

## OC Pizza

# Réalisation d'un système de gestion pour un ensemble de pizzerias

Dossier d'exploitation

Version 1.0

**Auteur**

Guillaume OJARDIAS  
*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
<b>4 - Procédure de déploiement.....</b>	<b>7</b>
4.1 - Déploiement de l'application web.....	7
<b>5 - Procédure de démarrage/arrêt.....</b>	<b>9</b>
5.1 - Serveur d'application.....	9
5.2 - Serveur Web.....	9
5.3 - Base de données.....	9
<b>6 - Procédure de mise à jour.....</b>	<b>10</b>
6.1 - Mise à jour de l'application.....	10
6.2 - Mise à jour de la base de données.....	10
<b>7 - Supervision et monitoring.....</b>	<b>11</b>
7.1 - Surveillance du serveur.....	11
7.2 - Surveillance de l'application.....	12
<b>8 - Procédure de sauvegarde et restauration.....</b>	<b>13</b>
8.1 - Sauvegarde.....	13
8.2 - Restauration.....	13
<b>9 - Glossaire.....</b>	<b>14</b>

# 1 - Versions

Auteur	Date	Description	Version
GO	07/11/2020	Création du document	1.0

## 2 - Introduction

### 2.1 - Objet du document

Le présent document constitue le dossier d'exploitation de l'application pour le système de gestion d'un ensemble de pizzerias.

L'objectif de ce document est de définir les différentes caractéristiques du système nécessaire à l'exécution de l'application.

Par ailleurs, ce document permet de détailler les différentes étapes de déploiement de la dite application, la procédure de mise à jour, de sauvegarde et de restauration.

### 2.2 - Références

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

1. Projet 9 - Dossier de conception fonctionnelle.
2. Projet 9 - Dossier de conception technique.
3. Projet 9 - PV de livraison.

## 3 - Pré-requis

### 3.1 - Système

L'ensemble de l'application utilisera un serveur ayant les caractéristiques minimales suivantes :

- > Système d'exploitation : Linux / **Ubuntu** version **20.04**.
- > Disque SSD de **25 GB**.
- > **8 GB** de mémoire.
- > **2** processeurs.

Sur ce serveur, un compte utilisateur **adminOcPizza** sera créé. Ce compte sera utilisé pour la mise en place et la maintenance de l'application.

Pour assurer le déploiement de l'application sur ce serveur, les packages suivants seront requis :

- > **GIT** en version **2.28.0** minimum.
- > **Python** en version **3.8** minimum
  - > **Virtualenv** en version **20.0.35** minimum
- > **Certbot** en version **0.40.0** minimum
- > **Supervisord** en version **4.1.0** minimum.

Le fournisseur de ce serveur sera Digital Ocean : <https://digitalocean.com>.

### 3.1.1 - Serveur de base de données

Le serveur de base de données aura les caractéristiques suivantes :

- **PostgreSQL**.
- Version **13** minimum.

Ce serveur sera configuré de la manière suivante :

- Une base de données nommée **OcPizza**.
- L'utilisateur **adminOcPizza** aura un droit d'écriture/lecture sur cette base de données.

### 3.1.2 - Serveur WEB

Le serveur WEB aura les caractéristiques suivantes :

- Serveur **NGINX**.
- Version **1.20** minimum.

**UFW** devra être configuré pour autoriser **NGINX** en connexions entrantes et sortantes. **Certbot** sera utilisé pour générer des certificats **SSL** conformes et assurer la bonne configuration du serveur **NGINX**.

Le fichier de configuration est placé dans `/etc/nginx/sites-available/oc-pizza`. Un lien symbolique est créé entre ce fichier et le dossier `/etc/nginx/sites-enabled` pour activer le serveur.

Le fichier `nginx_example` à la racine de l'application montre les informations attendues pour le paramétrage du serveur.

