



I.S

# OC-Pizza

## Application

Dossier d'exploitation

Version 1.0

**Auteur**

*ihсан salman*

# 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 - <i>Serveur de Base de données</i> .....	5
3.1.1.1 - Caractéristiques techniques.....	5
3.1.2 - <i>Serveur Web</i> .....	5
3.1.2.1 - Caractéristiques techniques.....	5
3.2 - Web-services.....	6
3.3 - Autres Ressources.....	6
<b>4 - Procédure de déploiement.....</b>	<b>7</b>
4.1 - Déploiement de l'Application Web.....	7
4.1.1 - <i>Variables d'environnement</i> .....	7
4.1.2 - <i>Configuration</i> .....	7
4.1.3 - <i>Ressources</i> .....	7
4.1.4 - <i>Vérifications</i> .....	7
<b>5 - Procédure de démarrage / arrêt.....</b>	<b>9</b>
<b>6 - Procédure de mise à jour.....</b>	<b>10</b>
6.1 - Base de données.....	10
6.2 - Application web.....	10
<b>7 - Supervision/Monitoring.....</b>	<b>11</b>
7.1 - Supervision de l'application web.....	11
7.2 - Gestion espace disque.....	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
IH	04/08/2021	Création du document	V 0.1

## 2 - INTRODUCTION

### 2.1 - Objet du document

Le présent document constitue le dossier d'exploitation de l'application Web de OC-Pizza afin de remplacer le système informatique actuel.

### 2.2 - Références

Pour de plus amples informations, se référer:

- **P9 – Dossier de conception fonctionnelle:** Dossier de conception fonctionnelle de l'application
- **P9 – Dossier de conception technique:** Dossier de conception technique de l'application
- **P9 – Livraison :** Livraison de l'application

# 3 - PRÉ-REQUIS

## 3.1 - Système

### 3.1.1 - Serveur de Base de données

La base de données est un serveur MySQL par sa simple utilisation. Le serveur de la base de données sera hébergé sur un Cloud d'OVH comme dis plutôt. La configuration du serveur se fera dans le fichier db\_config.py.

#### 3.1.1.1 - Caractéristiques techniques

Afin de relier la base de données et le fichier de configuration, on utilise la librairie mysql\_connector ce qui permet une gestion de la base de données simple.

Les requêtes HTTP de l'utilisateur seront automatiquement redirigé vers le site en HTTPS pour sécuriser les échanges de données entre l'utilisateur et le serveur.

### 3.1.2 - Serveur Web

Il s'agira de prendre l'offre CloudWeb d'OVH pour héberger le site internet pour un montant de 12 euros/mois, qui donne accès à 1 :

- vCore de 2,4 GHz (processeur)
- 2 Go RAM de mémoire vive
- 10 Go SSD (stockage très rapide en performance)

Afin de relier la base de données et le fichier de configuration, on utilise la librairie mysql\_connector ce qui permet une gestion de la base de données simple.

Les requêtes HTTP de l'utilisateur seront automatiquement redirigé vers le site en HTTPS pour sécuriser les échanges de données entre l'utilisateur et le serveur.

#### 3.1.2.1 - Caractéristiques techniques

Le serveur sera un serveur ubuntu 20.04, les mises à jour sont installées automatiquement et OC-Pizza sera averti par mail. Les dépendances liées au serveur bénéficient d'une mise à jour automatique du fait qu'elles sont installées depuis ubuntu.

Pour le bon fonctionnement du serveur il faut installer les paquets suivant :

- sudo
- net-tools
- python3-pip
- openssh-server
- docker

Pour les installer, exécuter la commande suivante : **apt-get install <nom du paquet>**

Un module python pour surveiller l'espace disque est nécessaire, pour cela utiliser la commande suivante : **pip3 install psutil.**

## 3.2 - Web-services

Les web services suivants doivent être accessibles et à jour :

- **Google map API:** Cette API permet de renvoyer des données GPS de l'utilisateur pur récupérer une carte de la localisation et du parcours vers l'adresse souhaitée. Elle nécessite une clé pour fonctionner.
- **Square:** Cette API permet de relier les données bancaires de l'utilisateur et le serveur de paiement en garantissant la sécurité de la transaction en plusieurs étapes de validation, où une clé de token sécurise chaque étape en respectant des normes de sécurité de transaction bancaire. Elle nécessite une clé pour fonctionner.

Toutes les clés de fonctionnement sont intégrés sur le serveur en variable d'environnement.

## 3.3 - Autres Ressources

Afin de permettre la connexion au serveur, un compte utilisateur **ocpizza** a été déjà crée avec une clé ssh fourni et un logiciel pour se connecter sur les postes a été installés. Les étapes pour installer ce logiciel gratuit ont été aussi fourni. Ce compte dispose des autorisations nécessaire pour accéder à tous les niveaux d'accessibilités du serveur.

Toutes les dépendances du systèmes nécessaire au bon fonctionnement du logiciel ont été faite avec la compatibilité du serveur.

