|  |
| --- |
| **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

3.1.1.1 -Caractéristiques techniques 6

3.1.2 -Serveur Web 6

3.1.2.1 -Caractéristiques techniques 6

3.1.3 -Serveur de Batches 6

3.1.4 -Serveur de Fichiers 6

3.2 -Bases de données 6

3.3 -Web-services 6

4 -Procédure de déploiement 7

4.1 -Déploiement de l'Application Web 7

4.1.1 -Artefacts 7

4.1.2 -Variables d'environnement 7

4.1.3 Configuration 7

4.1.4 -Ressources 8

4.1.5 -Vérifications 8

4.2 -Déploiement de l'Application Web 9

4.2.1 -Artefacts 9

4.2.2 -Environnement de l’application web 9

4.2.2.1 -Variables d’environnement 9

4.2.3 -Répertoire de configuration applicatif 9

4.2.4 -DataSources 9

5 -Procédure de démarrage / arrêt 11

5.1 -Base de données 11

5.2 -Application web 11

6 -Procédure de mise à jour 12

6.1 -Base de données 12

6.2 -Application web 12

7 -Supervision/Monitoring 13

7.1 -Supervision de l’application web 13

8 -Procédure de sauvegarde et restauration 14

# Versions

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

# Introduction

## 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.

## Références

Pour de plus amples informations, se référer :

1. **DCT – 1.0** : Dossier de conception technique de l'application.
2. **DE – 1.0 :** Dossier de l’exploitation de l’application.
3. **DCF – 1.0 :** Dossier de conception fonctionnelle de l’application.
4. **PVL – 1.0 :** Dossier de PV Livraison.

# Pré-requis

## Système

### **Serveur de Base de données**

Le serveur de base de données est hebergé par Digital Ocean avec Ubuntu 18.04.3 et MySQL 8.0.19.

#### 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 performances pourront toujours être changer éventuellement.

### **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 WGSI inclus chez Django et le serveur web sera surveillé par Supervisor 4.2.0

#### 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 performances pourront toujours être changer éventuellement.

### **Serveur de Batches**

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

## Bases de données

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

* DCT—1.0 : version 1.0

## 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

# Procédure de déploiement

## Déploiement des Batches

### 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](mailto:ocpizza@305.190.445.87) 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 présent sur notre serveur de production.

### **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 |

Définissez les variables d’environnement nécessaires en remplacant les XXXXX

### **Configuration**

Voici les différents fichiers de configuration :

* **\_\_init\_\_.py** : fichier de configuration en local
* **Production.py** : fichier de configuration de l'application en production

### **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

...

# Procédure de démarrage / arrêt

## 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 systemclt start mysql**
* Redémarrer la base de données : **sudo systemctl restart mysql**
* Arreter la base de données : **sudo systemctl stop mysql**

## Application web

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

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**

# Procédure de mise à jour

## 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**

## Application web

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

**python manage.py update\_ocpizza**

# Supervision/Monitoring

## 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, permets lui de surveille 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 caracteristiques techniques de notre serveur virtuel.

# Procédure de sauvegarde et restauration

Concernant la sauvegarde et la restauration, nous avons également comme pour la mise à jour de la base de données, programmer une tâche cron hebdomadaire afin que la sauvegarde de l’application s’execute automatiquement.