

IT Consulting & Development

# OC Pizza

Dossier d'exploitation d'un système de gestion de  
groupe de pizzerias

Dossier d'exploitation

Version 0.1

# Table des matières

---

<b>1 - Versions</b>	<b>1</b>
<b>2 - Introduction</b>	<b>2</b>
Objet du document	2
Rappel du contexte	2
Références	2
<b>3 - Pré-requis</b>	<b>3</b>
Système	3
Serveur web	3
Serveur de base de données	3
Web-services	3
<b>4 - Procédure de déploiement</b>	<b>4</b>
Déploiement de l'application web	4
Dépôt Github	4
Variables d'environnement	4
Configuration	4
Déploiement	5
Vérifications	5
<b>5 - Procédure de démarrage / arrêt</b>	<b>6</b>
Procédure de démarrage	6
Procédure d'arrêt	6
<b>6 - Procédure de mise à jour</b>	<b>7</b>
Application Web	7
<b>7 - Supervision / Monitoring</b>	<b>8</b>
Monitoring de l'Application Web	8
<b>8 - Procédure de sauvegarde et restauration</b>	<b>9</b>
Base de données	9
<b>9 - Glossaire</b>	<b>10</b>

## 1 - Versions

---

Auteur	Date	Description	Version
Sebastien Goliot	17/06/2021	Création du document	0.1

## 2 - Introduction

---

### Objet du document

Le présent document constitue le dossier d'exploitation du système de gestion des pizzerias du groupe OC Pizza.

Ce document a pour objectif de présenter le déploiement du projet.

### Rappel du contexte

“OC Pizza” est un jeune groupe de pizzerias spécialisé dans la vente de pizzas livrées ou à emporter, comprenant déjà 5 points de vente et prévoyant l'ouverture de 3 pizzerias supplémentaires d'ici 6 mois.

Leur système informatique ne répondant plus à leurs besoins, le groupe aimerait être plus efficace dans la gestion des commandes et bénéficier d'un suivi de chaque commande ainsi que des stocks.

La solution proposée devra être livrée pour l'ouverture des 3 nouvelles pizzerias.

### Références

Pour de plus amples informations, se référer à:

Document	Description
OC_p4-Dossier_fonctionnel.pdf	Dossier de spécifications fonctionnelles
OC_p6-Dossier_conception_technique.pdf	Dossier de conception technique

# Systeme

## Serveur web

## Serveur de base de données

## Web-services

L'application nécessite l'usage des web services suivants:

- Stripe : Pour la gestion des paiements
- Google Maps API : Pour la géolocalisation et le calcul d'itinéraire

Ces services nécessitent des clés d'API, disponibles aux emplacements suivants:

Nom	Emplacement
MAPS_API_KEY	<a href="https://console.cloud.google.com/google/maps-apis/credentials">https://console.cloud.google.com/google/maps-apis/credentials</a>
STRIPE_API_KEY	<a href="https://dashboard.stripe.com/apikeys">https://dashboard.stripe.com/apikeys</a>

Ces clés d'API seront à conserver dans le fichier .env tel que spécifié dans la section suivante.

Les autres variables d'environnement nécessaires:

[illegible]

## 4 - Procédure de déploiement

### Déploiement de l'application web

#### - Dépôt Github

L'application est versionnée avec Git et sera déployée depuis un dépôt Github.

#### - Variables d'environnement

Les variables d'environnement doivent être contenues dans un fichier .env avec les valeurs suivantes:

Nom	Description
POSTGRES_NAME	Le nom de la base de données PostgreSQL
POSTGRES_USER	Le nom d'utilisateur de la base de données
POSTGRES_PASS	Le mot de passe de l'utilisateur de la base de données
POSTGRES_HOST	Le nom d'hôte de la base de données
DJANGO_SETTINGS_MODULE	Le module de paramètre à utiliser Dans le cas du déploiement en production, cette valeur devra être égale à "ocpizza.settings.production"
SECRET_KEY	La clé secrète de Django, à générer lors du déploiement
SECRET_ADMIN_URL	Une URL secrète pour l'accès à l'interface d'administration
SENTRY_DSN	Le DSN fourni par Sentry
STRIPE_API_KEY	La clé d'API de Stripe
MAPS_API_KEY	La clé d'API de Google Maps

note : Un fichier .env.example est présent dans le dépôt, afin qu'il soit copié vers .env  
**toutes les variables ci-dessus sont obligatoires**

#### - Configuration

Voici les différents fichiers de configuration, tous présents dans le dépôt :

Nom	Description
.env	Fichier contenant toutes les variables d'environnement
ocpizza/settings/base.py	La configuration de base de l'application
ocpizza/settings/development.py	La configuration de développement de l'application
ocpizza/settings/production.py	La configuration de production de l'application
requirements.txt	Les dépendances qui seront installées lors du déploiement
ocpizza.conf	La configuration du projet pour Supervisor

## - Déploiement

Afin de permettre le déploiement, il faudra ajouter les identifiants de connexion aux variables secrètes du dépôt Github afin que le CI puisse s'occuper du déploiement après avoir lancé les tests.

Afin de s'assurer que le déploiement de l'application se déroule correctement, les variables secrètes suivantes sont nécessaires :

Nom	Description
SSH_HOST	L'adresse IP du serveur
SSH_PORT	Le port SSH du serveur
SSH_USER	L'utilisateur à utiliser pour le déploiement
SSH_KEY	La clé SSH privée à utiliser pour le déploiement
SSH_PASS	Le mot de passe de la clé SSH

Il suffira ensuite de lancer le workflow Github Actions et, si les tests sont verts, le déploiement sera effectué automatiquement.

