

# OC Pizza

## Système de gestion de pizzeria en ligne

Dossier d'exploitation

Version 1.0

**Auteur**

Clémence Robin  
*Analyste-programmeur*

# TABLE DES MATIÈRES

<b>1 - Versions.....</b>	<b>3</b>
<b>2 - Introduction.....</b>	<b>4</b>
2.1 - Objet du document.....	4
2.2 - Références.....	4
<b>3 - Pré-requis.....</b>	<b>5</b>
3.1 - Système.....	5
3.2 - Bases de données.....	5
3.3 - Web-services.....	5
<b>4 - Procédure de déploiement.....</b>	<b>6</b>
4.1 - Déploiement de l'application web.....	6
4.1.1 - Configuration de l'espace serveur de l'application.....	6
4.1.2 - Configururion de Unicorn.....	8
4.1.3 - Configuration de Nginx.....	9
4.1.4 - Configuration du nom de domaine.....	9
4.2 - Déploiement de la base de données.....	10
4.2.1 - Configuration de l'espace serveur de la base de données.....	10
<b>5 - Procédure de démarrage / arrêt.....</b>	<b>11</b>
5.1 - Base de données.....	11
5.2 - Application web.....	11
<b>6 - Procédure de mise à jour.....</b>	<b>12</b>
6.1 - Base de données.....	12
6.2 - Application web.....	12
<b>7 - Supervision/Monitoring.....</b>	<b>13</b>
7.1 - Supervision de l'application web.....	13
7.2 - Supervision de la base de données.....	13
<b>8 - Procédure de sauvegarde et restauration.....</b>	<b>14</b>
8.1 - Sauvegarde et restauration de l'application web.....	14
8.2 - Sauvergarde et restauration de la base de données.....	14
<b>9 - Glossaire.....</b>	<b>15</b>

# 1 - VERSIONS

Auteur	Date	Description	Version
Clémence R.	28/08/2020	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.

Il rassemble les informations dont l'équipe technique d'OC Pizza a besoin pour pouvoir assurer l'exploitation du système ainsi que les procédures à appliquer lorsqu'un problème survient.

### 2.2 - Références

Pour de plus amples informations, se référer :

1. **DCT – 1.0** : Dossier de conception technique de l'application
2. **DCF – 1.0** : Dossier de conception fonctionnelle de l'application

## 3 - PRÉ-REQUIS

### 3.1 - Système

L'application OC Pizza ainsi que la base de données sont hébergées sur des machines virtuelles Digital Ocean .

Le nom de domaine de l'application est le suivant : ocpizza.fr

### 3.2 - Bases de données

Les bases de données et schémas suivants doivent être accessibles et à jour :

- **PostgreSQL** : version 12

### 3.3 - Web-services

Les web services suivants doivent être accessibles et à jour :

- **Paypal API**
  - Clé secrète : xrtsa=v@w&katuwxq&q@z^og^#4o4m!rz1mns#-nj&w=%+ddnw

# 4 - PROCÉDURE DE DÉPLOIEMENT

## 4.1 - Déploiement de l'application web

### 4.1.1 - Configuration de l'espace serveur de l'application

#### 4.1.1.1 - Serveur dédié à l'application web

L'application est déployée sur un serveur loué chez Digital Ocean.

#### 4.1.1.2 - Système d'exploitation

Le système d'exploitation Debian sera déjà installé sur le serveur par Digital Ocean.

#### 4.1.1.3 - Accès à distance

Deux clés SSH devront être créées: la clé privée et la clé publique. Elles permettront de se connecter au serveur distant.

#### 4.1.1.4 - Librairies à installer avant l'application

Certaines librairies devront être installées afin de déployer l'application OC Pizza :

```
apt install git
apt install python3
apt install python3-pip
apt install python3-dev
apt install libpq-dev
apt install nginx
```

#### 4.1.1.5 - Téléchargement de l'application OC Pizza

Après avoir configuré Git pour obtenir les droits d'accès au *repository* de l'application, il faut télécharger l'application :

```
git clone https://github.com/clemenceclaireR/documentation_restaurant
```

#### 4.1.1.6 - Installation des librairies de l'application

Toutes les librairies sont installées par la commande :

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

Il faut pour cela se trouver au même niveau d'aborescence que le fichier *requirements.txt*.

