



Projet 10

Déployez votre application sur un serveur comme un pro !

OpenClassrooms
Développeur d'application - Python

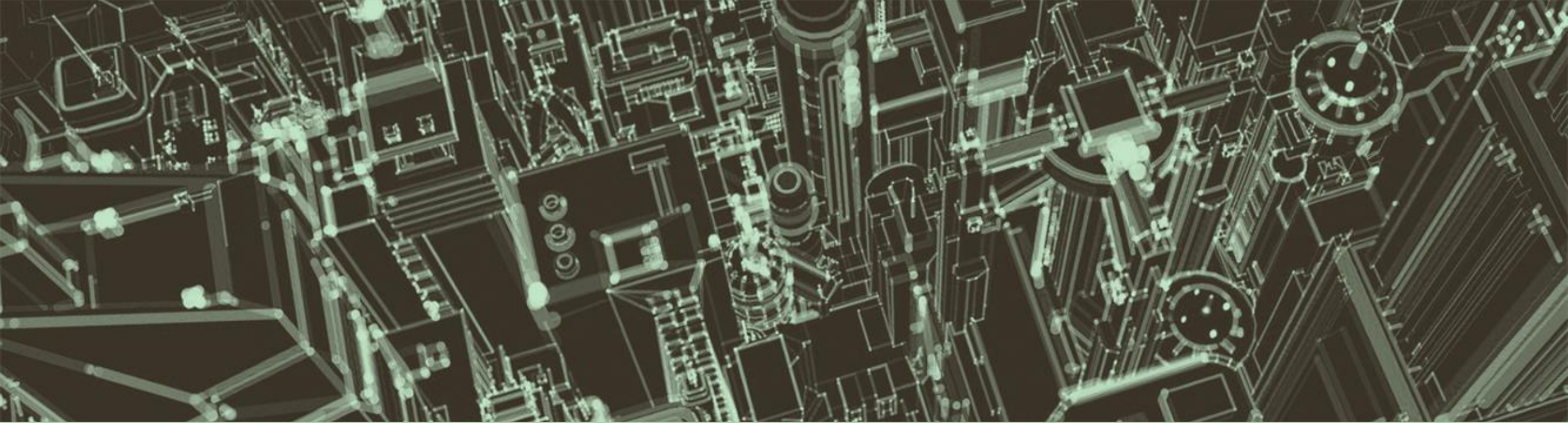


Le projet

Le but de ce projet était de déployer l'application PurBeurre sur un serveur qui n'automatise pas le déploiement comme Heroku et donc de le gérer entièrement.

Requis:

- Configurer le serveur et effectuer un déploiement en ligne de console
- Mettre en place l'intégration continue
- Surveiller le bon fonctionnement de l'application avec des outils de monitoring
- Automatiser la mise à jour les éléments récupérés d'Open Food Facts une fois par semaine.



Création et paramétrage du serveur




Création et paramétrage du serveur

- Droplet Digital Océan

DROPLETS (1)

purbeurre-mo1

Image	 Ubuntu 20.04 (LTS) x64	Region	LON1
Size	1 vCPUs 1GB / 25GB Disk (\$5/mo) Resize	IPv4	165.22.118.210
		IPv6	Enable
		Private IP	10.106.0.2
		VPC	default-lon1

Système d'exploitation :
Ubuntu (Linux)

Ajout de la clé SSH publique

Création et paramétrage du serveur

- Paramètres du Firewall

The screenshot shows the DigitalOcean dashboard for a project named 'manuo1'. The left sidebar contains navigation links for PROJECTS, MANAGE, and various services. The main content area displays the 'Firewalls' section for a specific firewall named 'purbeurre-mo1-firewall', which has 5 rules and is applied to 1 Droplet. The 'Rules' tab is selected, showing a list of inbound rules. Two rules are visible: SSH (TCP, port 22) and HTTP (TCP, port 80), both allowing traffic from all IPv4 and IPv6 addresses. To the right of the rules list, there are two dark grey buttons labeled 'Accès HTTP' and 'Accès SSH'.

Search by resource name or public IP (Ctrl+B) Create USAGE \$2.82

← Firewalls

purbeurre-mo1-firewall
5 Rules / 1 Droplet

Rules Droplets Destroy

Firewall rules control what inbound and outbound traffic is allowed to enter or leave a Droplet.

Inbound Rules

Set the Firewall rules for incoming traffic. Only the specified ports will accept inbound connections. All other traffic will be blocked.

