



Groupe OC-Pizza

Projet OC-Pizza

Dossier d'exploitation

Version 1.0

Auteur

Sébastien Ziegler
Etudiant/développeur



TABLE DES MATIÈRES

1 -Versions	3
2 -Introduction	4
2.1 -Objet du document	4
2.2 -Références	4
3 -Pré-requis	5
3.1 -Système	5
3.1.1 -Serveur	5
3.1.1.1 -Caractéristiques techniques	5
3.2 -Bases de données	5
3.3 -Web-services	5
4 -Procédure de déploiement	6
4.1 -Déploiement	6
4.1.1 -Variables d'environnement du projet	7
4.1.2 -Variables d'environnement du serveur Gunicorn	7
4.1.3 -Configuration	7
4.1.3.1 -Fichier production.py	8
4.1.3.2 -Fichier oc_pizza.conf (config supervisor)	8
4.1.3.3 -Fichier oc-pizza (config Nginx)	8
4.1.4 -DataSource	9
5 -Procédure de démarrage / arrêt	10
5.1 -Base de données	10
5.3 -Serveur d'application Gunicorn via Supervisor	10
5.3 -Serveur web Nginx	10
6 -Procédure de mise à jour	11
6.1 -Application	11
7 -Supervision/Monitoring	12
7.1 -Supervision de l'application web	12
7.2 -Supervision de l'infrastructure	12
7.2 -Supervision de l'intégration continu sur Github	12
8 -Procédure de sauvegarde et restauration	13
8.1 -Sauvegarde de la base de données	13
8.2 -Restauration de la base de données	13



1. VERSIONS

Auteur	Date	Description	Version
Sébastien Ziegler	25/12/2019	Création du document	1.0



2. INTRODUCTION

2.1. Objet du document

Le présent document constitue le dossier d'exploitation de l'application OC-Pizza

L' objectif du document est de fournir les informations donc l'équipe technique d'exploitation à besoin pour assurer une exploitation en règle du système.

2.2. Références

Pour de plus amples informations, se référer :

1. **DCT - Projet OC-Pizza** : Dossier de conception technique de l'application.
2. **DCF - Projet OC-Pizza** : Dossier de conception fonctionnelle.
3. **PVL - Projet OC-Pizza** : Procès verbal de livraison.



3. PRÉ-REQUIS

3.1. Système

3.1.1. Serveur

Le serveur fonctionne avec le système d'exploitation Linux Debian 10 (Buster) et utilise NGINX, Supervisor et Gunicorn pour la production.

Il utilise les web services New Relic pour le monitoring de l'infrastructure et le SDK Sentry pour le monitoring applicatif.

Le backend utilise Django 2.2.4(LTS) dans un environnement virtuel Python 3.8.

Le front utilise JQuery 3.4.0, Bootstrap 4.4.0 , ReactJS 16.12.0 .

La base de données utilise MySQL 8.0.

3.1.1.1. Caractéristiques techniques

Le serveur dispose de 4Gb de RAM , CPU avec 4 coeurs et d'un stockage de 100Gb.

Il fonctionne avec le système d'exploitation Linux Debian 10 (Buster).

3.2. Bases de données

Le projet utilise le serveur de base de données MySQL 8.0.

Les paramètres de connexion du projet à la bases de données sont défini dans le fichier ["production.py"](#).

3.3. Web-services

Le projet utilise de le service web de géolocalisation "Google Maps" ainsi que le service de paiement bancaire "Visa".



4. PROCÉDURE DE DÉPLOIEMENT

4.1. Déploiement

Ci-dessous un aperçu de la procédure de déploiement décrite de manière très détaillée dans le fichier README.md à la racine du repository Github du projet.

Le projet nécessite l'installation de Nginx, Supervisor, Python3.8, pip, pipenv et MySQL serveur.
Se référer aux documentations respectives des services.

Cloner le repository Github dans un répertoire sur le serveur (ex: /home/OC-Pizza/)

```
git clone git@github.com:OC-Pizza/oc_pizza_project.git
```

Créer et entrer dans l'environnement virtuel à la racine du projet:

```
cd oc_pizza_project  
pipenv shell
```

Installer les requirements.

```
pipenv install -r requirements.txt
```

Quitter l'environnement virtuel.

```
exit
```

Installer la base de données:

Se connecter en tant qu'utilisateur root et créer un utilisateur "OC_Pizza_user" pour la base de donnée.

```
mysql -u root -p  
GRANT ALL PRIVILEGES ON *.* TO 'OC_Pizza_user'@'localhost' IDENTIFIED BY 'password';  
/q
```

Création de la base de données OC_Pizza_db avec les recettes.

Ouvrir le fichier "creation_OC_Pizza_db_recettes.backup.sql" dans la console mysql (ressources/fichiers_configuration).

```
mysql -u OC_Pizza_user -p < creation_OC_Pizza_db_recettes.backup.sql  
/q
```

Copier le fichier "production.py" dans le répertoire "settings" du projet et configurer le fichier tel que défini dans la section 4.1.3.1.

```
cp /ressources/fichiers_configuration/product.py ./settings/
```



Configurer **NGINX** et **Supervisor** avec, respectivement les fichiers [oc-pizza](#) et [oc_pizza.conf](#)

4.1.1. Variables d'environnement du projet

Voici les variables d'environnement à modifier sur le serveur dans le fichier "production.py" :

Nom	Obligatoire	Description
SECRET_KEY	oui	Clé secrète d'application.
DEBUG	oui	Doit être à "False" en mode production.
ALLOWED_HOSTS	oui	Adresse autorisé à se connecté ("ex: oc-pizza.fr ")
DATABASES		
NAME	oui	Nom de la base de données
USER	oui	Nom d'utilisateur ayant des droits sur la base de données
PASSWORD	oui	Mot de passe d'accès à la base de données.
sentry_sdk.init()		
dns	oui	adresse "https" avec la clé de dns fourni par Sentry

4.1.2. Variables d'environnement du serveur d'application Gunicorn.

Le serveur d'application Gunicorn doit être exécuté avec la variable d'environnement suivante définie au démarrage. Elle est nécessaire afin de récupérer la configuration de l'application en production:

```
export DJANGO_SETTINGS_MODULE='oc_pizza_project.settings.production'
```

Cette variable d'environnement est défini dans le [fichier de configuration](#) de Supervisor.