#### 4.1.1.7 - Lancement de l'application

Spécifier le fichier de configuration à utiliser (ici, production) en rentrant la commande suivante dans le terminal :

```
export DJANGO_SETTINGS_MODULE=oc_pizza.settings.prod
```

Il est conseillé d'ajouter cette commande au fichier *.bashrc* du répertoire *home* afin d'éviter réitérer l'entrée de cette commande lorsqu'un nouveau terminal est ouvert.

Pour lancer l'application avec son environnement de production, rentrer la commande suivante dans le terminal :

```
python manage.py runserver
```

#### 4.1.1.8 - Variables d'environnement

Voici les variables d'environnement reconnues par les batches de l'application OC Pizza :

Nom	Obligatoire	Description
SECRET_KEY	oui	Clé secrète de l'application Django
ENV	oui	Définit le statut de l'environnement (développement ou production)
DEBUG	oui	Activation ou non du mode de débogage
ALLOWED_HOSTS	oui	Liste les noms de domaines accessibles pour l'application
STATIC_ROOT	oui	Dossier où sont collectés tous les fichiers statiques
STATIC_URL	oui	URL utilisée pour se référer aux fichiers statiques se trouvant dans STATIC_ROOT

### 4.1.2 - Configuration de Gunicorn

#### 4.1.2.1 - Configuration du fichier socket de Gunicorn

Créer le fichier *gunicorn.socket*, dans */etc/systemd/system*.

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

Le remplir avec ces informations :

[Unit]

```
Description=unicorn socket
[Socket]
ListenStream=/run/unicorn.sock
[Install ]
WantedBy=sockets.target
```

#### **4.1.2.2 - Configuration du fichier services de Unicorn**

Créer le fichier de service de Unicorn à l'emplacement /etc/systemd/system :

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

Le remplir avec ces informations :

```
[Unit]
Description= unicorn daemon
Requires= unicorn.socket
After= network.target

[Service]
User=
Group= www-data
WorkingDirectory= /home/user/oc_pizza
ExecStart=/home///bin/unicorn \
    --access-logfile - \
    --workers 3 \
    --bind unix:/run/unicorn.sock \
    .wsgi:application

[Install]
WantedBy=multi-user.target
```

#### **4.1.2.3 - Lancement du socket unicorn**

```
systemctl start unicorn.socket
```

#### **4.1.2.4 - Activation de unicorn**

```
systemctl enable unicorn.socket
```

### **4.1.3 - Configuration de Nginx**

Dans *home/user/oc\_pizza*, rentrer la commande suivante dans le terminal pour créer le fichier suivant :



```
nano /etc/nginx/sites-available/oc_pizza
```

Insérer le code suivant dans le fichier :

```
server {  
    listen 80;  
    server name xxx.xxx.xxx.xxx ocpizza.fr www.ocpizza.fr;  
    location = /favicon.ico { access_log off; log_not_found off; }  
    location /static/ {  
        root /home/ubuntu/myproject/weather;  
    }  
    location / {  
        include proxy_params;  
        proxy_pass http://unix:/run/gunicorn.sock;  
    }  
}
```

Lier le fichier au dossier sites-enabled de Nginx :

```
ln -s /etc/nginx/sites-available/oc_pizza /etc/nginx/sites-enabled
```

#### **4.1.4 - Configuration du nom de domaine**

Dans `home/user/oc_pizza/oc_pizza/settings/prod.py`, éditer la variable `ALLOWED_HOSTS` afin d'y inclure l'adresse IP du nouveau domaine :

```
ALLOWED_HOSTS = ['XXX,XXX,XXX,XXX', 'ocpizza.fr', 'www.ocpizza.com']
```

## **4.2 - Déploiement de la base de données**

### **4.2.1 - Configuration de l'espace serveur de la base de données**

#### **4.2.1.1 - Serveur dédié à la base de données**

La base de donnée est déployée sur un serveur loué chez Digital Ocean.

#### **4.2.1.2 - Accès au serveur de base de données**

Le serveur de base de données sera accessible avec l'API ou l'interface utilisateur mise à disposition par [Digital Ocean](#).