Type	Protocol	Port Range	Sources
SSH	TCP	22	All IPv4 All IPv6 More
HTTP	TCP	80	All IPv4 All IPv6 More

Accès HTTP

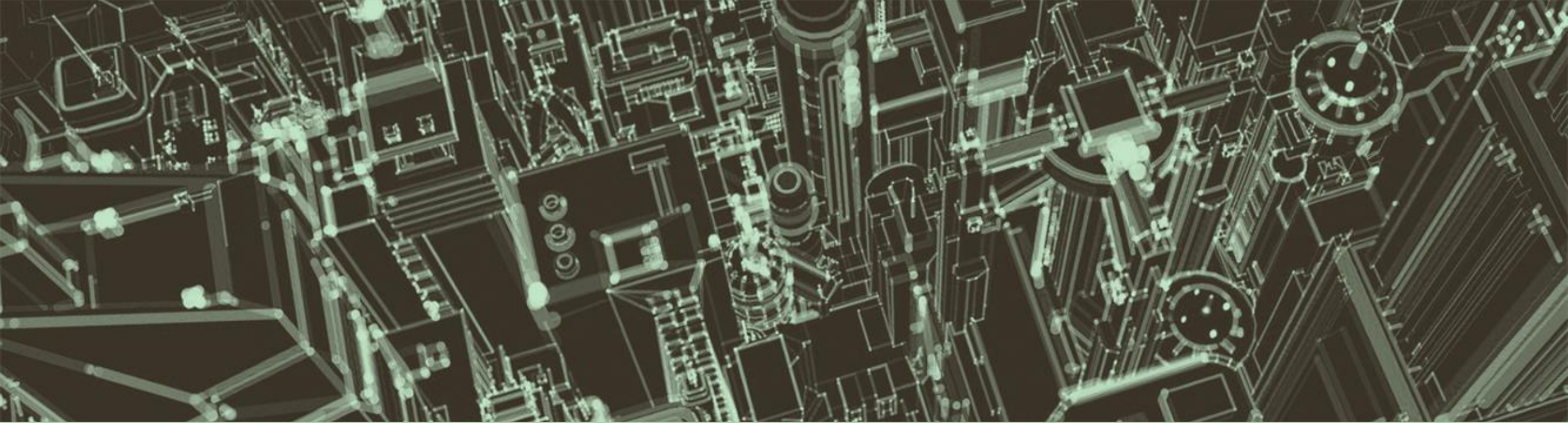
Accès SSH

Création et paramétrage du serveur

- Nouvel utilisateur non root mais super utilisateur (sudo)
- Désactivation de l'accès SHH pour l'utilisateur root

```
adduser manuo1  
gpasswd -a manuo1 sudo  
su - manuo1  
mkdir .ssh  
chmod 700 .ssh  
touch .ssh/authorized_keys  
vi .ssh/authorized_keys  
chmod 600 .ssh/authorized_keys  
exit
```

Aucune action importante ne peut être effectuée à distance sans le mot de passe du nouvel utilisateur.



Installation de l'application sur le serveur



Installation de l'application sur le serveur

- Installation de Python et PostgreSQL
- Pull du dépôt GitHub
- Mise en place de l'environnement virtuel
- Installation des dépendances
- Paramétrage de PostgreSQL

```
sudo apt-get install python3-pip python3-dev libpq-dev postgresql postgresql-contrib
git clone https://github.com/manuo1/P10PurBeurre-IAAS-deployment.git
mv P10PurBeurre-IAAS-deployment purbeurre-mo1
sudo apt install virtualenv
virtualenv env -p python3
source env/bin/activate
pip install -r purbeurre-mo1/requirements.txt
sudo -u postgres psql
    postgres=# CREATE DATABASE purbeurredb;
    CREATE USER manuo WITH PASSWORD 'cestunmotdepassesecurise';
    ALTER ROLE manuo SET client_encoding TO 'utf8';
    ALTER ROLE manuo SET timezone TO 'Europe/Paris';
    GRANT ALL PRIVILEGES ON DATABASE purbeurredb TO manuo;
    \q
```


