

# OC Pizza

## Système de gestion de prise de commande

Dossier d'exploitation

Version 1.0

**Auteur**

Alix VOINOT

*Developpeur*

# Sommaire

## 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
3.1.2 - Web service.....	5
<b>4 - Procédure de déploiement.....</b>	<b>6</b>
4.1 - Déploiement de l'application web.....	6
4.1.1 - Composition de l'application web.....	6
4.1.2 - Variables d'environnement.....	6
4.1.3 - Configuration.....	7
4.1.4 - Déploiement.....	7
4.1.5 - Vérifications.....	7
4.2 - Déploiement de la base de données.....	8
4.2.1 - Lancement des migrations.....	8
4.2.2 - Création super utilisateur.....	8
4.2.3 - Import des données.....	8
4.2.4 - Vérification.....	8
<b>5 - Procédure de démarrage / arrêt.....</b>	<b>9</b>
5.1 - Démarrage/arrêt depuis Heroku.....	9
5.2 - Démarrage/arrêt en ligne de commande.....	9
<b>6 - Procédure de mise à jour.....</b>	<b>10</b>
<b>7 - Supervision/Monitoring.....</b>	<b>11</b>
7.1 - Outils de métriques.....	11
<b>8 - Procédure de sauvegarde et restauration.....</b>	<b>12</b>
<b>9 - Glossaire.....</b>	<b>13</b>

# 1 - VERSIONS

Auteur	Date	Description	Version
Alix VOINOT	25/02/2019	Finalisation du document	1.0

## 2 - INTRODUCTION

### 2.1 - Objet du document

Le présent document constitue le dossier d'exploitation de l'application Ocpizza. L'objectif de ce document est de fournir à l'équipe technique de la société ocpizza les informations essentielles pour une bonne utilisation de l'application ainsi que les instructions à suivre pour le déploiement et la maintenance de celle-ci.

### 2.2 - Références

Pour de plus amples informations, se référer :

1. Dossier de conception technique
2. Dossier de conception fonctionnelle

## 3 - PRÉ-REQUIS

### 3.1 - Système

L'application web OCpizza sera hébergé sur la plateforme web Heroku. Elle sera hébergé sur ocpizza.com .

#### 3.1.1 - Serveur de Base de données

Le SGBD utilisé pour l'application sera MySQL 8.0.14. La base de données sera hébergé sur le serveur d'heroku.

#### 3.1.2 - Web service

Le fonctionnement de l'application nécessite que les différents services soient opérationnelles :

- API Google map : l'utilisation de cette API nécessite la clé d'API suivante : **zwv8vi48467v8LnxG7VGSBNN**
- Stripe : l'utilisation de cette API nécessite la clé d'API suivante : **TCDed77v85si8YtUK72Z5vDq**
- Mailjet : l'utilisation de cette API nécessite la clé d'API suivante : **ST262wV77eKf8WkN76YxaSmq**

Ces 3 services étant payants, il faudra choisir la formule adaptée aux besoin du système pour éviter la surfacturation.

# 4 - PROCÉDURE DE DÉPLOIEMENT

## 4.1 - Déploiement de l'application web

### 4.1.1 - Composition de l'application web

L'application Ocpizza est construite sous la forme d'une archive Zip contenant les répertoires suivants:

- **.oCPIZZA:** contient tout les fichiers de l'application et les fichiers de configuration Django
- **procfile:** indique au serveur le langage utilisé par l'application ainsi que le script à exécuter pour démarrer l'application
- **requirements.txt:** indique à l'environnement quels sont les dépendances dont il a besoin pour fonctionner
- **static:** contient les fichiers CSS et Javascript de l'application
- **template:** contient les fichiers HTML de l'application
- **documentations:** contient les fichiers de documentations de l'application

Le fichier Procfile contient les instructions à exécuter par Heroku pour démarrer l'application. Le contenu de ce fichier doit être à minima:

```
web: gunicorn ocpizza.run :app
```

### 4.1.2 - Variables d'environnement

Voici les variables d'environnement reconnues par l'application :

Nom	Obligatoire	Valeur	Description
ENV	oui	PRODUCTION	Indique à l'application qu'il faut utiliser la configuration de production et non de développement.
SECRET_KEY	Oui	fsdqfddf656565dfq6	Nécessaire au démarrage de Django. Cette clé ne doit figurer nulle part ailleurs que sur le serveur de production.