### 3.1.3 - Serveur d'application

Le serveur d'application aura les caractéristiques suivantes :

- Serveur **Gunicorn**.
- Version **20.0.4** minimum.

Le fonctionnement de ce serveur d'application sera géré grâce à l'outil **Supervisord**. En charge de son démarrage et redémarrage automatique.

Un fichier de configuration, **Supervisord** est créé dans le répertoire `/etc/supervisor/conf.d/oc-pizza.conf` pour que **Supervisord** puisse démarrer le serveur **Gunicorn**.

Le fichier `supervisor_example.conf` à la racine de l'application montre les informations attendues pour le paramétrage du serveur.

## 4 - Procédure de déploiement

### 4.1 - Déploiement de l'application web

#### 4.1.1 - Environnement de l'application

Dans un premier temps, l'utilisateur **adminOcPizza** se place dans son répertoire :

```
cd /home/adminOcPizza
```

Les fichiers de l'application sont clonés depuis le dépôt GitHub :

```
git clone https://github.com/oc-pizza/oc-pizza oc-pizza
```

L'utilisateur se place ensuite dans le répertoire de l'application :

```
cd ./oc-pizza
```

Un environnement virtuel est créé dans le répertoire du projet :

```
virtualenv venv # Création de l'environnement virtuel dans le répertoire venv/  
source venv/bin/activate # Activation de l'environnement virtuel
```

Les dépendances de l'application sont installées :

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

Il ne manque plus qu'à lancer la collecte des fichiers statiques :

```
python manage.py collectstatic
```

NOTE : attention l'environnement virtuel doit être activé pour installer les dépendances du projet et utiliser `manage.py`

### 4.1.2 - Configuration de l'application

Les variables d'environnement de l'application se trouvent dans un fichier `.env` à la racine du répertoire contenant les fichiers de l'application. Un fichier d'exemple, `env_example`, est fourni avec le code.

Le fichier `.env` sera créé directement sur le serveur au moment du déploiement de l'application.

De plus, la configuration des paramètres de l'application se trouve dans le fichier `config.py` lui aussi à la racine du projet.

### 4.1.3 - Initialisation de la base de données

Vérifier dans le fichier `.env` que les paramètres de la base de donnée sont correctement renseignés.

Pour créer les tables de la base de données :

```
python manage.py migrate          # Crée les tables dans la base de données
```

La base de données sera pré-remplies à l'aide des informations transmises auparavant par OC Pizza sous forme de fichiers CSV. Ces fichiers CSV seront directement téléchargés sur le serveur dans le répertoire `/home/adminOcPizza/seed-oc-pizza`.

La commande pour lancer l'initialisation :

```
python manage.py init_db -d /home/adminOcPizza/seed-oc-pizza
```

NOTE : attention l'environnement virtuel doit être activé pour pouvoir exécuter `manage.py`



## 5 - Procédure de démarrage/arrêt

### 5.1 - Serveur d'application

Pour rappel, le démarrage et redémarrage du serveur d'application est géré automatiquement par **Supervisord**. Cependant, si cela s'avère nécessaire l'utilisateur peut exécuter les commandes suivantes :

```
sudo supervisorctl start oc-pizza
```

Pour stopper le serveur, l'utilisateur doit exécuter la commande suivante :

```
sudo supervisorctl stop oc-pizza
```

Pour contrôler l'état du processus, l'utilisateur peut exécuter la commande suivante :

```
sudo supervisorctl status oc-pizza
```

### 5.2 - Serveur Web

Pour démarrer le serveur web Nginx, l'utilisateur doit exécuter la commande suivante :

```
sudo systemctl start nginx
```

Pour stopper le serveur, l'utilisateur doit exécuter la commande suivante :

```
sudo systemctl stop nginx
```

Pour contrôler l'état du serveur, l'utilisateur peut utiliser la commande suivante :

```
sudo systemctl status nginx
```

### 5.3 - Base de données

Pour démarrer la base de données, l'utilisateur doit exécuter la commande suivante :

```
sudo systemctl start postgresql
```

Pour stopper la base de données, l'utilisateur doit exécuter la commande suivante :

```
sudo systemctl stop postgresql
```

Pour contrôler l'état de la base de données, l'utilisateur peut utiliser la commande suivante :

```
sudo systemctl status postgresql
```

**ATTENTION** : Toute opération sur la base de données doit être effectuée alors que le serveur d'application est arrêté.