Installation de l'application sur le serveur

- Variables d'environnement

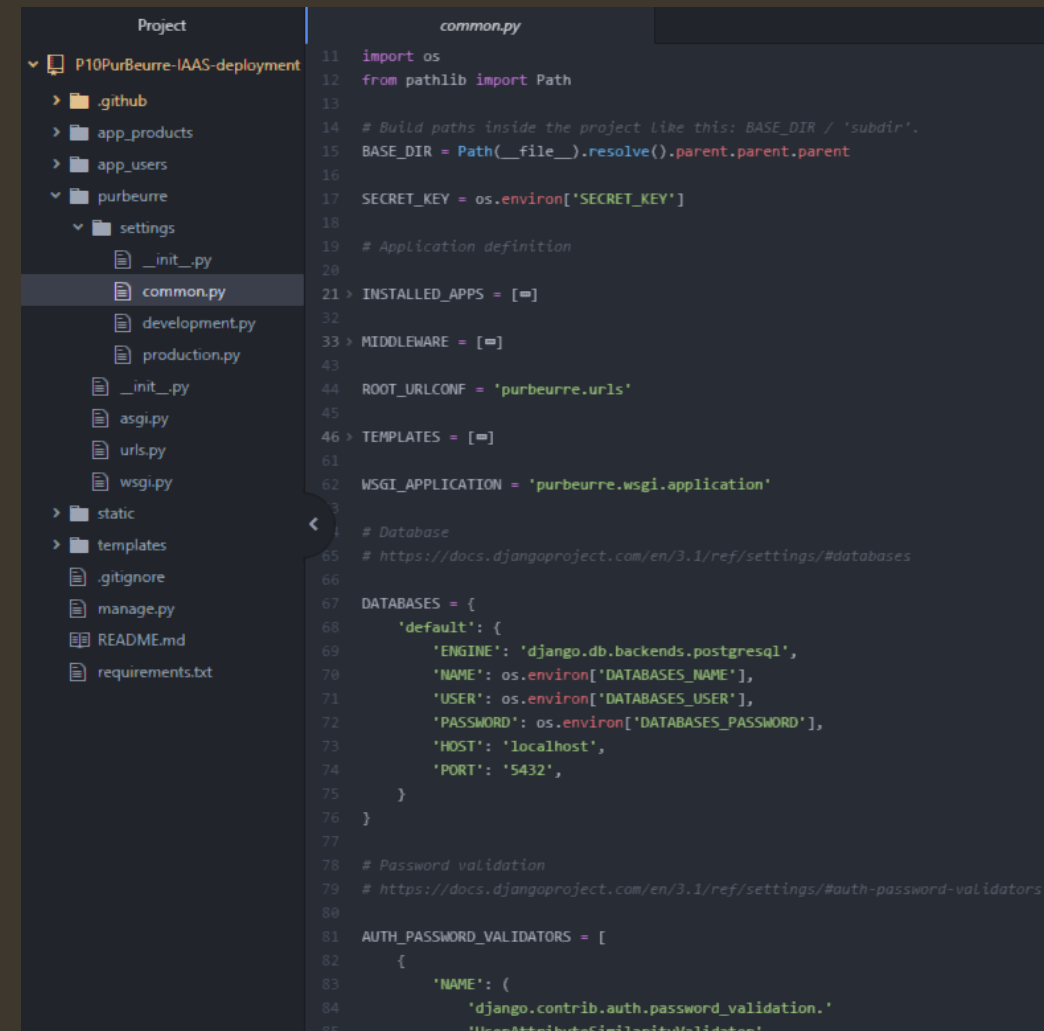
vi .env

```
ENVIRONMENT = 'production'
SECRET_KEY = 'cécîèstùnètrèsràndèclèsècrètédjàngo'
DATABASES_NAME = 'purbeurredb'
DATABASES_USER = 'manuo'
DATABASES_PASSWORD = 'cestunmotdepassesecurise'
```

Tous les fichiers sont versionnés

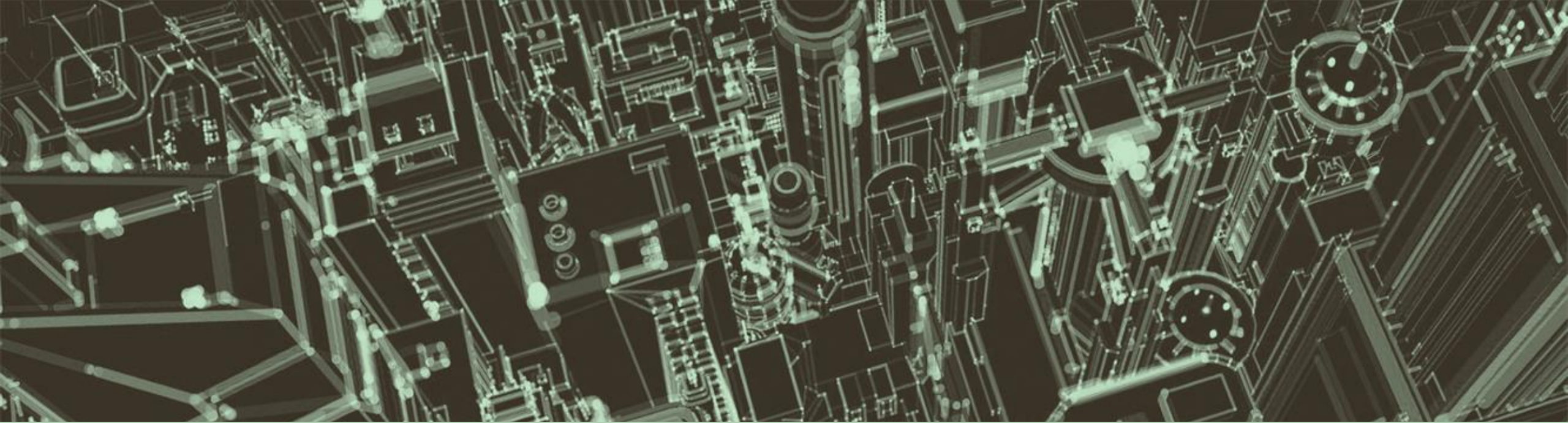
Rien à modifier dans le code pour le déploiement

