



It Consulting  
&  
Development

# **OC PIZZA**

## **OC\_pizza**

Dossier d'exploitation

Version 1.0

**Auteur**  
Jérémy Guyot

# 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
3.1.1 - Serveur de Base de données.....	5
3.1.1.1 - Caractéristiques techniques.....	5
3.1.2 - Serveur Web.....	5
3.1.2.1 - Caractéristiques techniques.....	5
3.1.3 - Serveur de Batches.....	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 - Artefacts.....	6
4.1.2 - Environnement de l'application web.....	6
4.1.2.1 - Variables d'environnement.....	6
4.1.3 - Répertoire de configuration applicatif.....	7
4.1.4 - DataSources.....	7
4.1.5 - Vérifications.....	7
<b>5 - Procédure de démarrage / arrêt.....</b>	<b>8</b>
5.1 - Base de données.....	8
5.2 - Application web.....	8
<b>6 - Procédure de mise à jour.....</b>	<b>9</b>
6.1 - Base de données.....	9
6.2 - Application web.....	9
<b>7 - Supervision/Monitoring.....</b>	<b>10</b>
7.1 - Supervision de l'application web.....	10
<b>8 - Procédure de sauvegarde et restauration.....</b>	<b>11</b>
<b>9 - Glossaire.....</b>	<b>12</b>

# 1 - VERSIONS

Auteur	Date	Description	Version
GUYOT Jeremy	06/07/2020	Creation du document	1.0

## 2 - INTRODUCTION

### 2.1 - Objet du document

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

Ce document représente la mise en production des systèmes développés par it consulting & development.

### 2.2 - Références

Pour de plus amples informations, se référer :

1. **P9 \_ DCT - 1.0** : Dossier de conception technique de l'application.
2. **P9 \_ DE - 1.0** : Dossier d'exploitation de l'application.
3. **P9 \_ DCF - 1.0** : Dossier de conception fonctionnelle de l'application.
4. **P9 \_ PVL – 1.0** : Dossier de PV Livraison.

## 3 - PRÉ-REQUIS

### 3.1 - Système

#### 3.1.1 - Serveur de Base de données

Le serveur de base de données est hébergé par Digital Ocean avec Ubuntu 18.04.3 et MySQL 8.0.19.

##### 3.1.1.1 - Caractéristiques techniques

On va utiliser Digital Ocean sur lequel il y aura 1 vCPU (private CPU), 1 Gb Mémoire(RAM), 1 SSD 25 Gb, 1Tb transfer et le datacenter sera à Londres.

#### 3.1.2 - Serveur Web

Le serveur virtuel hébergeant l'application web est fourni par Digital Ocean avec Ubuntu 18.04.3 et Nginx 1.18.0 Gunicorn 20.0.4 et Supervisor 4.2.0.

##### 3.1.2.1 - Caractéristiques techniques

On va utiliser Digital Ocean sur lequel il y aura 1 vCPU (Shared CPU), 1 Gb Mémoire(RAM), 1 SSD 25 Gb, 1Tb transfer et le datacenter sera à Londres

#### 3.1.3 - Serveur de Batches

Pour le serveur de Batches, ça sera le même que pour le Web, il va nous être utile pour les tâches Cron.

### 3.2 - Bases de données

La base de données et gérer par les modèles de notre application Django et son diagramme de classe est disponible dans le dossier de conception technique.

- **P9 \_ DCT – 1.0** : version 1.0

### 3.3 - Web-services

On utilise les liens suivants :

- **Paypal** : <https://developer.paypal.com/docs/api/overview/#>
- **Google Maps (geocoding)** : <https://developers.google.com/maps/documentation/geocoding/start?hl=fr>

# 4 - PROCÉDURE DE DÉPLOIEMENT

## 4.1 - Déploiement de l'Application Web

### 4.1.1 - Artefacts

Les fichiers de l'application Web se trouvent dans un dépôt de code sur GitHub.

Il faudra le cloner avec la commande :

**git clone [https://github.com/OC\\_pizza/OC\\_pizza.git](https://github.com/OC_pizza/OC_pizza.git)**

Une fois qu'on a cloné le projet, il faudra se rendre dans le répertoire du projet et faire la commande :

**git checkout production**

Vu que vous voudrez accéder au serveur de production. Ensuite il faudra entrer les identifiants suivants :

**ssh ocpizza@206.189.117.65**

Ce qui va vous permettre, d'avoir accès au serveur.

### 4.1.2 - Environnement de l'application web

#### 4.1.2.1 - Variables d'environnement

Pour ce qui concerne les variables d'environnement, il vous faudra remplacer ces valeurs (XXXXX)

EMAIL_USER=XXXXX	C'est l'e-mail qui va vous servir de messagerie
EMAIL_PASSWORD=XXXXX	C'est le mot de passe que vous devez créer
DB_USER=XXXXX	Sert à se connecter à la Data Base
DB_PASSWORD=XXXXX	Le mot de passe qui sera liée au compte de la DB

Vous pourrez remplacer ces valeurs dans le fichier : .env

### 4.1.3 - Répertoire de configuration applicatif

Pour la partie settings, on l'a découpée en 3 parties.