Il suffit de démarrer Supervisor qui prend en charge le démarrage du serveur d'application Gunicorn.

4.1.3. Configuration

Voici les différents fichiers de configuration :

- **production.py** : fichier de configuration de production.
- **oc_pizza.conf** : fichier de configuration de Supervisor.



- **oc-pizza** : fichier de configuration de NGINX

4.1.3.1. Fichier *production.py*

```
SECRET_KEY = 'Votre clé secrète' # Remplacez par votre clé secrète d'application
DEBUG = False
ALLOWED_HOSTS = ['oc-pizza.fr', 'www.oc-pizza.fr'] # Adresses autorisées à se connecter

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql', # Adaptateur MySQL
        'NAME': 'nom de la base de donnée', # Remplacez par le nom de notre base de données créée.
        'USER': 'nom utilisateur', # Remplacez par votre nom d'utilisateur de la base de données.
        'PASSWORD': 'mot de passe', # Remplacez par le mot de passe d'accès à la base de données.
        'HOST': '',
        'PORT': '5432',
    }
}

sentry_sdk.init(
    dsn="https://clé DNS de sentry", # Remplacez par la clé DNS fourni par Sentry
    integrations=[DjangoIntegration()],
    send_default_pii=True
)
```

4.1.3.2. Fichier *oc_pizza.conf* (/etc/supervisor/con.d/*)

```
[program:pur_beurre]
command=/home/OC-Pizza/.local/share/virtualenvs/oc_pizza_projet-CuTitLlg/bin/gunicorn
oc_pizza.wsgi:application

user = sebastien
directory = /home/OC-Pizza/webapps/oc_pizza_projet
autostart = true
autorestart = true
environment = DJANGO_SETTINGS_MODULE='oc_pizza_projet.settings.production'
```

4.1.3.3. Fichier *oc-pizza* (/etc/nginx/sites-available/*)

```
server {
    listen 443;
    server_name oc-pizza.fr www.oc-pizza.fr;
    ssl on;
    ssl_certificate /etc/ssl/private/certificats/oc-pizza.fr.crt;
    ssl_certificate_key /etc/ssl/private/oc-pizza.fr.key;

    root /home/OC-Pizza/webapps/oc_pizza_projet/;

    location /static {
        alias /home/OC-Pizza/webapps/oc_pizza_projet/oc_pizza/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;
        }
    }
}
```



4.1.4. **DataSources**

Un fichier de recette de pizza "recettes.sql" se trouve dans la le dossier "ressources/fichiers_configuration/".



5. PROCÉDURE DE DÉMARRAGE / ARRÊT

5.1. Base de données

Arrêt du service mysql

```
sudo service mysql stop
```

Démarrage du service mysql

```
sudo service mysql start
```

5.2. Server d'application Gunicorn via Supervisor

Supervisor prend en charge le démarrage et l'arrêt du serveur d'application **Gunicorn**.

Arrêt du service Supervisor

```
sudo service supervisor stop
```

Démarrage du service Supervisor

```
sudo service supervisor start
```

5.3. Server web Nginx

Supervisor prend en charge le démarrage et l'arrêt du serveur d'application **Gunicorn**.

Arrêt du service Supervisor

```
sudo service nginx stop
```

Démarrage du service Supervisor

```
sudo service nginx start
```

6. PROCÉDURE DE MISE À JOUR

6.1. Application

Après une mise à jour du repository Github, il suffit de faire un faire une pull du repository Git du serveur depuis la racine du projet, puis de redémarrer le serveur d'application Gunicorn via Supervisor.

```
cd ./oc_pizza_project
git pull origin master
sudo service supervisor reload
```

Ce processus de mise à jour peut être automatisé dans une tâche cron selon un périodicité définie.



7. SUPERVISION/MONITORING

7.1. Supervision de l'application web

La supervision et monitoring de la partie applicative se fait via le dashboard de [Sentry](#).

7.2. Supervision de l'infrastructure

La supervision et monitoring de l'infrastructure se fait via le dashboard du service [New Relic](#).

7.3. Supervision de l'intégration continu sur Github.

La supervision et monitoring de l'intégration continu du repository Github se fait via le dashboard du service d'intégration continu [Travis-ci](#).



8. PROCÉDURE DE SAUVEGARDE ET RESTAURATION

8.1. Sauvegarde de la base de données

La mise à jour de la base de données, se fait avec la commande:

```
mysqldump -u OC_Pizza_user -p OC_Pizza_db > OC_Pizza_db.backup.sql
```

Ce processus de sauvegarde peut être automatisé dans une tâche cron selon une périodicité définie.

8.2. Restauration de la base de données

La restauration de la base de données via le fichier backup, se fait avec la commande:

```
mysql -u OC_Pizza_user -p < OC_Pizza_db.backup.sql
```