Toutes les données sensibles sont cachées



The screenshot shows a code editor with a project file structure on the left and a `common.py` file on the right. The project structure includes a `P10PurBeurre-IAAS-deployment` directory with subdirectories `.github`, `app_products`, `app_users`, `purbeurre`, and `settings`. The `settings` directory contains `_init_.py`, `common.py`, `development.py`, and `production.py`. The `purbeurre` directory contains `_init_.py`, `asgi.py`, `urls.py`, and `wsgi.py`. The `common.py` file contains the following code:

```
11 import os
12 from pathlib import Path
13
14 # Build paths inside the project like this: BASE_DIR / 'subdir'.
15 BASE_DIR = Path(__file__).resolve().parent.parent.parent
16
17 SECRET_KEY = os.environ['SECRET_KEY']
18
19 # Application definition
20
21 INSTALLED_APPS = []
22
23 MIDDLEWARE = []
24
25 ROOT_URLCONF = 'purbeurre.urls'
26
27 TEMPLATES = []
28
29 WSGI_APPLICATION = 'purbeurre.wsgi.application'
30
31 # Database
32 # https://docs.djangoproject.com/en/3.1/ref/settings/#databases
33
34 DATABASES = {
35     'default': {
36         'ENGINE': 'django.db.backends.postgresql',
37         'NAME': os.environ['DATABASES_NAME'],
38         'USER': os.environ['DATABASES_USER'],
39         'PASSWORD': os.environ['DATABASES_PASSWORD'],
40         'HOST': 'localhost',
41         'PORT': '5432',
42     }
43 }
44
45 # Password validation
46 # https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators
47
48 AUTH_PASSWORD_VALIDATORS = [
49     {
50         'NAME': (
51             'django.contrib.auth.password_validation.'
52             'UserAttributeSimilarityValidator'
```



Serveur HTTP
Serveur d'application WSGI
Supervisor



Serveurs HTTP et d'application WSGI

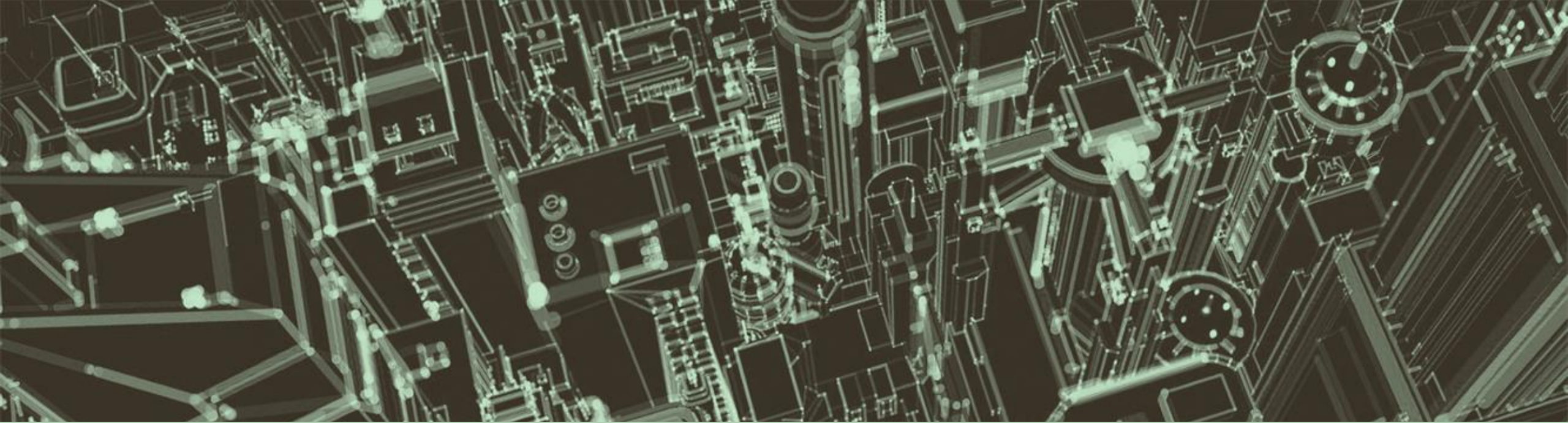
- Le serveur HTTP **NGinx** a été installé et configuré pour servir les fichiers statiques (CSS, JavaScript, images...) directement mais va faire suivre les requêtes correspondant à une ressource dynamique vers le serveur d'application WSGI.
- Le serveur d'application WSGI **Gunicorn** avait été installé avec les dépendances de l'application et Django définit par défaut une configuration WSGI qui sera utilisé par Gunicorn.

```
server {  
    listen 80;  
    server_name 165.22.118.210;  
    root /home/manuo1/purbeurre-mo1/  
    location /static {  
        alias /home/manuo1/purbeurre-mo1/staticfiles/  
    }  
    location / {  
        proxy_set_header Host $http_host;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_redirect off;  
        if (!-f $request_filename) {  
            proxy_pass http://127.0.0.1:8000;  
            break;  
        }  
    }  
}
```


