OC-Pizza

Gestion de pizzeria

Dossier d'exploitation

Version V1.0.0

Auteur
Faure Quentin
Analyste-Programmeur

Table des matières

1 - Versions

Auteur	Date	Description	Version
Quentin Faure	19/04/23	Création du document	0.0.1
Quentin Faure	20/05/23	Modification des champs	1.0.0

2 - Introduction

2.1 - Objet du document

Le présent document constitue le dossier d’exploitation de l'application de Gestion de pizzeria.

L'objectif de ce document est de faciliter l'exploitation de l'application de gestion de pizzeria et pour assurer sa stabilité à long terme. Ce dossier d'exploitation est à l'attention de l'équipe technique du client afin de lui fournir toutes les informations nécessaires pour réagir en cas de problème et pour assurer l'exploitation efficace de la solution proposée.

2.2 - Références

Pour de plus amples informations, se référer :

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

3 - Pré-requis

3.1 - Système

L'application sera déployée sur un serveur Linux avec le système d'exploitation Ubuntu Server sans interface graphique.

3.1.1 - Hébergement

L'hébergement se fera via un Droplet DigitalOcean et le nom de domaine sera réservé via IONOS pour : oc-pizza.fr

3.1.2 - Serveur Web

Le serveur est un serveur dédié, ce qui permet une flexibilité, des performances et un contrôle supérieur pour l'application.

3.1.2.1 - Caractéristiques techniques

- Système d'exploitation : Ubuntu Server 20.04 LTS
- Version de Gunicorn : 20.1.0
- Version de Python : 3.9
- Capacité de stockage : 200 Go NVMe SSD
- RAM : 16 Go
- Processeur : Intel Xeon E5-2650 v4
- Transfert de donnée : 6TB / Mois

3.2 - Bases de données

Le serveur de base de données hébergeant la base de données de l'application sera sur un autre serveur dédié, toujours hébergé chez DigitalOcean

3.2.1 - Serveur de Base de données

La base de données sera gérée par le SGBD MySQL (v8.0.33).

3.2.1.1 - Caractéristiques techniques

- Système d'exploitation : Ubuntu Server 20.04 LTS
- Version de MySQL : 8.0.26
- Capacité de stockage : 50 Go NVMe SSD
- RAM : 16 Go
- Processeur : Intel Xeon E5-2650 v4

3.3 - Web-services

Les web services suivants doivent être accessibles et à jour :

- Google Maps Api : Pour la géolocalisation des points de vente et des trajets de livraison.
- SQUARE: Une plateforme de paiement tout-en-un qui propose des solutions de paiement en ligne, en personne et sur mobile, offrant une gamme complète de fonctionnalités de paiement, y compris des terminaux de paiement, des solutions de point de vente (POS) et des options de traitement des paiements en ligne.

3.4 - Autres Ressources

Nous conseillons l'utilisation des outils suivants:

- Sentry: Pour les tâches de monitoring de l'applicaiton Django,
- New Relic : Pour les tâches de monitoring des performances du serveur.
- Travis CI: Pour la maintenance et l'intégration continue.

4 - Procédure de déploiement

4.1 - Déploiement du serveur d'application

4.1.1 - Configuration

L'étape de configuration concerne la mise en place du mot de passe et de l'identifiant pour la connexion SSH sur votre serveur d'application. Dans le contexte où vous utilisez un droplet DigitalOcean, voici comment procéder :

- Connectez-vous à votre compte DigitalOcean et accédez à la section Droplets.
- Sélectionnez le droplet correspondant à votre serveur d'application.
- Dans la page du droplet, recherchez la section "Access" ou "Accès" qui contient les informations relatives à la connexion SSH.
- Si vous n'avez pas déjà une paire de clés SSH configurée, vous pouvez en créer une en cliquant sur "Add SSH Key" ou "Ajouter une clé SSH". Suivez les instructions pour générer et ajouter votre clé SSH publique.
- Une fois que vous avez configuré votre clé SSH, vous pouvez vous connecter à votre droplet à l'aide d'un client SSH (par exemple, PuTTY sur Windows ou Terminal sur macOS/Linux) en utilisant l'adresse IP de votre droplet et votre identifiant SSH (qui est généralement "root" pour les droplets DigitalOcean).
- Lors de votre première connexion, vous serez invité à changer le mot de passe de votre identifiant SSH. Suivez les instructions pour définir un nouveau mot de passe sécurisé.

