py migrate
```

Lancer :

```
python manage.py runserver
```

✓ 10. Configurer Celery dans Django

Dans backend/backend/celery.py :

```
from __future__ import absolute_import  
import os  
from celery import Celery  
  
os.environ.setdefault("DJANGO_SETTINGS_MODULE", "backend.settings")  
app = Celery("backend")  
app.config_from_object("django.conf:settings", namespace="CELERY")  
app.autodiscover_tasks()
```

Dans backend/backend/__init__.py :

```
from .celery import app as celery_app  
__all__ = ("celery_app",)
```

Installer dépendances :

```
pip install flower
```

✓ 11. Installer Vue.js 3 + Vite

Dans un dossier frontend :

```
cd ~/dev/projet_fullstack  
npm create vue@latest frontend
```

```
cd frontend
npm install
npm run dev
```



12. Intégration Django + Vue.js (fullstack)

Utiliser :

- ✓ **django** pour l'API
 - ✓ **django-cors-headers** pour autoriser Vue.js
 - ✓ **Vite** pour build frontend → production dans /dist
-



13. Installer les meilleurs outils DevOps

1. Poetry (gestion des dépendances Python)

```
curl -sSL https://install.python-poetry.org | python3 -
```

2. Docker Compose pour orchestrer :

- Django API
- PostgreSQL
- Redis
- Celery worker
- Celery beat
- Flower
- Vue.js dev/prod

Fichier `docker-compose.yml` modèle :

```
version: "3.9"

services:
  db:
    image: postgres:15
    environment:
      POSTGRES_USER: dev
      POSTGRES_PASSWORD: dev
      POSTGRES_DB: devdb
    ports:
      - "5432:5432"
    volumes:
      - postgresdata:/var/lib/postgresql/data

  redis:
    image: redis:7
    ports:
      - "6379:6379"

  backend:
    build: ./backend
    command: gunicorn backend.wsgi:application --bind 0.0.0.0:8000
    volumes:
      - ./backend:/app
    ports:
```

```
- "8000:8000"
depends_on:
  - db
  - redis
```

```
celery:
  build: ./backend
  command: celery -A backend worker --loglevel=INFO
  volumes:
    - ./backend:/app
  depends_on:
    - backend
    - redis
```

```
celery_beat:
  build: ./backend
  command: celery -A backend beat --loglevel=INFO
  volumes:
    - ./backend:/app
  depends_on:
    - backend
    - redis
```

```
volumes:
  postgresdata:
```



14. Outils recommandés pour la productivité



Visual Studio Code

```
sudo apt install -y wget gpg
wget -qO- https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor > packages.microsoft.gpg
sudo install -D packages.microsoft.gpg /etc/apt/keyrings/microsoft.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/microsoft.gpg] \
https://packages.microsoft.com/repos/code stable main" | sudo tee /etc/apt/sources.list.d/
vscode.list
sudo apt update
sudo apt install code
```

Extensions utiles :

- ✓ Python
- ✓ Pylance
- ✓ Vue Language Features
- ✓ Docker
- ✓ GitLens



15. Optimisation de performance sur HP Pavilion

👉 Activez l'accélération CPU :

```
sudo apt install -y intel-microcode
```

👉 Tuning Postgres :

```
sudo apt install -y pg_tune
```

👉 Utilisez un SSD si possible

👉 Activez swap zram pour accélérer un vieux CPU/RAM :

```
sudo apt install zram-tools
```



Votre PC Debian 12 est maintenant une plateforme complète fullstack + IA !

Si vous voulez, je peux aussi vous fournir :

- ✓ un modèle complet d'architecture Django/DRF + Vue
- ✓ une structure de projet propre
- ✓ un Dockerfile optimisé
- ✓ un pipeline CI/CD DevOps
- ✓ la configuration Celery prête à l'emploi
- ✓ un exemple d'intégration IA (classifieur Transformers dans Django)

Souhaitez-vous ces éléments ?