### 4.1.3 - Configuration

Voici les différents fichiers de configuration:

- **Procfile** : indique au serveur le langage utilisé par l'application ainsi que le script à exécuter pour démarrer l'application
- **requirements.txt** : indique à l'environnement quels sont les dépendances dont il a besoin pour fonctionner
- **ocpizza/ocpizza/settings.py**: fichiers de configuration de l'application Django

### 4.1.4 - Déploiement

Heroku propose différentes méthodes pour déployer son application. Nous conseillons cependant d'effectuer un déploiement en ligne de commande en utilisant "Heroku CLI". Les étapes présentées ci-dessous nécessitent que "Heroku CLI" soit installé et que le projet soit suivi avec Git. Le projet doit être sur la branche master.

#### 4.1.4.1 - Création de l'application

Une fois "Heroku CLI" installé et la connexion à ocpizza effectuée, il faut se rendre à la racine du projet et rentrer la commande : *heroku create ocpizza* .

#### 4.1.4.2 - Configurer les variables d'environnement

Les variables d'environnement sont rentrés via les lignes de commande de la façon suivante:

```
heroku config:set ENV=PRODUCTION
```

```
heroku config:set SECRET_KEY=SDFFSGSGFD
```

Pour vérifier que les variables ont bien été configuré, il faut entrer la commande:

```
heroku config
```

#### 4.1.4.3 - Envoyer l'application sur le serveur

Une fois les étapes précédentes réalisées, effectuez la commande *git push heroku master*.

### 4.1.5 - Vérifications

Si votre site a bien été déployé, [www.ocpizza.com](http://www.ocpizza.com) sera accessible sur votre navigateur.

## 4.2 - Déploiement de la base de données

Le déploiement de la base de données sur la plateforme Heroku nécessite que l'application ait été déployée comme présenté précédemment. Nous utiliserons Heroku CLI pour le déploiement de la base de données.

### 4.2.1 - Lancement des migrations

Avant de lancer les migrations, nous allons créer les tables avec la commande suivante:

```
heroku run python manage.py migrate
```

### 4.2.2 - Création super utilisateur

Une fois la BDD créé, il faut ensuite créer un super-utilisateur via la commande suivante:

```
heroku run python manage.py create_superuser .
```

### 4.2.3 - Import des données

Il est possible d'importer les données de la version de développement de l'application sur le serveur de production. Pour cela, il faut créer un dump de la base de données de développement puis l'importer sur le serveur heroku.

```
./manage.py dumpdata pizzapp > pizzapp/dumps/pizzapp.json - Import des données dans la base :  
$ heroku run python manage.py loaddata pizzapp/dumps/pizzapp.json
```

### 4.2.4 - Vérification

Nous avons fourni le script `verif.py`, qui, une fois lancé, vérifie que tout les éléments sont présent dans la BDD.



## 5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

La gestion de l'application peut s'effectuer de 2 manières différentes:

- depuis le site de Heroku
- en ligne de commande après avoir téléchargé l'IHM de Heroku

### 5.1 - Démarrage/arrêt depuis Heroku

Une fois déployé sur Heroku, l'application est par défaut en ligne. Afin de désactiver l'application, il faut se rendre dans la rubrique settings du site et activer le mode maintenance.

On dispose à présent de la possibilité d'effectuer nos maintenances.