## 6 - Procédure de mise à jour

### 6.1 - Mise à jour de l'application

Mise à jour des fichiers source depuis le dépôt GitHub :

```
git pull origin master
```

On met à jour les fichiers statiques :

```
python manage.py collectstatic
```

NOTE : attention l'environnement virtuel doit être activé pour pouvoir exécuter `manage.py`

ATTENTION: pour effectuer une opération de mise à jour de l'application, il est recommandé d'arrêter celle-ci.

### 6.2 - Mise à jour de la base de données

Une fois les fichiers mis à jour il est recommandé de toujours migrer la base de données afin de s'assurer que celle-ci soit toujours dans le dernier état :

```
python manage.py migrate
```

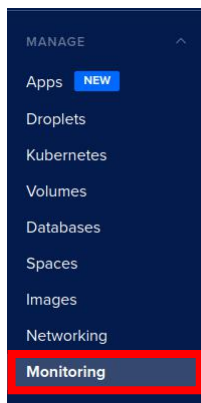
NOTE : attention l'environnement virtuel doit être activé pour pouvoir exécuter `manage.py`

ATTENTION: pour effectuer une opération de mise à jour de la base de données, il est recommandé d'arrêter celle-ci ainsi que le serveur d'application.

# 7 - Supervision et monitoring

## 7.1 - Surveillance du serveur

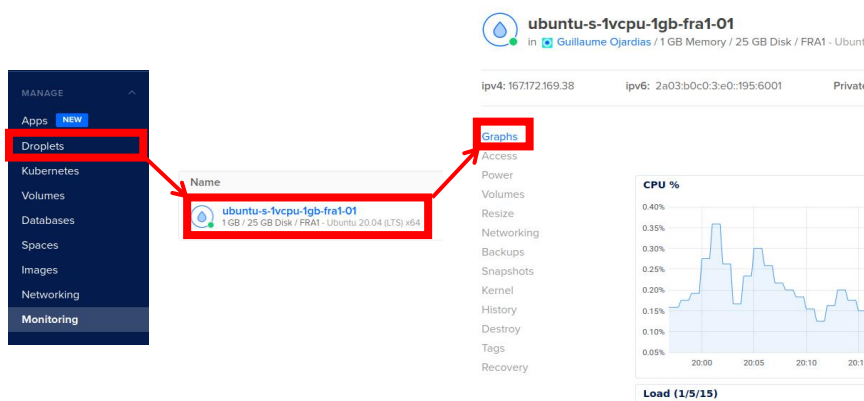
La surveillance du serveur se fait depuis l'espace client de Digital Ocean dans la partie **Monitoring** du dashboard :



Des alertes ont déjà été paramétrées pour les éléments suivants :

- > CPU;
- > utilisation du disque;
- > utilisation de la mémoire.

Les métriques en temps réel peuvent être consultées dans la partie **Graphs** des détails du droplet Digital Ocean :