#### **4.2.1.3 - Configuration de la base de données**

*Note : Digital Ocean fournit un [guide](#) pour configurer sa base de données.*

L'interface graphique permet de créer et configurer directement la base de données ainsi que ses utilisateurs.

#### **4.2.1.4 - Insertion des données dans la base de données**

Une fois la base de données créée, il faut insérer le fichier SQL contenant les tables ainsi que leurs données de base nécessaires au bon fonctionnement de l'application.

# 5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

## 5.1 - Base de données

La base de donnée est accessible dès qu'elle est créée et configurée sur le serveur Digital Ocean.

## 5.2 - Application web

L'application se lance avec la commande suivante dans le terminal, à la racine du projet :

```
python manage.py runserver
```

Pour l'arrêter, il suffit de presser les touches CTRL+C ou CTRL + BREAK/PAUSE sur le clavier.

## 6 - PROCÉDURE DE MISE À JOUR

### 6.1 - Base de données

Digital Ocean permet de planifier les dates et heures des mises à jours de la base de données via son interface utilisateur.

### 6.2 - Application web

Afin d'éviter toute incompatibilité, les versions des librairies Django spécifiées dans le fichier *requirements.txt* doivent être mises à jour manuellement.

# 7 - SUPERVISION/MONITORING

## 7.1 - Supervision de l'application web

Digital Ocean met à disposition des outils de [monitoring](#) dans l'interface utilisateur, sans coûts additionnels. Ils sont intégrés par défaut :

- Un tableau de bord donnant des métriques sur le CPU, la mémoire et l'utilisation du disque ainsi que l'espace disponible
- Création d'alertes : il est possible de définir ses propres alertes sur les métriques de son choix et de recevoir une notification quand une métrique dépasse la limite définie

Il est possible de personnaliser les outils de monitoring par l'API.

## 7.2 - Supervision de la base de données

Digital Ocean met également à disposition des [outils](#) de métriques et d'alertes pour superviser le bon fonctionnement de la base de données. Il est là aussi possible de définir ses propres alertes sur les métriques de son choix et d'être notifié en cas de dépassement des dites métriques.

D'autres fonctionnalités sont accessibles avec une base de données PostgreSQL:

- connexions à la base de données
- taux de réussite de la mémoire cache
- rythme de création, modification, suppression et extraction de données

# 8 - PROCÉDURE DE SAUVEGARDE ET RESTAURATION

## 8.1 - Sauvegarde et restauration de l'application web

Digital Ocean génère automatiquement des [sauvegardes](#) toutes les semaines pour un coût supplémentaire de 20%. Ils sont conservés pendant quatre semaines.

Il est possible de consulter la date de procédure de *backup*, déterminée par Digital Ocean, dans le panneau de configuration mis en place dans l'interface utilisateur.

## 8.2 - Sauvegarde et restauration de la base de données

Le serveur Digital Ocean fait automatiquement des sauvegardes quotidiennes de la base de données, conservées pendant sept jours.

Pour restaurer une sauvegarde de la base de données, il faut se rendre dans la section *Overview* de sa base de données et cliquer sur le bouton *Action* à droite de la fenêtre pour activer le menu déroulant. Cliquer sur l'option *Restore from backup*.

**example-database**  
in **example-project** / 1 GB RAM / 1vCPU / 10 GB Disk / Primary only / BLR1 - PostgreSQL 12

**Overview** Insights Logs & Queries Users & Databases Connection Pools Settings

**DATABASE CLUSTER TOTAL COST** [How does pricing work?](#)

**\$15** monthly rate  
This amount is prorated, and does not reflect your month to date usage.

**TRUSTED SOURCES** [Edit Sources](#)

**CONNECTION DETAILS**

**Public network** Private network Conf

**username** = doadmin  
**password** = \*\*\*\*\* [show](#)  
**host** = example-database-do-user-6607900-0.a.db.ondigitalocean.com

**Actions** ^

- Connection details
- Add a read-only node
- Fork database cluster
- Restore from backup**
- Edit tags
- Move to...
- Destroy

## 9 - GLOSSAIRE

<b>repository</b>	dépôt
<b>backup</b>	sauvegarde