OC Pizza

Réalisation d'un système de gestion pour un ensemble de pizzerias

Dossier d'exploitation

Version 1.0

Auteur
Guillaume OJARDIAS
Analyste Programmeur

IT Consulting & Development

www.developement.fr

8 rue Mozzarella 75001 PARIS
Tél : 01 02 03 04 05 – E-mail : contact@development.fr

S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Paris
SIREN 999 999 999 – Code APE : 6202A

TABLE DES MATIÈRES

8 rue Mozzarella 75001 PARIS
Tél : 01 02 03 04 05 – E-mail : contact@development.fr

S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Paris
SIREN 999 999 999 – Code APE : 6202A

1 - Versions

Auteur	Date	Description	Version
GO	xx/09/20	Création du document	1.0

2 - Introduction

2.1 - Objet du document

Le présent document constitue le dossier d'exploitation de l'application pour le système de gestion d'un ensemble de pizzerias.

L'objectif du document est de...

2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants :

1. Projet 9 - Dossier de conception fonctionnelle.
2. Projet 9 - Dossier de conception technique.
3. Projet 9 - PV de livraison.

3 - Pré-requis

3.1 - Système

L'ensemble de l'application utilisera un serveur ayant les caractéristiques minimales suivantes :

- Système d'exploitation : Linux / **Ubuntu** version **20.04**.
- Disque SSD de **25 GB**.
- **8 GB** de mémoire.
- **2** processeurs.
- **4 TB** de bande passante par mois.

Sur ce serveur, un compte utilisateur ***adminOcPizza*** sera créé avec les droits d'administrateur. Ce compte sera utilisé pour la mise en place et la maintenance de l'application.

Pour assurer le déploiement de l'application sur ce serveur, les packages suivants seront requis :

- **GIT** en version **2.28.0** minimum.
- **Python** en version **3.8** minimum
  - **Virtualenv** en version **20.0.35** minimum
- **OpenSSH** en version **8.4** minimum
- **UFW** en version **0.36** minimum
- **Certbot** en version **0.40.0** minimum
- **Supervisord** en version **4.1.0** minimum.

Le fournisseur de ce serveur sera Digital Ocean : https://digitalocean.com.

3.1.1 - Serveur de base de données

Le serveur de base de données sera un serveur virtuel ayant les caractéristiques suivantes :

- **PostgreSQL**.
- Version **13** minimum.

Ce serveur sera configuré de la manière suivante :

- Une base de données nommée **OcPizza.**
- L'utilisateur ***adminOcPizza*** aura un accès administrateur sur cette base de données.

3.1.2 - Serveur WEB

Le serveur WEB sera un serveur virtuel ayant les caractéristiques suivantes :

- Serveur **NGINX.**
- Version **1.20** minimum.

UFW devra être configuré pour autoriser **NGINX** en connexions entrantes et sortantes. **Certbot** sera utilisé pour générer des certificats **SSL** conformes et assurer la bonne configuration du serveur **NGINX**.

Le fichier de configuration est placé dans `/etc/nginx/sites-available/oc-pizza`. Un lien symbolique est créé entre ce fichier et le dossier `/etc/nginx/sites-enabled` pour activer le serveur.

3.1.3 - Serveur d'application

Le serveur d'application sera un serveur virtuel ayant les caractéristiques suivantes :

- Serveur **Gunicorn.**
- Version **20.0.4** minimum.

Le fonctionnement de ce serveur d'application sera géré grâce à l'outil **Supverisord**. En charge de son démarrage et redémarrage automatique.

Le serveur sera installé via grâce au fichier de `requirements.txt` du repository et la commande `pip install -r`.

Un fichier de configuration, **Supervisord** est créé dans le répertoire `/etc/supervisor/conf.d/oc-pizza.conf` pour que **Supervisord** puisse démarrer le serveur **Gunicorn**.

4 - Procédure de déploiement

4.1 - Déploiement de l'application web

4.1.1 - Environnement de l'application

Dans un premier temps, l'utilisateur ***adminOcPizza*** se place dans son répertoire :

```
cd /home/adminOcPizza
```

Les fichiers de l'application sont clonés depuis le repository GitHub :

```
git clone https://github.com/oc-pizza/oc-pizza oc-pizza
```

L'utilisateur se place ensuite dans le répertoire de l'application :

```
cd ./oc-pizza
```

Un environnement virtuel est créé dans le répertoire du projet :

```
virtualenv venv   # Création de l'environnement virtuel dans le répertoire venv/

source venv/bin/activate  # Activation de l'environnement virtuel
```

Les dépendances de l'application sont installées :

```
pip install -r requirements.txt
```

Il ne manque plus qu'a lancer la collecte des fichiers statiques :

```
python manage.py collectstatic
```

> NOTE : attention l'environnement virtuel doit être activé pour installer les dépendances du projet et utiliser `manage.py`

4.1.2 - Configuration de l'application

Les variables d'environnement de l'application se trouvent dans un fichier `.env` à la racine du répertoire contenant les fichiers de l'application. Un fichier d'exemple, `env_example`, est fourni avec le code.

De plus, la configuration des paramètres de l'application se trouve dans le fichier `config.py` lui aussi à la racine du projet.

4.1.3 - Initialisation de la base de données

Vérifier dans le fichier `.env` que les paramètres de la base de donnée sont correctement renseignés.