Supervisor

- **Supervisor** est utilisé pour démarrer ou redémarrer le serveur d'application automatiquement

```
[program:purbeurre-mo1-gunicorn]
command = /home/manuo1/env/bin/gunicorn purbeurre.wsgi:application
user = manuo1
directory = /home/manuo1/purbeurre-mo1
autostart = true
autorestart = true
stderr_logfile=/home/manuo1/purbeurre-mo1/supervisor_stderr.log
```



Intégration Continue








Intégration Continue

GitHub Actions

Simplement ajouter des scripts, appelés « workflows », au dépôt GitHub du projet.

Cela permet d'utiliser des variables d'environnement cryptés qui pourront être utilisées dans les workflows.

Repository secrets		
 ID_RSA	Updated 12 days ago	Update Remove
 PASSPHRASE	Updated 12 days ago	Update Remove
 SERVER_IP	Updated 12 days ago	Update Remove
 SERVER_USERNAME	Updated 12 days ago	Update Remove
 SUDO_PASSWORD	Updated 12 days ago	Update Remove

[Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)

Get started with GitHub Actions

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow template to get started.

Skip this and [set up a workflow yourself](#) →

Workflows made for your Python repository Suggested


Publish Python Package

By GitHub Actions

Publish a Python Package to PyPI on release.

[Set up this workflow](#)

```
python -m pip install --upgrade pip
pip install setuptools wheel twine
python setup.py sdist bdist_wheel
```

 actions/starter-workflows Python

Intégration Continue

[Workflow - testing&deployment.yml](#)

- Déclencheurs
- Mise en place de l'environnement
- Tests
- Si les tests réussissent
=> Déploiement

[Workflow - testing on pull request.yml](#)

