



OC PIZZA

Projet 9

Dossier d'exploitation

Version 1.0

Auteur
Yann Hamdi



TABLE DES MATIÈRES

1 - VERSIONS	4
2 - INTRODUCTION	5
2.1 - Objet du document	5
2.2 - Références	5
3 - PRÉ-REQUIS	6
3.1 - Système	6
3.1.1 - Serveur de Base de données	6
3.1.2 - Serveur Web	6
3.1.3 - Serveur de Batches	6
3.2 - Bases de données	6
3.3 - Web-services	6
4 - PROCÉDURE DE DÉPLOIEMENT	8
4.1 - Déploiement des Batches	8
4.1.1 - Artefacts	8
4.1.2 - Variables d'environnement	8
4.1.3 - Configuration	8
4.1.4 - DataSources	9
5 - PROCÉDURE DE DÉMARRAGE / ARRÊT	10
5.1 - Base de données	10
5.2 - Application web	10
6 - PROCÉDURE DE MISE À JOUR	11
6.1 - Base de données	11
6.2 - Application web	11



7 - SUPERVISION/MONITORING	12
7.1 - Supervision de l'application web	12
8 - PROCÉDURE DE SAUVEGARDE ET RESTAURATION	13
9 - GLOSSAIRE	14



1 - VERSIONS

Auteur	Date	Description	Version
Yann Hamdi	19/11/2020	Création du document	1.0



2 - INTRODUCTION

2.1 - Objet du document

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

Ce document représente la mise en production des systèmes développés par IT EXPERTS.

2.2 - Références

Pour de plus amples informations, se référer :

1. **DCT – 1.0** : Dossier de conception technique de l'application.
2. **DCF – 1.0** : Dossier de conception fonctionnelle de l'application.



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

Nous allons utiliser Digital Ocean sur lequel il y aura 1 private CPU, 1Gb de Mémoire, 1 SSD 25 Gb. 1Tb transfert et le Datacenter sera à Londres. Les spécifications du serveur pourront être revaloriser en cas d'augmentation du trafic ou des besoins de l'application.

3.1.2 - Serveur Web

Le serveur virtuel hébergeant l'application web sera fourni par Digital Ocean sous Ubuntu, les trafics gérés par Nginx 1.18.0, la communication vers le serveur Python sera gérée par Gunicorn 20.0.4 grâce à son module WSGI inclus chez Django et le serveur web sera surveillé par Supervisor 4.2.0

3.1.3 - Serveur de Batches

Concernant le serveur de Batches, ça sera le même que pour le web, il nous servira pour l'exécution des tâches CRON.

3.2 - Bases de données

Les bases de données et schémas suivants doivent être accessibles et à jour :

- DCT—1.0 : version 1.0

3.3 - Web-services

Les web services suivants doivent être accessibles et à jour :

- Paypal : <https://developer.paypal.com/docs/api/overview/#>
- Google

Maps:



<https://developers.google.com/maps/documentation/geocoding/start?hl=fr>



4 - PROCÉDURE DE DÉPLOIEMENT

4.1 - Déploiement des Batches

4.1.1 - Artefacts

Les fichiers de notre application web se trouvent sur un repository GitHub.

Il faudra donc cloner sur notre serveur le repository. Pour cela nous allons nous connecter au serveur de production sous ssh en entrant cette commande :

Ssh [ocpizza@305.190.445.87](https://github.com/ocpizza/OC_pizza) la suite de numéro étant l'adresse IP qui nous sera fourni par digital ocean.

Puis nous clonons notre application avec la commande :

```
git clone https :github.com/ocpizza/OC_pizza.git
```

Les fichiers de l'application seront alors présents sur notre serveur de production.

4.1.2 - Variables d'environnement

Voici les variables d'environnement reconnues par les batches de l'application oc_pizza :

Nom	Description
EMAIL= XXXX	Il s'agit du mail qui servira pour la messagerie
EMAIL_PW=XXXX	Mot de passe que vous devez créer
DB_USER= XXXXX	Utilisateur pour se connecter à la base de données
DB_PW= XXXX	Mot de passe pour la base de données
SECRET_KEY	Clé secrète pour Django

Définissez les variables d'environnement nécessaires en remplaçant les XXXXX

4.1.3 - Configuration

Voici les différents fichiers de configuration :

- **__init__.py** : fichier de configuration en local
- **Production.py** : fichier de configuration de l'application en production



4.1.4 - DataSources

La base de données est configurée grâce aux fichiers de configuration de notre application web.

Nous récupérons nos données par la commande Django :

```
./manage.py migrate
```

...



5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

5.1 - Base de données

Concernant la base de données, elle est gérée par MYSQL, il nous faudra d'abord nous connecter, puis on pourra démarrer ou arrêter la base de données avec les commandes suivantes :

- Connection à la base de données : **mysql -h localhost -u root** (le mot de passe vous sera demandé)
- Démarrer la base de données : **sudo systemctl start mysql**
- Redémarrer la base de données : **sudo systemctl restart mysql**
- Arrêter la base de données : **sudo systemctl stop mysql**
-

5.2 - Application web

L'application est gérée par Supervisor par le biais de son fichier de configuration qui lancera automatiquement Unicorn.

On lancera le serveur grâce à la commande: **sudo supervisorctl start oc_pizza**

On peut arrêter le serveur grâce à la commande : **sudo supervisorctl stop oc_pizza**

Nous pouvons vérifier également que supervisor est bien en route par la commande : **sudo supervisorctl status**

ALLER SUR SITE WEB



6 - PROCÉDURE DE MISE À JOUR

6.1 - Base de données

En ce qui concerne la mise à jour de notre base de données, nous avons programmé une mise à jour automatique par le biais de CRON, qui effectuera une mise à jour automatique hebdomadairement, nous l'avons programmé pour le dimanche soir.

Mais , si éventuellement , vous souhaitez effectuer une mise à jour manuelle, cela est possible en effectuant la commande **python manage.py updatedb**

6.2 - Application web

Afin d'effectuer la mise à jour de l'application web, il faudra utiliser la commande :

GIT PULL PLUS PROCEDURE DEPLOIEMENT

python manage.py update_ocpizza



7 - SUPERVISION/MONITORING

7.1 - Supervision de l'application web

Afin de tester que l'application web est toujours fonctionnelle, nous avons mis en place une surveillance des logs, de l'application, vous serez averti directement par mail, dans le cas d'une anomalie.

Puis, Digital Ocean, permet de surveiller les performances techniques de notre application, vous serez également averti dans le cas d'une surutilisation des performances et dans ce cas, il nous faudra upgrader les caractéristiques techniques de notre serveur virtuel.



8 - PROCÉDURE DE SAUVEGARDE ET RESTAURATION

Cron va créer un fichier de sauvegarde chaque minuit avec date jour du Back Up

Voici les commandes à utiliser :

Python manage.py backup (sert à faire une restauration)

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



9 - GLOSSAIRE

Base de données	La base de données est au centre des dispositifs informatiques de collecte, mise en forme, stockage et utilisation d'informations.
Variables d'environnement	les variables d'environnement sont des variables dynamiques utilisées par les différents processus d'un système d'exploitation (Windows , Unix...). 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.