

# OC Pizza

## Application OC Pizza

Dossier d'exploitation

Version 1.0

### **Auteur**

Gabrielle Barboteau  
*Développeuse Junior*

### A Faire :

- Remplacer manuellement les valeurs entre <>
- Les valeurs entre {} sont à renseigner dans les propriétés du document

# 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 et serveur.....	5
3.2 -Bases de données.....	5
3.3 -Web-services.....	5
<b>4 -Procédure de déploiement.....</b>	<b>6</b>
4.1 -Configuration du serveur.....	6
4.1.1 -Configuration d'un utilisateur du serveur.....	6
4.1.2 -Installation de Python.....	6
4.1.3 -Installation de proftpd.....	6
4.1.4 -Installation de nginx.....	6
4.1.5 -Installation et configuration de la base de donnée.....	6
4.2 -Déploiement de l'application Web.....	7
4.2.1 -Upload sur le serveur.....	7
4.2.2 -Variables d'environnement.....	7
4.2.3 -Configuration.....	7
4.2.3.1 -Fichier requirements.txt.....	7
4.2.3.2 -Fichier setting.py.....	7
4.2.3.3 -Fichier migrate.py.....	7
4.2.4 -Paramétrage de nginx.....	8
4.2.5 -Installation et paramétrage de Gunicorn.....	8
4.2.6 -Installation et paramétrage de supervisor.....	8
<b>5 -Procédure de démarrage / arrêt.....</b>	<b>10</b>
<b>6 -Procédure de mise à jour.....</b>	<b>11</b>
6.1 -Base de données.....	11
6.2 -Application web.....	11
<b>7 -Supervision/Monitoring.....</b>	<b>12</b>

# 1 - VERSIONS

Auteur	Date	Description	Version
Gabrielle Barboteau	20/08/92	Création du document	01/01/00

## 2 - INTRODUCTION

### 2.1 - Objet du document

Le présent document constitue le dossier d'exploitation de l'application OC Pizza, son objectif étant de permettre à l'équipe technique de notre client OC Pizza d'utiliser l'application que nous avons développée.

### 2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants:

1. **PDOCPizza\_01\_fonctionnelle** : Dossier de conception fonctionnelle de l'application
2. **PDOCPizza\_02\_technique** : Dossier d'exploitation de l'application

## 3 - PRÉ-REQUIS

### 3.1 - Système et serveur

L'hébergement de la solution se fera sur un serveur dédié, que nous vous préconisons de choisir un chez OVH ou Digital Ocean, avec 1To de stockage et 8Go de RAM minimum.

Le serveur fonctionnera sous Ubuntu 18.04 (la dernière version LTS).

Les éléments suivants seront également nécessaire pour faire fonctionner l'application :

- **Python** : version 3.7
- **Django** : version 2.2
- **Gunicorn** : version 20
- **MySQL-connector** : version 8.0
- **Nginx** : version 1.17
- **Supervisor** : version 4.1
- **Proftpd** : version 1.3
- **Raven** : version 6.10

### 3.2 - Bases de données

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

- **MySQL** : version 8.0

### 3.3 - Web-services

Les web services suivants doivent être accessibles et à jour :

- **API de Google Maps** : notez cette API est payante au-delà de 2500 requêtes par jour (0,50\$ par 1000 requêtes supplémentaires).
- **Paypal** : Nous permettra de gérer les paiements en ligne sécurisés.

# 4 - PROCÉDURE DE DÉPLOIEMENT

## 4.1 - Configuration du serveur

### 4.1.1 - Configuration d'un utilisateur du serveur

Pour correctement fonctionner, le serveur doit bien évidemment avoir un utilisateur. C'est dans son répertoire que l'application sera uploadée, il est donc nécessaire de le créer avant toute chose. Nous le nommerons ocpizzauser par souvis de clareté. La démarche est la suivante :

- **# adduser ocpizzauser**
- **# passwd -a ocpizzauser**

### 4.1.2 - Installation de Python

La version 3.7 de Python nécessite une installation particulière sur Ubuntu 18.04. Elle se fait ainsi :

- **# sudo apt update**
- **# sudo apt install software-properties-common**
- **# sudo add-apt-repository ppa:deadsnakes/ppa**
- **# sudo apt update**
- **# sudo apt install python3.7**

### 4.1.3 - Installation de proftpd

Afin d'uploader les fichiers de l'application (mais pas que) sur le serveur, nous installerons proftpd, qui est un serveur FTP simple et puissant. Il s'installe avec la commande suivante :

- **# sudo apt-get install proftpd**

L'application peut ensuite être uploadée sur serveur via un client FTP comme FileZilla.

### 4.1.4 - Installation de nginx

Nginx est le serveur HTTP que nous utiliserons avec notre application. Il s'installe avec la commande suivante :

- **# sudo apt-get install nginx**

Nous le configurerons plus tard.