- Déclencheurs
- Mise en place de l'environnement
- Tests

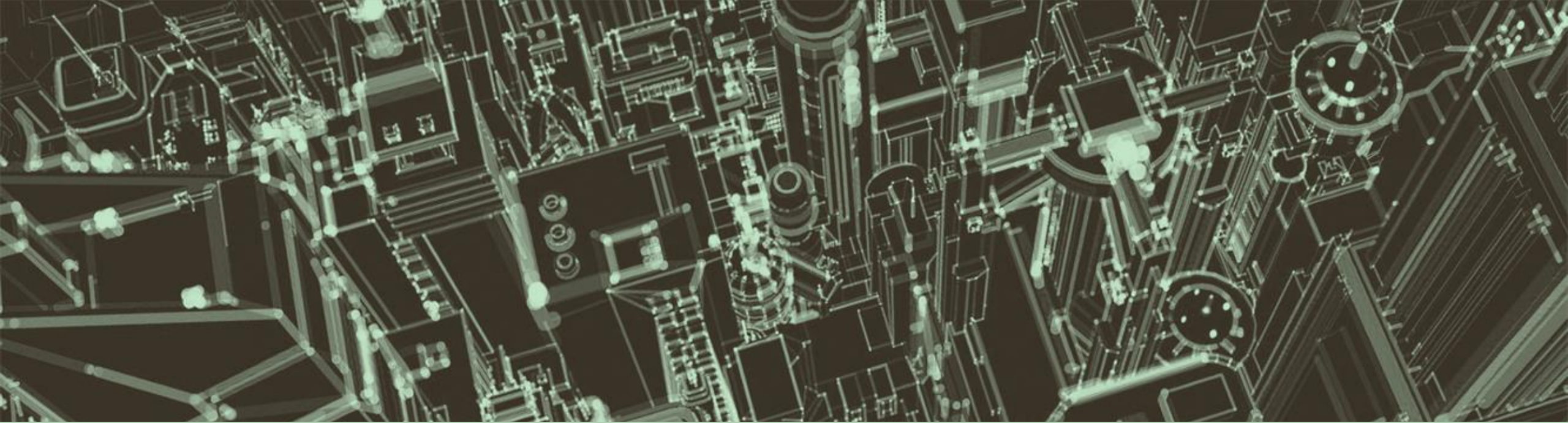
105 lines (101 sloc) | 3.81 KB

Raw

Blame



```
1  name: testing and deployment
2
3  # ${ secrets.XXXX } are defined in github secrets :
4  # on your repositories -> Settings -> Secrets
5
6  on:
7    push:
8      branches: [ master ]
9    pull_request:
10     branches: [ master ]
11
12  jobs:
13    # #####
14    # Tests execution Job
15    # #####
16
17    test:
18      name: Tests execution Job
19      runs-on: ubuntu-latest
20      strategy:
21        matrix:
22          python-version: [3.8]
23      env:
24        ENVIRONMENT: 'production'
25        SECRET_KEY: 'secret_key'
26        DATABASES_NAME: 'purbeurredb'
27        DATABASES_USER: 'manuo'
28        DATABASES_PASSWORD: 'mdp'
29      services:
30        postgres:
31          # Docker Hub image
32          image: postgres
33          # Provide the data for postgresSQL
```



Monitoring



Monitoring

- [Digital Ocean](#)

L'hébergeur propose une surveillance du serveur, qui permet de suivre et créer des alertes par exemple sur le taux d'utilisation du processeur ou de la mémoire.







Si ces taux étaient durablement trop élevé il conviendrait de changer de « plan » pour conserver un confort de navigation sur l'application.

```
curl -sSL https://agent.digitalocean.com/install.sh | sh
```

Monitoring

Alert Policies

[Setup instructions](#)[Create alert policy](#)

Name	Applied to	
 CPU is running higher than 70% CPU is above 70% for 5 min	 purbeurre-mo1	More 
 Memory Utilization is running higher than 70% Memory Utilization is above 70% for 5 min	 purbeurre-mo1	More 

Send alerts via

☒ Email manuo1rm@gmail.com

Monitoring

- [Sentry](#)

Sentry affiche un tableau de bord pour dans ce ca, la surveillance des logs de l'application.

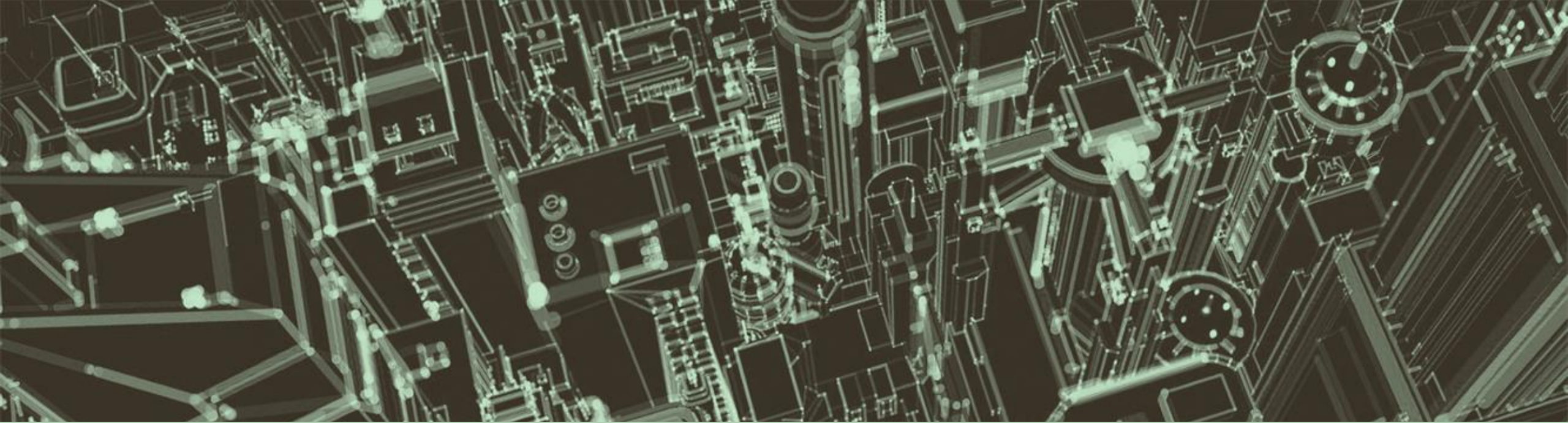
Surveiller les logs permet de remonter certaines erreurs qui subsisteraient dans le code de l'application ou une éventuelle mauvaise utilisation.