Pour initialiser la base de données :

```
python manage.py migrate        # Crée les tables dans la base de données
```

Cette étape est aussi l'occasion pour créer un super-utilisateur, pour cela, il suffit d'utiliser la commande suivante et de renseigner les champs qui s'affichent.

```
python manage.py createsuperuser  # Création d'un super-utilisateur
```

NOTE : attention l'environnement virtuel doit être activé pour pouvoir exécuter `manage.py`

5 - Procédure de démarrage/arrêt

5.1 - Serveur d'application

Pour rappel, le démarrage et redémarrage du serveur d'application est géré automatiquement par ***Supervisord***. Cependant, si cela s'avère nécessaire l'utilisateur peut exécuter les commandes suivantes :

```
sudo supervisorctl start oc-pizza
```

Pour stopper le serveur, l'utilisateur doit exécuter la commande suivante :

```
sudo supervisorctl stop oc-pizza
```

Pour contrôler l'état du processus, l´utilisateur peut exécuter la commande suivante :

```
sudo supervisorctl status oc-pizza
```

5.2 - Serveur Web

Pour démarrer le serveur web Nginx, l'utilisateur doit exécuter la commande suivante :

```
sudo systemctl start nginx
```

Pour stopper le serveur, l'utilisateur doit exécuter la commande suivante :

```
sudo systemctl stop nginx
```

Pour contrôler l'état du serveur, l'utilisateur peut utiliser la commande suivante :

```
sudo systemctl status nginx
```

5.3 - Base de données

Pour démarrer la base de données, l'utilisateur doit exécuter la commande suivante :

```
sudo systemctl start postgresql
```

Pour stopper la base de données, l'utilisateur doit exécuter la commande suivante :

```
sudo systemctl stop postgresql
```

Pour contrôler l'état de la base de données, l'utilisateur peut utiliser la commande suivante :

```
sudo systemctl status postgresql
```

6 - Procédure de mise à jour

6.1 - Mise à jour de l'application

Mise à jour des fichiers source depuis le dépôt GitHub :

```
git pull origin master
```

On met à jour les fichiers statiques :

```
python manage.py collectstatic
```

NOTE : attention l'environnement virtuel doit être activé pour pouvoir exécuter `manage.py`

6.2 - Mise à jour de la base de données

Une fois les fichiers mis à jour il est recommandé de toujours migrer la base de données afin de s'assurer que celle-ci soit toujours dans le dernier état :

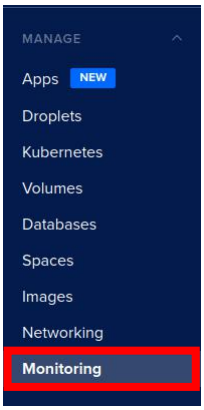
```
python manage.py migrate
```

NOTE : attention l'environnement virtuel doit être activé pour pouvoir exécuter `manage.py`

7 - Supervision et monitoring

7.1 - Surveillance du serveur

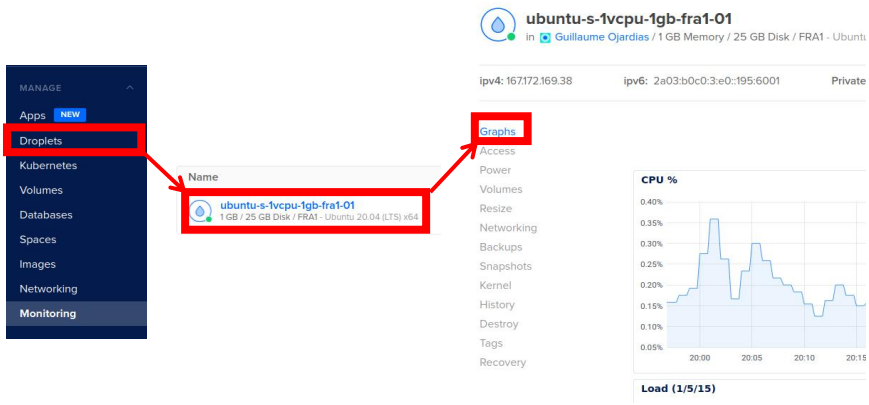
La surveillance du serveur se fait depuis l'espace client de Digital Ocean dans la partie ***Monitoring*** du dashboard :



Des alertes ont déjà était paramétrées pour les éléments suivants :

- CPU;
- utilisation du disque;
- utilisation de la mémoire.

Les métriques en temps réel peuvent être consultées dans la partie ***Graph*** des détails du droplet Digital Ocean :



7.2 - Surveillance de l'application

8 - Procédure de sauvegarde et restauration

8 rue Mozzarella 75001 PARIS
Tél : 01 02 03 04 05 – E-mail : contact@development.fr

S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Paris
SIREN 999 999 999 – Code APE : 6202A

9 - Glossaire

Persona	Utilisateur fictif utilisé dans la démarche de conception
Cas d'utilisation	Unité représentant une fonctionnalité d'un système visible de l'extérieur (exemple : se connecter)
Diagramme de Cas d'utilisation	Diagramme mettant en œuvre les relations entre les acteurs d'un système et les différents cas d'utilisation le composant.
Package	Ensemble de cas d'utilisation (exemple : interface client)
Diagramme de Packages	Diagramme mettant en œuvre les relations entre les acteurs d'un système et les différents packages le composant.
Workflow	Parcours d'un utilisateur, pour un cas d'utilisation donné, ici représenté sous forme d'un diagramme
Scénario nominal	Scénario principal d'un cas d'utilisation
Scénario d'exception	Scénario alternatif d'un cas d'utilisation

8 rue Mozzarella 75001 PARIS
Tél : 01 02 03 04 05 – E-mail : contact@development.fr

S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Paris
SIREN 999 999 999 – Code APE : 6202A