### 5.2 - Démarrage/arrêt en ligne de commande

Pour démarrer le serveur, il faut utiliser les commande suivantes:

- `heroku run` pour démarrer le serveur
- `heroku ps :scale web = 0` pour éteindre le serveur

## 6 - PROCÉDURE DE MISE À JOUR

La mise à jour de l'application nécessite de passer celle-ci en mode maintenance. Pour effectuer cette action, veuillez vous référer au point 5. Nous pouvons aussi effectuer cette action en passant par ligne de commande avec la commande *heroku maintenance :on*.

Lorsque le mode maintenance est activé, un message d'alerte s'affiche sur l'application. Les utilisateurs n'y ont plus accès. Vous pouvez maintenant effectuer toutes les modifications nécessaires pour mettre à jour votre application.

Une fois la mise à jour effectuée, il faut remettre le mode maintenance en off avec la commande: *heroku maintenance :off*. Le message d'alerte sera alors retiré. Après quelques minutes, les utilisateurs devraient avoir à nouveau accès au site.

# 7 - SUPERVISION/MONITORING

## 7.1 - Outils de métrics

La plateforme Heroku fournit ses propres outils de métrics dans la rubrique métrics. En les consultant, vous aurez accès au temps de réponse, aux nombres de requêtes en erreur sur votre site, l'usage mémoire détaillé de votre serveur ainsi que les évènements de votre site ( migrations, pic de connexion ...).

Dans cette partie, vous pourrez aussi configurer un ou plusieurs seuils d'alerte en fonction des temps de réponse sur votre site.

Pour de plus amples informations, vous pouvez consulter cette page:

<https://devcenter.heroku.com/articles/metrics#metrics-gathered-for-web-dynos-only>

Nous vous recommandons de mettre en place une alerte sur les *down times* pour vérifier que votre site est continuellement accessible.

Nous avons aussi configuré un sentry qui permet de suivre toutes les requêtes effectuées par le site.

## 8 - PROCÉDURE DE SAUVEGARDE ET RESTAURATION

Le processus de sauvegarde et de restauration vous permet de sauvegarder les données utilisateurs et du site de façon rapide et sécurisé. Pour cela, nous avons mis en place un batch qui fait la sauvegarde des données tous les jours à 4 heures du matin. Ces sauvegardes sont accessibles dans le fichiers /backup. Le nom des backups correspond à leur date de sauvegarde. Ils disposent d'une durée de vie d'un mois.

Pour effectuer une sauvegarde manuelle, il faut entrer la commande suivante:

```
heroku pg:backups:capture --app ocpizza
```

Pour effectuer la restauration de la base de données, il faut entrer la commande suivante:

```
heroku pg:backups:restore /backup/<nom sauvegarde> DATABASE_URL --app ocpizza
```

## 9 - GLOSSAIRE

SGBD	Système de gestion de base de données. C'est un logiciel permettant de stocker des informations dans une base de données.
API	Application programming interface. Une interface de programmation est une façade clairement délimitée par laquelle un logiciel offre des services à d'autres logiciels. L'objectif est de fournir une porte d'accès à une fonctionnalité en cachant les détails de la mise en œuvre.
Variables d'environnement	Les variables d'environnement sont des variables dynamiques utilisées par les différents processus d'un système d'exploitation (Windows, Unix, etc.). Elles servent à communiquer des informations entre programmes qui ne se trouvent pas sur la même ligne hiérarchique, et ont donc besoin d'une convention pour se communiquer mutuellement leurs choix
BDD	Base de données. Un "conteneur" qui stocke les données du système.
IHM	Interface homme-machine. Cela correspond à la partie de notre système que l'utilisateur a sur son écran, soit le site internet visible depuis son navigateur.
Monitoring	Le monitoring est une activité de surveillance et de mesure d'une activité informatique.
Outil de métrics	Outils permettant de mesurer ses performances sur le web
Down times	Période durant laquelle un système n'est pas disponible
Sentry	Outil de surveillance pour les adresses IP