[Sentry issues](#)

```
pip install --upgrade sentry-sdk
```

```
#purbeurre\settings\production.py
import sentry_sdk
from sentry_sdk.integrations.django import DjangoIntegration
sentry_sdk.init(
    dsn="https://751878c9886d41ed853ed05124aa3c62@o516111.ingest.sentry.io/5622155",
    integrations=[DjangoIntegration()],
    traces_sample_rate=1.0,
    send_default_pii=True
)
```

<input type="checkbox"/>	<input checked="" type="checkbox"/> Resolve <input type="button" value="v"/>	<input checked="" type="checkbox"/> Ignore <input type="button" value="v"/>	Merge	<input type="button" value="⋮"/>	<input type="button" value="▶"/>	GRAPH:	24h 14d	EVENTS	USERS	ASSIGNEE
<input checked="" type="checkbox"/>	ZeroDivisionError /sentry-debug/									
	Unhandled division by zero							2	2	v
	PURBEURRE-A	a minute ago — 6 days old								
<input type="checkbox"/>	Forbidden (CSRF cookie not set.): /									
	/							42	27	v
	PURBEURRE-5	28 minutes ago — 9 days old								
		django.security.csrf								



Mise à jour des données alimentaires



Mise à jour des données alimentaires

- **Script de mise à jour**

Le dossier /bin qui contient le script a été ajouté aux paths.

Le script utilise Python de l'environnement virtuel et lance la commande personnalisée django.

On rend le script exécutable.

Et une méthode a été ajoutée à la commande personnalisée pour enregistrer dans un fichier de logs les dates de chaque mise à jour.

```
mkdir $HOME/bin  
export PATH=$PATH:$HOME/bin  
vi populatedb.sh
```

```
#!/bin/bash  
/home/manuo1/env/bin/python /home/manuo1/purbeurre-mo1/manage.py populatedb 10000
```

```
chmod +x populatedb.sh
```

```
63 #app_products\management\commands\populatedb.py  
64 def add_infos_to_populatedb_log(self, msg):  
65     populatedb_log_file = os.path.join(settings.BASE_DIR, 'populatedb.log')  
66     with open(populatedb_log_file, "a") as log:  
67         log.write("\n" + str(datetime.now()) + "\n" + msg)  
68         log.write(str(self.db_infos()))
```


Mise à jour des données alimentaires

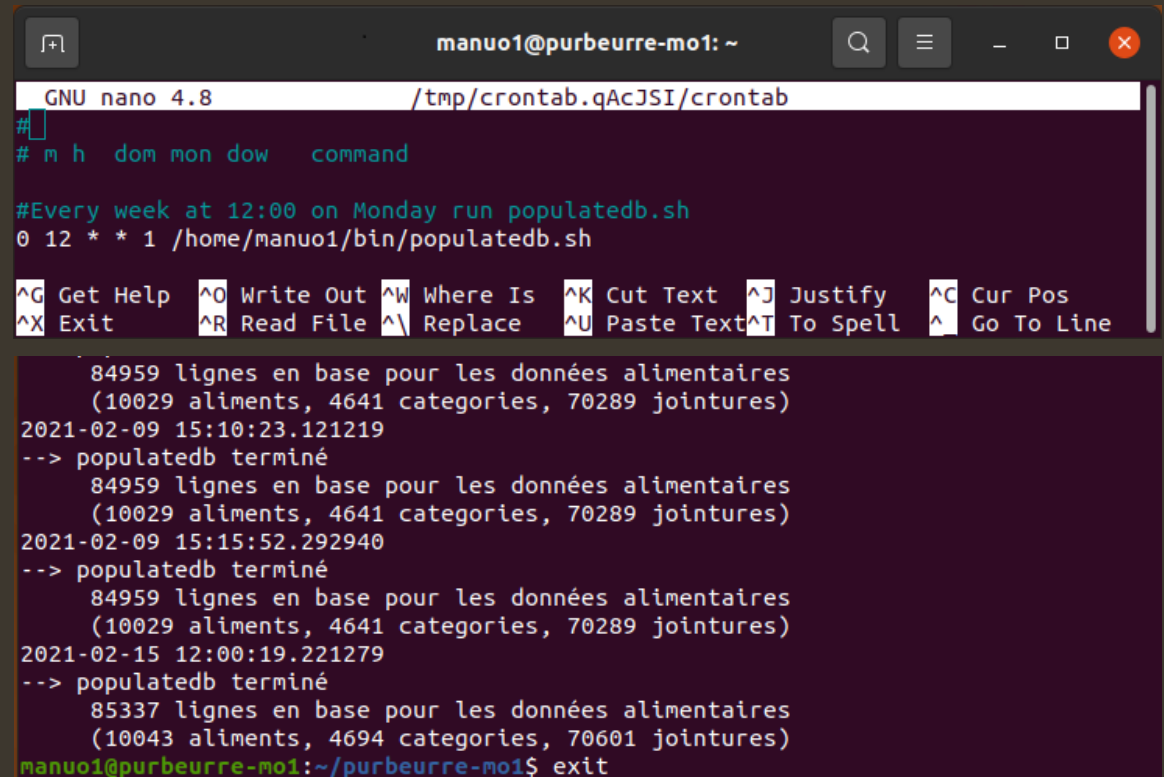
- Tache CRON

cron est un programme qui permet notamment d'exécuter automatiquement des scripts selon un cycle défini.