1. **\_\_init.py\_\_** : Elle va regrouper la structure du projet, donc ce qui concerne :  
BAS\_DIR, SECRET\_KEY, INSTALLED\_APPS, MIDDLEWARE, DATABASES, etc.
2. **production.py** : Elle va regrouper la production, donc ce qui concerne import de Sentry, SECRET\_KEY, DATABASES, ce qui concerne l'email et l'API de sentry.
3. **travis.py** : Elle va regrouper uniquement le DATABASES avec comme utilisateur ADMIN « admin ».

### 4.1.4 - DataSources

La base de données est configurée par les fichiers de configuration de notre application web.

Les sources des données pourront être chargées par une commande Django, qui sera :

**python manage.py initdata**

### 4.1.5 - Vérifications

Les tests ont été rédigé pour tester le bon fonctionnement de l'application et on peut lancer les tests avec la commande : **python manage.py test deployment**

Si on veut vérifier que tout fonctionne correctement.

# 5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

## 5.1 - Base de données

Pour ce qui concerne la base de données, elle est gérée par MySQL. Il faudra d'abord vous connecter, puis vous pourrez démarrer ou arrêter la base de données avec ces commandes ci-dessous.

Pour démarrer la base de données, il faudra utiliser les commandes :

**mysql -h localhost -u root** (puis entrer le mot de passe : **ocpizza**)

**sudo systemctl start mysql**

Pour redémarrer la base de données, il faudra utiliser la commande :

**sudo systemctl restart mysql**

Pour arrêter la base de données, il faudra utiliser la commande :

**sudo systemctl stop mysql**

## 5.2 - Application web

L'application web est gérée par Supervisor.

Pour démarrer l'application web, il faudra utiliser la commande :

**Sudo supervisorctl start OC\_pizza**

Pour arrêter l'application web, il faudra utiliser la commande :

**Sudo supervisorctl stop OC\_pizza**



# 6 - PROCÉDURE DE MISE À JOUR

## 6.1 - Base de données

Pour ce qui concerne la base de données, elle est gérée par MySQL. Il faudra d'abord vous connecter, puis vous pourrez lancer la mises à jour avec ces commandes ci-dessous.

Pour démarrer la base de données, il faudra utiliser les commandes :

**mysql -h localhost -u root** (puis entrer le mot de passe : **ocpizza**)

**sudo systemctl start mysql**

Pour lancer votre mises à jour vous devrez utiliser cette commande :

**sudo systemctl updatedb mysql**

## 6.2 - Application web

Pour mettre à jour l'application web, il faudra utiliser la commande :

**python manage.py updateweb**

# 7 - SUPERVISION/MONITORING

## 7.1 - Supervision de l'application web

Afin de tester que l'application web est toujours fonctionnelle, il faudra aller sur **Sentry**. <https://docs.sentry.io/>

Sentry, permet de tester directement le site en production et en continu, pour pouvoir vérifier et ou intervenir au besoin. Il va nous déclarer toutes les anomalies qu'il pourra rencontrer lors de l'utilisation du site, par les utilisateurs ou par ceux qui l'utiliseront.

On a aussi **Digital Ocean** <https://www.digitalocean.com/docs/> qui permet de vérifier tous les ajouts qu'on peut faire sur le site de production. On aura des graphiques comme le CPU, Load, Memory, Disk I/O, Disk Usage et la Public Bandwidth, qui vont nous permettre de pouvoir observer l'utilisation du site en production.

## 8 - PROCÉDURE DE SAUVEGARDE ET RESTAURATION

Pour ce qui concerne la sauvegarde et la restauration, nous avons prévu des sauvegardes automatiques tous les 7 jours, donc une fois par semaine.

On pourra voir cela ensemble, jour et heure à votre convenance, on aura aussi une commande si vous avez besoin de faire une sauvegarde manuelle.

Et Donc une procédure de restauration, qui va nous permettre de faire un backup.

Vois-ci les commandes utiliser :

**python manage.py backup** ( sert à faire une restauration )

**python manage.py loadbackup** ( sert à faire une sauvegarde manuelle )

## 9 - GLOSSAIRE

<b>Base de données</b>	La base de données est au centre des dispositifs informatiques de collecte, mise en forme, stockage et utilisation d'informations.
<b>Variables d'environnement</b>	les variables d'environnement sont des variables dynamiques utilisées par les différents processus d'un système d'exploitation ( <a href="#">Windows</a> , <a href="#">Unix</a> ...). Elles servent à communiquer des informations entre les programmes qui ne se trouvent pas sur la même ligne hiérarchique, et qui ont donc besoin d'une convention pour communiquer mutuellement leurs choix.