OC Pizza

Application Web

Dossier d'exploitation

Version 1.1

Auteur
Eri Schön
Développeur Python

IT Consulting
www.itc.fr

4 rue Bariton – 06.15.57.89 – contact@itc.fr
S.A.R.L. au capital de 1 000 € – SIREN 999 999 999 – RCS de Paris - Code APE : 6202A

TABLE DES MATIÈRES

1 - Versions

Auteur	Date	Description	Version
Eri Schön	17/05/21	Création du document	1.0
Eri Schön	18/05/21	Ajout du contenu	1.1

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 les différentes caractéristiques du système nécessaire à l'exécution de l'application.

Il établit également les versions et outils utilisés pour l'exploitation de l'application OC PIZZA.

2.2 -Références

Pour de plus amples informations, se référer :

1. **PDOCPizza_01** : Dossier de conception fonctionnelle de l'application
2. **PDOCPizza_02** : Dossier de conception technique de l'application

3 - Pré-requis

3.1 -Système

L'ensemble de l'application Web sera hébergée sur un serveur Linux Ubuntu 20.04 chez DigitalOcean (hébergement de type IaaS).

Au préalable au déploiement un Setup du serveur sera effectué :

1 - Logging in as root

```
$ ssh -i /Users/ocpizza/.ssh/id_digitalocean root@137.79.46.197
```

2 - Creating a New User

```
$ adduser ocpizza
```

3 - Granting Administrative Privileges

```
$ usermod -aG sudo ocpizza
```

4 - Setting Up a Basic Firewall

```
$ ufw app list
$ ufw allow OpenSSH
$ ufw enable
$ ufw status
```

5 - Enabling External Access for the Regular User

```
$ rsync --archive --chown=ocpizza:ocpizza ~/.ssh /home/ocpizza
```

6 - Connect to the server with regular user

```
$ ssh -i /Users/ocpizza/.ssh/id_digitalocean ocpizza@137.79.46.197
```

Nous installerons ensuite les prérequis au déploiement :

```
$ sudo apt update
$ sudo apt upgrade
$ sudo apt dist-upgrade

$ sudo apt install python3-pip python3-dev libpq-dev
$ sudo apt install postgresql postgresql-contrib
$ sudo apt install nginx curl
```

3.2 - Web-Services

Les service suivant seront utilisés :

- Stripe
- Paypal
- Quartix

4 - Procédure de déploiement

4.1 - Déploiement de l'Application Web

4.1.1 - Création de la Base de donnée

```
sudo -u postgres psql

CREATE DATABASE ocpizza;
CREATE USER ocpizza WITH PASSWORD '<unpassword>';

ALTER ROLE ocpizza SET client_encoding TO 'utf8';
ALTER ROLE ocpizza SET default_transaction_isolation TO 'read committed';
ALTER ROLE ocpizza SET timezone TO 'UTC';

GRANT ALL PRIVILEGES ON DATABASE ocpizza TO ocpizza;
```

4.1.2 - Création d'un environnement virtuel et déploiement

Créer un répertoire ocpizza et se mettre dedans

```
$ mkdir ocpizza
$ cd ocpizza
```

Vérifier que git est installé

```
$ git --version
```

Cloner le repo

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

Installer pipenv

```
$ pip3 install --user pipenv
```

Créer le fichier pour les variables d'environnement et y mettre les variables d'environnement de production

```
$ sudo nano .env
```

Créer le répertoire pour l'environnement virtuel

```
$ mkdir .venv
```

Lancer l'environnement virtuel et installer les requirements

```
$ pipenv shell
$ pipenv install
```

Lancer la migration

```
$ python manage.py migrate
```

Créer un super user

```
$ python manage.py createsuperuser
```

Récupérer les staticfiles

```
$ python manage.py collectstatic
```

Vérification :

```
$ sudo ufw allow 8000
$ python manage.py runserver 0.0.0.0:8000
$ http://137.79.46.197:8000/admin
```