### 4.1.5 - Installation et configuration de la base de donnée

Installez MySQL avec la commande :

- **apt-get install mysql-server**

Créez ensuite la base de donnée :

- **# mysql -u ocpizza -p**
- **CREATE DATABASE ocpizza**

## 4.2 - Déploiement de l'application Web

### 4.2.1 - Upload sur le serveur

Grâce à proftpd, il vous sera possible d'uploader l'application avec un glisser-déposer. Faites attention à la mettre dans le bon dossier, qui est /home/ocpizzauser/.

Pensez aussi à créer un environnement virtuel afin d'y manipuler l'application (vous pouvez installer virtualenv avec la commande **\$ pip install virtualenv**).

### 4.2.2 - Variables d'environnement

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

Nom	Obligatoire	Description
SECRET_KEY	oui	Permet à Django de sécuriser l'application
GOOGLE_KEY	oui	Clé nous permettant de communiquer avec l'API de Google Maps

Définissez les variables d'environnement en modifiant le fichier settings.py sur le serveur (en utilisant Vim).

### 4.2.3 - Configuration

Voici les différents fichiers de configuration :

- **requirements.txt** : fichier de téléchargement des dépendances de l'application
- **setting.py** : fichier de configuration de l'application
- **migrate.py** : fichier de migration de base de donnée

#### 4.2.3.1 - Fichier requirements.txt

Lancez ce fichier dans virtualenv avec la commande suivante :

- **\$ pip install -r requirements.txt**

#### 4.2.3.2 - Fichier setting.py

Modifiez ce fichier pour y inclure votre SECRET\_KEY et votre GOOGLE\_KEY.

#### 4.2.3.3 - Fichier migrate.py

Tapez la commande suivante :

- **# ./ocpizza/manage.py migrate**

#### 4.2.4 - Paramétrage de nginx

L'application étant uploadée, nous pouvons paramétrer nginx.

IL est d'abord nécessaire de créer un fichier de configuration, de cette manière, en vous plaçant dans le répertoire /etc/nginx :

- **# sudo touch sites-available/ocpizza**

Puis, en retournant à la racine du serveur

- **# sudo ln -s /etc/nginx/sites-available/ocpizza /etc/nginx/sites-enabled**

La dernière étape est d'ouvrir le document que vous avez créé, et d'y taper :

```
server {  
  
    listen 80; server_name ocpizza.fr;  
    root /home/ocpizzauser/ocpizza/  
  
    location / {  
        proxy_set_header Host $http_host;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_redirect off;  
        proxy_pass http://127.0.0.1:8000;  
    }  
}
```

Notez que le nom utilisé pour `server_name` est le nom de domaine de votre choix ou l'adresse du serveur.

#### 4.2.5 - Installation et paramétrage de Gunicorn

Gunicorn devrait déjà être installé car présent dans le fichier **requirements.txt**. Si ce n'est pas le cas, lancez la commande suivante :

- **# pip install gunicorn**

Lancez ensuite gunicorn avec la commande suivante :

- **# gunicorn ocpizza.wsgi:application**

#### 4.2.6 - Installation et paramétrage de supervisor

Installez supervisor avec la commande suivante :

- **# pip install supervisor**

Allez ensuite dans /etc/supervisor/conf.d, et créez un fichier de configuration nommé **ocpizza-gunicorn.conf**, et modifiez-le ainsi :



```
[program:ocpizza-gunicorn]
command = /home/ocpizza/env/bin/gunicorn ocpizza:wsgi.application
user = ocpizza-gunicorn
directory = /home/ocpizza/ocpizza
autorestart = true
autorestart = true
```

Enfin, lancez supervisor, en vous mettant dans le dossier où il se situe. Cela aura pour effet de lancer votre application, et elle devrait être ensuite accessible :

- **# sudo supervisorctl reread**
- **# sudo supervisorctl update**

## 5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

L'application est lancée avec cette commande de supervisor :

- **# sudo supervisorctl start ocpizza-gunicorn**

Et s'arrête avec celle-ci :

- **# sudo supervisorctl stop ocpizza-gunicorn**

# 6 - PROCÉDURE DE MISE À JOUR

## 6.1 - Base de données

Les mises à jour seront effectuées sur le serveur, avec la commande :

- **# sudo apt-get update**
- **# sudo apt-get install mysql-server**

## 6.2 - Application web

Vous mettrez à jour l'application en remplaçant les fichiers nécessaires grâce au client FTP, puis en relançant supervisor de cette manière :

- **# sudo supervisorctl reread**
- **# sudo supervisorctl update**

## 7 - SUPERVISION/MONITORING

Nous utiliserons Sentry, un tableau de bord permettant de visualiser le statut d'une application Django. Après vous être créé un compte sur leur site, installez la librairie Raven :

- **\$ pip install Raven**

Le fichier de configuration que nous vous avons fourni prend déjà en compte l'installation de Raven, vous devrez juste changer votre clé dans RAVEN\_CONFIG:

```
RAVEN_CONFIG = {
    'dsn': 'https://somethingverylong@sentry.io/XXXXXXX', # INSEREZ ICI
                                                         VOTRE CLE

    # If you are using git, you can also automatically configure the
    # release based on the git info.
    'release': raven.fetch_git_sha(os.path.dirname(os.pardir)),
}
```

Relancez supervisor pour que les changements soient pris en compte.