OC Pizza

Projet 9 - Documentez votre système de gestion de pizzeria

Dossier d'exploitation

Version 1

Auteur
Jérôme Divry
Développeur d'application junior

Table des matières

1 - Versions

Auteur	Date	Description	Version
Jérôme Divry	14/09/21	Création du document	1

2 - Introduction

2.1 - Objet du document

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

L'objectif de ce document est de préciser l'environnement, ses spécifications et de détailler la mise en production de l'application.

2.2 - Références

Pour de plus amples informations, se référer :

1. **DCT - OC Pizza** : Dossier de conception technique de l'application
2. **DCF** - **OC Pizza**: Dossier de conception fonctionnelle de l'application hébergé sur github.

3 - Pré-requis

3.1 - Système

3.1.1 - Serveur de Base de données

MySql 8.0.26 est le SGBD hébergeant la base *ocpizza*

3.1.2 - Serveur Web

Hébergé par Digital Ocean qui propose des VPS qui fonctionnent essentiellement sous Linux. Cette application fonctionne dès lors sous Ubuntu 20.04 (dernière version LTS).

3.2 - Bases de données

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

- **MySQL :** mysql Ver 8.0.26-0ubuntu0.20.04.2 for Linux on x86_64 ((Ubuntu))
- Le schémas ocpizza est créé et accessible via mysql -u ocp_admin -p

4 - Procédure de déploiement

4.1 - Déploiement de l'application

La connexion SSH au droplet ocpizza à cette adresse :

```
ssh ocpizza_admin@123.132.77.57
```

Le mot de passe est fourni dans l'email crypté.

4.1.1 - Clone du projet

Dans /home/ocpizza_admin/ :

```
git clone ocpizza_project.git
```

4.1.2 - Configuration de Gunicorn

Le serveur WSGI Gunicorn est installé avec pip dans l'environnement virtuel (venv), il fait partie du fichier requirements.txt.

Sa configuration se trouve dans le fichier wsgi.py :

```
gunicorn ocp.wsgi:application
```

4.1.3 - Configuration de Supervisor

Supervisor est actif dès son installation. Il permet de contrôler/surveiller l'ensemble des processus dans un fichier `.conf` et de les démarrer automatiquement.

Son fichier de configuration se trouve ici :

```
~/etc/supervisor/conf.d/ocpizza-gunicorn.conf
```

et dans ce fichier se trouvent les commande dédiées au démarrage/redémarrage de gunicorn, le répertoire à partir duquel est activé la commande, l'utilisateur qui a les droits pour son exécution (ocpizza_admin).

Lancement des processus de Supervisor :

```
sudo supervisorctl reread
sudo supervisorctl update
sudo supervisorctl status # vérifie le statut
```

4.1.4 - Accès à l'administration du site

Via les infos de connexion transmises dans l'email :

https://ocpizza.fr/admin

4.1.5 - Variables d'environnement

Voici les variables d'environnement reconnues par l'application OCPizza :

Nom	Obligatoire	Description
~/home/ocpizza_admin/ocpizza_project	oui	Répertoire racine de l'installation de l'application
ADMIN_NAME	oui	Nom de l'administrateur
ADMIN_PASSWORD	oui	Mot de pass de l'administrateur
DB_NAME	oui	Nom de la base de données
DB_USER	oui	Nom de l'utilisateur de la base de données
DB_PASSWORD	oui	Mot de passe de la base de données
DB_HOST	oui	Hôte de la base de données
DB_PORT	oui	Port de la base de données
SECRET_KEY	oui	Clé secrète de Django pour l'application ocpizza
STRIPE_API_KEY	oui	Clé d'api secrète (production)
GOOGLE_KEY	oui	Clé d'api secrète
SENTRY_DSN	oui	Data Source Name de Sentry

situées dans `/etc/supervisor/conf.d/ocpizza-gunicorn.conf`

En local, elles se retrouvent également dans le fichier `/ocpizza_project/.env`

4.1.6 - Fichiers de configuration

Fichier	Description
/ocpizza_project/.gitignore	Exclusion des fichiers spécifiés dans le dépot git
/ocpizza_project/.travis.yml	Configuration des tests avant déploiement
/ocpizza_project/app/settings.py	Fichier de configuration principal de l'application
/etc/supervisor/conf.d/ocpizza-gunicorn.conf	Configure gunicorn et contient les variables d'environnement
/etc/nginx/sites-enabled/ocpizza	Configuration de nginx

5 - Procédure de démarrage / arrêt

5.1 - Base de données

Les services mysql sont actifs. Voici les commandes pour en vérifier le statut, les démarrer et les arrêter :

```
sudo service mysql status
sudo service mysql start
sudo service mysql stop
```

5.2 - Application web

Le démarrage et/ou l'arrêt de l'application dépend du statut du serveur nginx.

Il est nécessaire de redémarrer le serveur après une modification de sa configuration.

```
sudo service nginx start
sudo service nginx stop
sudo service nginx restart
```

Supervisor est installé et prend en charge les services Gunicorn (vérification du statut et redémarrage automatique).

6 - Procédure de mise à jour

6.1 - Base de données

La base de données MySQL est mise à jour via la commande personnalisée Django :

```
python3 manage.py createdb
```

Cette commande est activée par une tâche Cron paramétrée pour lancer la procédure de mise à jour une fois par semaine (les dimanche à 3h12) :

```
~/home/ocpizza_admin/update_db.sh
```

6.2 - Application web

La mise à jour de l'application se fait à la racine du projet :

```
~/home/ocpizza_admin
```

Via git avec la commande :

```
git pull origin main
```

Puis en cas de modification des fichiers models.py (modifications de la structure de la base de données) au sein de l'application il est nécessaire d'exécuter :

1. pour faire la vérification des modifications de la base et créer les migrations

```
python3 manage.py makemigrations
```

2. pour les appliquer

```
python3 manage.py migrate
```

Et pour que le serveur applique à son tour la mise à jour, il faut relancer gunicorn via supervisor :

```
sudo supervisorctl reread
sudo supervisorctl update
sudo supervisorctl status # vérifie le statut -> RUNNING = actif
```

7 - Supervision/Monitoring

7.1 - Supervision de l'application web

Afin de tester que l'application web est toujours fonctionnelles, le monitoring du droplet sous Digital Ocean est actif:

curl -sSL https://agent.digitalocean.com/install.sh | sh

Il donne accès à l'interface d'administration et permet de configurer des alertes :

Monitoring > Create alert policy

Sa configuration actuelle active des alertes en priorité pour le CPU, la RAM et la bande passante.

Un How to est disponible https://docs.digitalocean.com/products/monitoring/how-to/install-agent/

Sentry est implémenté et initialisé dans settings.py pour visualiser les événements au sein de l'application Django.

Le How To dédié : https://docs.sentry.io/platforms/python/guides/django/

8 - Procédure de sauvegarde et restauration

A la racine du projet, le fichier save_db.sh lance l'éxécution d'une tâche Cron qui permet la sauvegarde de la base de données, après la mise à jour, les dimanche à 4h12.

9 - Glossaire

VPS	Virtual Private Server ou serveur virtuel privé
SGBD	Système de gestion de base de données