4.1.3 - Installation de Gunicorn

Bind et check

```
$ gunicorn --bind 0.0.0.0:8000 core.wsgi
```

l'url http://137.79.46.197:8000 doit toujours être active

Création d'un service et d'un socket

```
$ sudo nano /etc/systemd/system/gunicorn.socket
$ sudo nano /etc/systemd/system/gunicorn.service
```

Lancement

```
$ sudo systemctl start gunicorn.socket
$ sudo systemctl enable gunicorn.socket
```

4.1.4 - Lancement de Ngnix

```
$ sudo nano /etc/nginx/sites-available/core
```

On active

```
$ sudo ln -s /etc/nginx/sites-available/core /etc/nginx/sites-enabled
```

Tester la syntax

```
$ sudo nginx -t
```

Lancer Nginx

```
$ sudo systemctl restart nginx
```

Modifier le firewall

```
$ sudo ufw delete allow 8000
$ sudo ufw allow 'Nginx Full'
```

4.1.5 - Ressources

- R4121 : les variables d'environnement
- R4151 : le contenu du gunicorn.socket
- R4152 : le contenu du gunicorn.service
- R4153 : le contenu du paramétrage Nginx

5 - Procédure de démarrage / arrêt

5.1 - Base de données

Checker le status de PostgreSQL

```
$ sudo systemctl status postgresql
```

Démarrage de PostgreSQL

```
$ sudo systemctl start postgresql
```

Arrêter PostgreSQL

```
$ sudo systemctl stop postgresql
```

Relancer PostgreSQL

```
$ sudo systemctl reload mysql
```

5.2 - Serveur d’application

Recharger Gunicorn

```
$ sudo systemctl daemon-reload
```

Relancer Gunicorn

```
$ sudo systemctl restart gunicorn
```

Arrêter

```
$ sudo systemctl stop gunicorn
```

Status

```
$ sudo systemctl status gunicorn
```

5.3 - Serveur Web

Démarrer Nginx

```
$ sudo service nginx start
```

Status

```
$ sudo service nginx status
```

Arrêter Nginx

```
$ sudo service nginx stop
```

Recharger

```
$ sudo service nginx reload
```

6 - Procédure de mise à jour

6.1 -Système

Les packages du serveur sont à mettre à jour régulièrement :

```
$ sudo apt-get update
```

6.2 -Application web

Les mises à jour de l'application se feront via le repo Github dédié.

```
$ git pull
```

Si des mises à jour au niveau des fichiers statiques est à faire :

```
$ python manage.py collectstatic
```

Si des mises à jour au niveau du modèle sont à faire :

```
$ python manage.py migrate
```

7 - Supervision/Monitoring

7.1 - Monitoring du serveur

L'ensemble des outils de monitoring du serveur seront accessibles sur le dashboard de Digitalocean dans la section monitoring.

7.2 -Supervision de l'application web

La supervision de l'application sera accessible sur Sentry :

https://sentry.io/organizations/ocpizza/projects/

7.3 - Logs

Journal des process logs Nginx

```
$ sudo journalctl -u nginx
```

Journal des access logs Nginx

```
$ sudo less /var/log/nginx/access.log
```

Journal des error logs Nginx

```
$ sudo less /var/log/nginx/error.log
```

Journal des application logs Gunicorn

```
$ sudo journalctl -u gunicorn
```

Journal des sockets logs Gunicorn

```
$ sudo journalctl -u gunicorn.socket
```

8 - Procédure de sauvegarde et restauration

Une sauvegarde de la base de données sera effectuée automatiquement chaque jour à 1h am.

Si vous souhaitez en faire une manuellement :

```
$ sudo -u postgres psql
```

Sauvegarde

```
pg_dump ocpizza > sauvegarde_db.sql
```

Restauration

```
psql ocpizza < sauvegarde_db.sql
```

9 - Glossaire