4.1.1.1 - Le code source

Le code source de l'application est à télécharger depuis la branche 'main' du lien Github fourni au client. Elle est au format .zip et est donc à dézipper avant utilisation.

4.1.1.2 - Packaging

Toutes les commandes suivantes sont à exécuter dans la console du serveur.

Mettons à jour notre système:

```
$ sudo apt update
```

Installez ensuite les dépendances pour les scripts JavaScript de l'application React :

```
$ sudo apt-get install nodejs
$ sudo apt-get install npm
```

Installons ensuite les librairies Python pour l'application Django ainsi que le serveur Nginx :

```
$ sudo apt-get install python3 python3-pip python3-dev
$ sudo apt-get install nginx supervisor
```

4.1.2 - Déploiement de l'Application Web

Pour la suite, il faudra se déplacer à la racine du dossier (de l'application) dézippé.

4.1.3 - Environnement virtuel et dépendances

Il convient ensuite d'initialiser un enviroonnement virtuel. On installe donc pipenv:

```
pip install pipenv
pipenv shell
```

Nous installons ensuite les dépendances de l'application avec la commande:

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

4.1.4 - Variables d'environnement

Dans un soucis de sécurité des seveurs et de l'application, nous isolerons les informations d'authentification et autres clefs au travers de variabels d'environnements.

Nom	Description
'DJANGOSECRETKEY'	Clef renseignée dans le fichier settings.py de l'application Django
'GOOGLEAPIKEY'	Clef fournie par l'API de Google lors de la création du projet sur leur plateforme de localisation.

4.1.5 - Serveurs Nginx/Gunicorn

Pour le fichier de configuration de l'applicatif web vous devez :

- Déplacer le fichier conf de NGINX dans le repertoire des fichiers conf:

```
mv ocpizza /etc/nginx/sites-available
```

- Créer un lien symbolique dans site-enable

```
ln -s /etc/nginx/sites-available/ocpizza /etc/nginx/sites-enabled
```

- Déplacer le fichier conf de Supervisor dans le repertoire des fichiers conf

```
mv ocpizza.conf /etc/supervisor/conf.d/
```

-ATTENTION- Il faut changer le chemin vers l'exécutable de Gunicorn dans le fichier de configuration car l'application est installée dans l'environnement virtuel créé pour python et le projet. Copier le chemin qui se trouve ici(en changeant les valeurs en gris):

```
$ ls /home/nom_user/.virtualenvs/nom_environnement_virtuel/bin/gunicorn
```

et le reporter dans la ligne command du fichier de configuration.

Le fichier de configuration de Supervisor va dire à Gunicorn de démarrer l'application Django ocpizza et de la redémarrer automatiquement en cas d'arrêt.

Il conviendra ensuite de redémarrer les deux services pour qu'ils prennent en compte les modifications:

```
$ sudo service nginx reload
$ sudo supervisorctl update
$ sudo supervisorctl reload
```

4.2 - Déploiement du serveur de base de données

4.2.1.1 - Packaging

Toutes les commandes suivantes sont à exécuter dans la console du serveur.

Mettons à jour notre système:

```
$ sudo apt update
```

Installez et démarrons ensuite le SGBD:

```
$ sudo apt install mysql-server
$ sudo systemctl start mysql.service
```

Pour les nouvelles installations de MySQL, vous devrez exécuter le script de sécurité inclus dans le SGBD. Ce script modifie certaines des options par défaut les moins sûres pour des choses comme les connexions root distantes et les sample users.