## - Vérifications

Afin de s'assurer du bon déploiement de l'application, certains points seront à vérifier après son démarrage (spécifié dans la section suivante):

- Le site est bien accessible via son URL.
- Aucune erreur n'est remontée par Sentry

## 5 - Procédure de démarrage / arrêt

---

### Procédure de démarrage

Afin de démarrer l'application, il est nécessaire d'installer et de configurer Supervisor.

Ce dernier a besoin d'une configuration propre à l'application à démarrer, qui est présente dans le dépôt du projet sous la forme du fichier `ocpizza.conf`

Il faudra donc déplacer ce fichier à l'emplacement suivant :

```
/etc/supervisor/conf.d/ocpizza.conf
```

Il faudra ensuite créer le dossier de logs:

```
mkdir /var/log/ocpizza
```

Et enfin, activer Supervisor et s'assurer que tout est ok

```
systemctl enable supervisor  
supervisorctl reread  
supervisorctl update  
supervisorctl status
```

Si besoin, il est possible de démarrer l'application manuellement à l'aide de la commande suivante :

```
supervisorctl start ocpizza
```

### Procédure d'arrêt

Tout comme la procédure de démarrage, l'arrêt s'effectue aussi à l'aide de Supervisor, et ce avec la commande suivante :

```
supervisorctl stop ocpizza
```

Il est aussi possible de redémarrer l'application avec :

```
supervisorctl restart ocpizza
```

## 6 - Procédure de mise à jour

---

### Application Web

La mise à jour de l'application web s'effectue automatiquement à l'aide du workflow de CI Github Actions.

Son fonctionnement est le suivant:

- Après avoir effectué une modification, ces modifications arriveront sur le dépôt Github.
- Le CI Github Actions lancera automatiquement les tests dès qu'il recevra les modifications.
- Si les tests sont verts, le CI effectuera le déploiement des mises à jour.



## 7 - Supervision / Monitoring

---

### Monitoring de l'Application Web

Le monitoring de l'application est effectué à l'aide de Sentry, ce dernier fournit des rapport d'erreur et permet la correction rapide de tout éventuel problème rencontré.

## 8 - Procédure de sauvegarde et restauration

### Base de données

La base de données utilisée étant gérée par DigitalOcean, il n'est pas nécessaire de se soucier outre-mesure de la sécurité des données.

DigitalOcean effectue des sauvegardes quotidiennes des données, et ces dernières sont conservées pendant 7 jours.

En cas d'incident il est possible de manuellement restaurer les données en suivant les instructions présentes dans la documentation de DigitalOcean.

[How to Manually Restore PostgreSQL Database Clusters from Backups :: DigitalOcean Documentation](#)

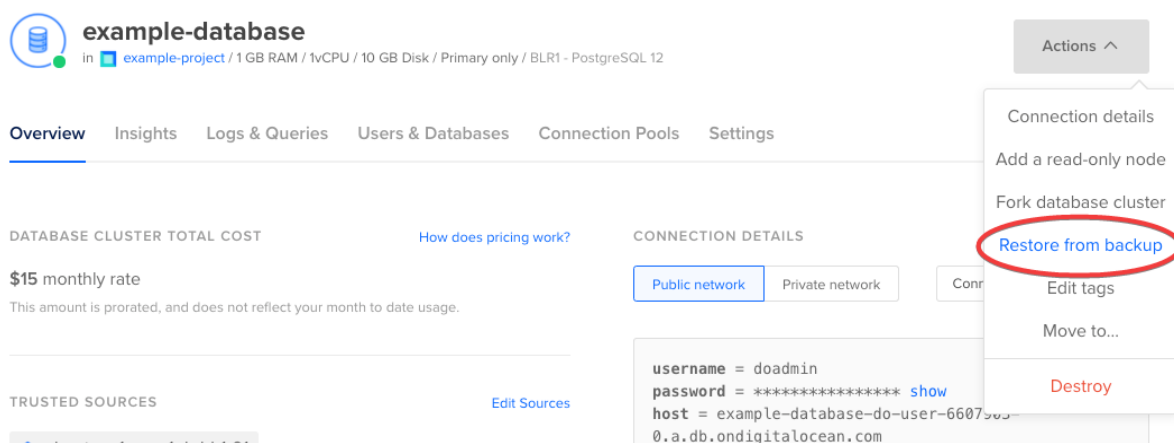


fig. 01

## 9 - Glossaire

---

DigitalOcean	Un fournisseur d'infrastructure cloud.
Droplet	L'appellation d'une unité de serveur chez Digitalocean.
NGINX	Un logiciel de serveur web.
Reverse-Proxy (Proxy inverse)	Un type de configuration serveur permettant d'exposer des ressources internes sur le réseau internet.
WSGI	(Web Server Gateway Interface) est une interface entre serveurs web et applications Python
Gunicorn	Un serveur web WSGI permettant la communication avec une application Python
SGBD	(Système de Gestion de Base de Données)
Git	Git est un logiciel de gestion de versions.
Github	Github est un hébergeur de dépôts Git.
CI	(Continuous Integration) est une pratique permettant de s'assurer de la qualité de code au fil des modifications.
Github Actions, Workflow	Github Actions permet de réaliser des actions automatisées telles que du CI, un workflow est un ensemble d'actions.
Supervisor	Un logiciel de contrôle de processus
Sentry	Un logiciel de monitoring et de suivi d'erreurs.