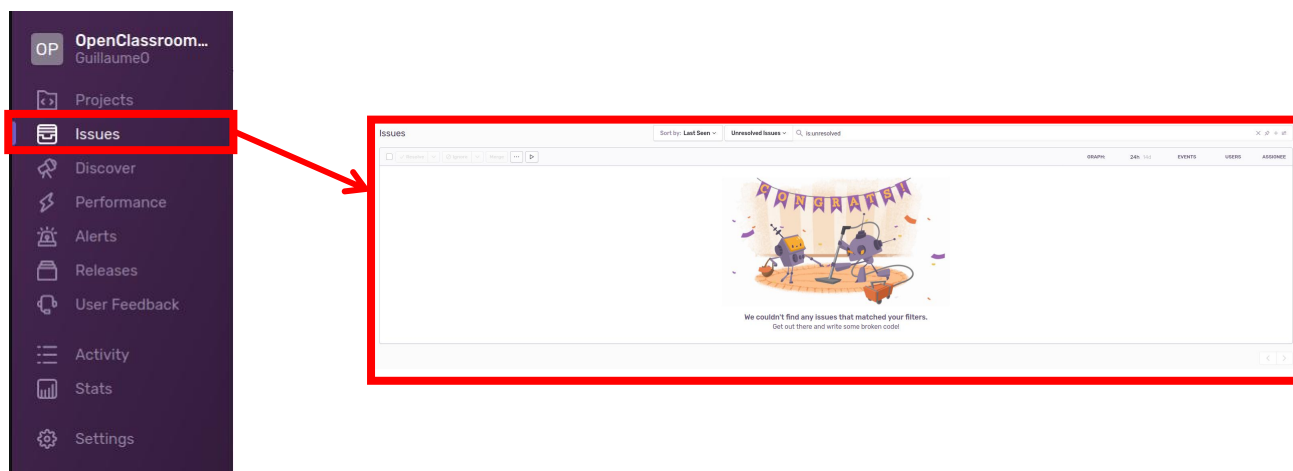
## 7.2 - Surveillance de l'application

La surveillance de l'application est assurée par Sentry : <https://sentry.io>

Sentry va permettre de logger les erreurs non gérées par l'application et d'en informer les personnes concernées.

Pour cela, des alertes sont créées afin de signaler toute nouvelle erreur sur l'adresse e-mail [admin@oc-pizza.com](mailto:admin@oc-pizza.com).

Pour visualiser les alertes il faut se rendre sur le lien **Issue** de la barre latérale :



## 8 - Procédure de sauvegarde et restauration

### 8.1 - Sauvegarde

Chaque jour, à 2h00 du matin heure de Paris, une sauvegarde de la base de données est effectuée. Cette sauvegarde est réalisée par un script, `backup.sh`, présent dans la base de code, et une tâche CRON.

Les sauvegardes se trouvent dans le dossier `/home/adminOcPizza/backups/oc-pizza/`.

Les sauvegardes sont nommées de la manière suivante : `oc-pizza-AAAA-MM-JJ-HH.gz` avec :

- > AAAA => l'année;
- > MM => le mois;
- > JJ => le jour.
- > HH => l'heure.

### 8.2 - Restauration

Pour restaurer la base de données à un état donné :

```
gunzip -c /home/adminOcPizza/backups/oc-pizza/oc-pizza-AAAA-MM-JJ-HH.gz | psql oc-pizza
```

NOTE: Ne pas oublier de modifier le nom de fichier de sauvegarde pour une exécution correcte de la commande.

## 9 - Glossaire

<b>Certbot</b>	Logiciel de certification édité par l'autorité de certification Let's Encrypt.
<b>Django</b>	Framework écrit dans le langage Python.
<b>GIT</b>	Logiciel de gestion de versions d'une base de code.
<b>GitHub</b>	Plateforme d'hébergement en ligne de code source.
<b>Gunicorn</b>	Logiciel de serveur d'application Python.
<b>Nginx</b>	Logiciel de serveur web.
<b>OpenSSH</b>	Ensemble d'outils utilisant le protocole sécurisé SSH.
<b>Postgresql</b>	Logiciel de gestion de base de données.
<b>Python</b>	Langage de programmation.
<b>Supervisord</b>	Logiciel de gestion des processus système.
<b>UFW</b>	Utilitaire pour la gestion du pare-feu.
<b>Virtualenv</b>	Outil permettant la création d'environnement Python isolé du reste du système.