Exécutez le script de sécurité avec sudo :

```
$ sudo mysql_secure_installation
```

Vous serez alors guidé à travers une série d'invites où vous pourrez apporter quelques modifications aux options de sécurité de votre installation MySQL. La première invite vous demandera si vous souhaitez configurer le plugin Validate Password, que vous pouvez utiliser pour tester la solidité de votre mot de passe MySQL.

Si vous choisissez de mettre en place le plugin Validate Password, le script vous demandera de choisir un niveau de validation du mot de passe. Le niveau le plus fort - que vous sélectionnez en entrant 2 - exigera que votre mot de passe comporte au moins huit caractères, dont un mélange de majuscules, de minuscules, de chiffres et de caractères spéciaux.

Que vous choisissiez ou non de configurer le plugin Validate Password, l'invite suivante vous demandera de définir un mot de passe pour l'utilisateur root de MySQL. Entrez et confirmez le mot de passe sécurisé de votre choix :

```
Output
Please set the password for root here.
New password:
Re-enter new password:
```

Si vous avez utilisé le plugin Validate Password, vous recevrez des commentaires sur la force de votre nouveau mot de passe. Ensuite, le script vous demandera si vous voulez continuer avec le mot de passe que vous venez de saisir ou si vous voulez en saisir un nouveau. En supposant que vous êtes satisfait de la force du mot de passe que vous venez d'entrer, saisissez Y pour poursuivre le script :

À partir de là, vous pouvez appuyer sur Y puis sur ENTER pour accepter les valeurs par défaut pour toutes les questions suivantes. Cela supprimera les utilisateurs anonymes et la base de données de test, désactivera les connexions root à distance, et chargera ces nouvelles règles afin que MySQL respecte immédiatement les modifications que vous avez apportées.

5 - Procédure de démarrage / arrêt

5.1 - Base de données

Par défaut, le serveur de base de données est démarré.

Pour le stopperon peut utiliser la commande:

```
sudo systemctl stop mysql.service
```

Pour démarrer le service :

```
sudo systemctl start mysql.service
```

5.2 - Application web

Comme nous avons utilisé Supervisor pour gérer le redémarrage automatique de l'application, il conviendra de commenter la ligne autorestartdu fichier de configuration de Supervisoravant d'arrêter l'application.

6 - Procédure de mise à jour

6.1 - Application web

– ATTENTION -

Toute modification ou mise à jour sur un serveur en production est à préparer en amont avec l'utilisation de l'outil d'intégration continue afin de s’assurer de la bonne compatibilité des différentes versions de logiciels et de garder une continuité de service.

En fonctionnant avec l'outil d'intégration continue Travis CI, chaque nouvel envoi au repository Github de l'application passera la batterie de test mise en place avant d'accepter son envoi en production.

7 - Supervision/Monitoring

7.1 - Supervision de l'application web

Afin de tester que l'application web est toujours fonctionnelle, nous utiliserons deux outils de monitoring déjà présentés:

7.1.1 - Sentry

Pour le monitoring de l'application: en se connectant sur l'interface web de l'outil, https://sentry.io/, on peut consulter les logs et le suivi de fonctionnement de l'application.

7.1.2 - New Relic

Pour le monitoring du serveur: en seconnectant sur l'interface web de l'outil, https://one.eu.newrelic.com/ , on peut consulter les logs et l'activité du serveur en temps réel (utilisation de la charge CPU, des disques durs, de la bande passante…).

8 - Procédure de sauvegarde et restauration

8.1 - Sauvegarde:

Une tâche CRON va exécuter une sauvegarde quotidienne de la base de données avec une rotation sur 10 jours pour que les fichiers ne prennent pas trop de place.
Pour cela on utilise l'outil en ligne de commande 'mysqldump' qui permet de sauvegarder une base de données MySQL en créant une copie complète ou partielle de la base de données au format SQL.

8.2 - Restauration:

Pour restaurer une base de données depuis une sauvegarde il faut taper la commande:

```
mysql -u root -p pizzeria_management < [chemin_vers_fichier_sauvegarde.sql]
```

9 - Glossaire
