

DEPLOIEMENT EC2 AWS

Voici les **étapes complètes pour déployer un projet Django sur une instance AWS EC2 Ubuntu avec Gunicorn et Nginx** :

PRÉREQUIS

- Avoir un compte AWS et une instance EC2 Ubuntu en fonctionnement.
- Avoir un projet Django prêt.
- Un nom de domaine (optionnel).
- Clé SSH pour accéder à l'instance EC2(optionnel).

1. Connexion SSH à l'instance EC2

```
ssh -i chemin/vers/ta-cle.pem ubuntu@<ip-public-de-ton-ec2>
```

Ou tu peux aller dans ton instance Ec2 crée puis dans se connecter et enfin client SSH.

2. Installation des dépendances

```
sudo apt update && sudo apt upgrade -y
```

```
sudo apt install python3-pip python3-dev libpq-dev nginx curl git -y
```

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

```
sudo apt install mysql-server libmysqlclient-dev -y
```

3. Configurer MySQL (création de la base de données et de l'utilisateur)

Accéder à MySQL :

```
sudo mysql
```

Puis dans la console MySQL(Cas Esatic concours) :

```
CREATE DATABASE esatic_bd CHARACTER SET UTF8MB4 COLLATE utf8mb4_general_ci;
```

```
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'esatic@2025';
```

```
GRANT ALL PRIVILEGES ON esatic_bd.* TO 'admin'@'localhost';
```

```
FLUSH PRIVILEGES;
```

```
EXIT;
```

4. Cloner ton projet

```
git clone https://github.com/ton-projet.git
cd ton-projet
```

5. Créer et activer l'environnement virtuel

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

6. Installer les dépendances du projet

```
pip install -r requirements.txt
```

7. Configurer les paramètres Django Si pas encore fait

Dans settings.py :

- DEBUG = False
- ALLOWED_HOSTS = ['<ip-public>', 'nom-de-domaine.com']
- Configurer STATIC_ROOT :

```
STATIC_ROOT = os.path.join(BASE_DIR, 'static/')
```

- Utiliser Mysql :
import pymysql
pymysql.install_as_MySQLdb()

8. Collecter les fichiers statiques et exécuter les migrations pour la base de donnée

```
python manage.py collectstatic
python manage.py migrate
```

9. Configurer Gunicorn

Test Gunicorn :

```
gunicorn --bind 0.0.0.0:8000 nom_du_projet.wsgi:application
```

(Pour que le test fonctionne ajouter un TCP personnalisé sur la plage de port 8000 et sur la source 0.0.0.0/0(n'import ou) dans les règles de sécurité entrantes de l'instance)

Créer un service systemd :

```
sudo nano /etc/systemd/system/gunicorn.service
```

Ensuite à l'intérieur du fichier :

[Unit]

Description=gunicorn daemon

After=network.target

[Service]

User=ubuntu

Group=www-data

WorkingDirectory=/home/ubuntu/ton-projet

ExecStart=/home/ubuntu/ton-projet/venv/bin/gunicorn --workers 3 --bind 0.0.0.0:8000 ton-projet.wsgi:application

[Install]

WantedBy=multi-user.target

Puis :

```
sudo systemctl start gunicorn
```

```
sudo systemctl enable gunicorn
```

```
sudo systemctl status gunicorn
```

9. Configurer Nginx

Créer un fichier de config :

```
sudo nano /etc/nginx/sites-available/monprojet
```

ensuite entrer ses données :

```
server {
```

```
    listen 80;
```

```

server_name <ip-public-ou-nom-de-domaine>;

location = /favicon.ico { access_log off; log_not_found off; }

location /static/ {
    alias /home/ubuntu/tonprojet/staticfiles/;
    try_files $uri $uri/ =404;
}

location /media/ {
    alias /home/ubuntu/tonprojet/media/;
    try_files $uri $uri/ =404;
}

location / {
    proxy_pass http://0.0.0.0:8000;
    include proxy_params;
}
}

```

Activer la config :

```

sudo ln -s /etc/nginx/sites-available/monprojet /etc/nginx/sites-enabled
sudo nginx -t
sudo systemctl restart nginx

```

10. Autoriser les ports dans AWS

- Aller dans l'onglet **Sécurité > Groupe de sécurité**.
- Éditer les règles d'entrée : autoriser le port **80 (HTTP)** et **443 (HTTPS)** si SSL.

11. Redémarrer les services

```

sudo systemctl daemon-reload
sudo systemctl restart gunicorn

```

```
sudo systemctl restart nginx
```

```
sudo systemctl reload nginx
```

cas de problème GUNICORN :

```
sudo systemctl daemon-reexec
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart gunicorn
```

NB :

```
sudo chmod o+x /home/ubuntu
```

Pour avoir accès au static donner les permissions :

```
sudo chown -R www-data:www-data /home/ubuntu/tonprojet/staticfiles
```

```
sudo chmod -R 755 /home/ubuntu/tonprojet/staticfiles
```

Pour avoir accès au static donner les permissions :

```
sudo chown -R www-data:www-data /home/ubuntu/tonprojet/media
```

```
sudo chmod -R 755 /home/ubuntu/tonprojet/media
```