Le site internet oc-pizza fonctionne en HTTPS, des fichiers nécessaires au TLS se trouvent dans un répertoire etc/tsl/ocpizza. Les clé TLS sont valables 36 mois, il faudra donc les renouveler, pour cela :

- se mettre dans le répertoire etc/tsl/ocpizza
- entrer la commande suivante : `sudo openssl req -new -key ocpizza.key -out ocpizza.csr`
- répondre aux questions et se rendre sur le site de la société SSLs <https://www.ssls.com> et acquérir une nouvelle clé
- placer le fichier .crt obtenu dans le répertoire cité dans l'étape 1.

# 4 - PROCÉDURE DE DÉPLOIEMENT

## 4.1 - Déploiement de l'Application Web

### 4.1.1 - Clonage du dépôt git, migration des données et fichier static...

Afin de récupérer le projet hébergé sur github, il faut cloner le code dans un dépôt local de cette manière:

- installer git à partir de cette ligne de commande  
**sudo apt install git**
- utiliser cette commande pour cloner le projet dans le chemin désiré  
git clone <lien git de votre projet qui se finit par .git>

Ensuite il faut migrer les données de l'application vers la base de données:

- **cd ocpizza** (se placer à l'intérieur du projet)
- **./manage.py migrate** (migration des données vers la base de données).

Et enfin, il ne reste plus qu'à récupérer tous les fichiers static de l'application et les placer dans un fichier pour faciliter le service:

- **./manage.py collectstatic**

### 4.1.2 - Variables d'environnement

Le serveur d'application JOnAS doit être exécuté avec la variable d'environnement suivante définie au démarrage. Elle est nécessaire afin de récupérer le répertoire contenant les fichiers de configuration de l'application :

**-Dcom.ocpizza.apps.conf=\$home\_application\_conf\_directory**

INFO : il ne faut pas mettre de « / » à la fin de la valeur de la variable et ne pas utiliser d'espace dans le chemin.

Voici les variables d'environnement nécessaire pour le bon fonctionnement de l'application:

Nom	Obligatoire	Description
DJANGO_SETTINGS_MODULE	OUI	Situe le fichier settings.py de production
SECRET_KEY	Oui	Nécessaire à la sécurité des échanges utilisateur/serveur

### 4.1.3 - Configuration

Le répertoire de configuration applicatif doit être créé sur le système de fichier et définit de la façon suivante :

```
$home_application_conf_directory/applicationX
```

### 4.1.4 - Ressources

Les accès aux bases de données doivent se configurer à l'aide des fichiers...

Le fichier de drivers **postgresql (postgresql-9.2.x.)** doit être déposé dans le répertoire :

```
$home_server/lib/ext
```

### 4.1.5 - Vérifications

Afin de vérifier le bon déploiement des batches, entrez la commande suivante:

```
sudo docker exec -ti ocpizza/usr/bin/python3 manage.py test
```

Ce test va permettre de vérifier l'ensemble des informations du déploiements qui doivent passer sans erreur. Par la suite la connection au site devrais se faire normalement.



## 5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

L'arrêt du serveur n'est pas envisageable et nécessaire, cependant le tableau de bord OVH permet d'arrêter et de redémarrer le serveur. Après la fin de l'arrêt du serveur, le redémarrage est automatique.

2 scripts permettent de mettre le site en indisponibilité et en disponibilité en exécutant respectivement les commandes suivantes :

**sudo home/ocpizza/docker\_indispo.sh**

**sudo home/ocpizza/docker\_dispo.sh**

## 6 - PROCÉDURE DE MISE À JOUR

### 6.1 - Base de données

La base de données est facilement accessible pour l'administrateur et les changements dans le script de la base de données sera automatiquement déployés après que les tests auront été en succès.

Le fait de passer des tests avant de mettre à jour l'application permet d'assurer que les changements apporter n'affectent pas le reste.

### 6.2 - Application web

Sur le même principe que les mises à jours de la base de données, les changement apportés devront d'abord passer des tests et ensuite déployer la mise à jour sans mettre en arrêt le serveur de l'application.

# 7 - SUPERVISION/MONITORING

## 7.1 - Supervision de l'application web

La supervision de l'application se fait au travers du tableau de bord d'OVH, dispose d'outils de paramétrage afin de garantir la bonne exécution de l'application. En cas de soucis au niveau du serveur de l'application, un mail sera envoyé à l'administrateur du site.

## 7.2 - Gestion espace disque

L'hébergeur du site ne permet d'avertir de problèmes liés à l'espace de disque. Pour cela, une procédure a été créée, dans le répertoire root/ et se nomme disk\_check\_space.py. Ce processus est lancé automatiquement le matin.

## 8 - PROCÉDURE DE SAUVEGARDE ET RESTAURATION

La base de données est hébergé sur le Cloud d'OVH donc elle est automatiquement sauvegardé quotidiennement, cependant il est possible de le faire manuellement sous l'utilisateur ocpizza :

**ocpizza → pg\_dump ocpizza\_db -F c -f /var/tmp/ocpizza\_db.dmp**

La restauration de la base de données reste un risque rare et exceptionnel, qui peut entrainer une perte de données importante. Le client devra informer l'équipe de développeur pour effectuer le travail nécessaire à la mise en place d'une procédure de restauration

## 9 - GLOSSAIRE