Tous les lundi à 12H le script est exécuté.

On peu vérifier que les mise a jours s'ajoutes bien dans les logs.

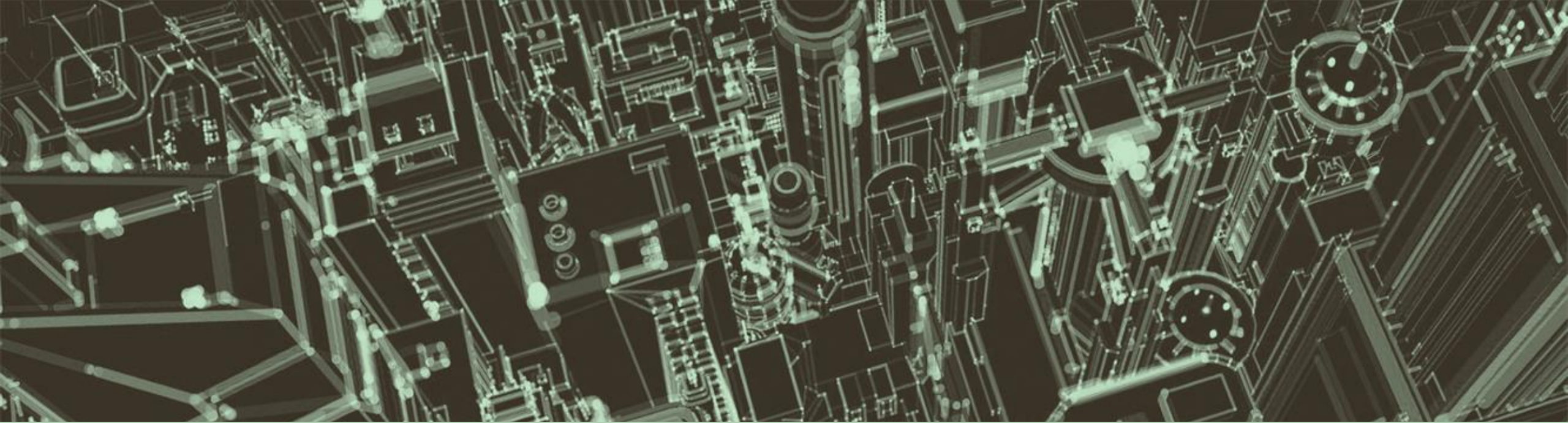
```
sudo apt install cron
sudo systemctl enable cron
crontab -e
```



```
manuo1@purbeurre-mo1: ~
GNU nano 4.8 /tmp/crontab.qAcJSI/crontab
#
# m h dom mon dow   command
#Every week at 12:00 on Monday run populatedb.sh
0 12 * * 1 /home/manuo1/bin/populatedb.sh

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line

84959 lignes en base pour les données alimentaires
(10029 aliments, 4641 categories, 70289 jointures)
2021-02-09 15:10:23.121219
--> populatedb terminé
84959 lignes en base pour les données alimentaires
(10029 aliments, 4641 categories, 70289 jointures)
2021-02-09 15:15:52.292940
--> populatedb terminé
84959 lignes en base pour les données alimentaires
(10029 aliments, 4641 categories, 70289 jointures)
2021-02-15 12:00:19.221279
--> populatedb terminé
85337 lignes en base pour les données alimentaires
(10043 aliments, 4694 categories, 70601 jointures)
manuo1@purbeurre-mo1:~/purbeurre-mo1$ exit
```



Difficultés et axes d'améliorations



Difficultés et axes d'améliorations

Difficulté

La principale difficulté a été pendant le travail sur l'intégration continue.

Le cours proposé pour ce projet détaillait beaucoup la démarche avec Travis et malgré une documentation plutôt complète il m'a fallu beaucoup de recherches pour réussir cette partie et notamment la connexion SSH pour effectuer le « pull » depuis le serveur.

Améliorations

Une des améliorations à apporter au projet sera de modifier la commande django pour le peuplement et les mises à jour de la base de données