

OC Pizza

Système de gestion de pizzeria en ligne

Dossier d'exploitation

Version 1.0

Auteur

Clémence Robin
Analyste-programmeur

TABLE DES MATIÈRES

1 - Versions.....	3
2 - Introduction.....	4
2.1 - Objet du document.....	4
2.2 - Références.....	4
3 - Pré-requis.....	5
3.1 - Système.....	5
3.2 - Bases de données.....	5
3.3 - Web-services.....	5
4 - Procédure de déploiement.....	6
4.1 - Déploiement de l'application web.....	6
4.1.1 - Configuration de l'espace serveur de l'application.....	6
4.1.2 - Configuration de Supervisor et de Gunicorn.....	8
4.1.3 - Configuration de Nginx.....	8
4.1.4 - Configuration de Redis.....	9
4.1.5 - Lancement de l'application Angular.....	9
4.2 - Déploiement de la base de données.....	10
4.2.1 - Configuration de l'espace serveur de la base de données.....	10
5 - Procédure de démarrage / arrêt.....	11
5.1 - Base de données.....	11
5.2 - Application web.....	11
5.2.1 - Django.....	11
5.2.2 - Angular.....	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
7.2 - Supervision de la base de données.....	13
8 - Procédure de sauvegarde et restauration.....	14
8.1 - Sauvegarde et restauration de l'application web.....	14
8.2 - Sauvegarde et restauration de la base de données.....	14
9 - Glossaire.....	15

1 - VERSIONS

Auteur	Date	Description	Version
Clémence R.	15/09/2020	Création du document	1.0

2 - INTRODUCTION

2.1 - Objet du document

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

Il rassemble les informations dont l'équipe technique d'OC Pizza a besoin pour pouvoir assurer l'exploitation du système ainsi que les procédures pour démarrer, arrêter et superviser le système.

Ce document s'adresse à l'équipe technique d'OC Pizza.

2.2 - Références

Pour de plus amples informations, se référer :

1. **DCT – 1.0** : Dossier de conception technique de l'application
2. **DCF – 1.0** : Dossier de conception fonctionnelle de l'application

3 - PRÉ-REQUIS

3.1 - Système

L'application OC Pizza ainsi que la base de données sont hébergées sur des machines virtuelles Digital Ocean .

Le nom de domaine de l'application est le suivant : ocpizza.fr

3.2 - Bases de données

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

- **PostgreSQL** : version 12

3.3 - Web-services

Les web services suivants doivent être accessibles et à jour :

- **Paypal API**
 - Clé secrète : xrtsa=v@w&katuwxq&q@z^og^#4o4m!rz1mns#-nj&w=%+ddnw

4 - PROCÉDURE DE DÉPLOIEMENT

4.1 - Déploiement de l'application web

4.1.1 - Configuration de l'espace serveur de l'application

4.1.1.1 - Serveur dédié à l'application web

L'application est déployée sur un serveur privé virtuel (VPS) loué chez Digital Ocean.

4.1.1.2 - Système d'exploitation

Le système d'exploitation Debian sera déjà installé sur le serveur par Digital Ocean.

4.1.1.3 - Accès à distance

Deux clés SSH devront être créées : la clé privée et la clé publique. Elles permettront de se connecter au serveur distant.

4.1.1.4 - Librairies à installer avant l'application

Certaines librairies devront être installées afin de déployer l'application OC Pizza :

```
apt install git
apt install python3
apt install python3-pip
apt install python3-dev
apt install libpq-dev
apt install nginx
apt install supervisor
apt install python3-venv
apt install redis-server
[Installation de Nodejs , Angular et Firebase]
apt install software-properties-common
curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
sudo apt-get install nodejs
npm install -g @angular/cli@9
npm install -g firebase-tools
```

4.1.1.5 - Téléchargement de l'application OC Pizza

Après avoir configuré Git pour obtenir les droits d'accès au *repository* de l'application, il faut télécharger l'application (ici, dans le dossier *home*) :

```
git clone https://github.com/oc_pizza/oc_pizza_application
```

4.1.1.6 - Installation de l'environnement virtuel

Créer l'environnement virtuel avec la commande suivante (ici, dans le dossier *home*)

```
python3 -m venv venv_oc_pizza
```

4.1.1.7 - Installation des librairies de l'application

Activer l'environnement virtuel (ici, à partir du dossier *home*) :

```
source venv_oc_pizza/bin/activate
```

Toutes les librairies sont installées par la commande :

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

Il faut pour cela se trouver au même niveau d'arborescence que le fichier *requirements.txt*.

4.1.1.8 - Edition de *production.py*

Éditer le fichier de configuration de production pour ajouter l'adresse IP du serveur de l'application ainsi que le nom de domaine dans "ALLOWED_HOSTS".

Une fois le serveur de base de données lui aussi déployé (voir section [4.2 - Déploiement de la base de données](#)), ajouter l'adresse IP du serveur de base de données pour "HOST" ainsi que le mot de passe correspondant à l'utilisateur dans "PASSWORD".

```
from . import *

SECRET_KEY = '$vf@$z6x3u)=v4cj5dig08o6d16d8*xtais+z(fe_@o*x9jd=m'
DEBUG = False
ALLOWED_HOSTS = ['xxx.xxx.xxx.xxx', 'ocpizza.fr', 'www.ocpizza.fr']

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'oc_restaurant',
        'USER': 'oc_user',
        'PASSWORD': 'MDP',
        'HOST': 'xxx.xxx.xxx.xxx',
        'PORT': 5432,
    }
}
```

```
}
```

4.1.1.9 - Lancement de l'application

Pour lancer l'application avec son environnement de production, rentrer la commande suivante dans le terminal :

```
gunicorn oc_pizza.wsgi:application
```

4.1.2 - Configuration de Supervisor et de Gunicorn

Supervisor permet de contrôler l'ensemble des processus du système d'exploitation. Il permet de lancer et redémarrer les services lorsque ces derniers rencontrent un échec. Il permettra ici de lancer l'application via gunicorn.

Créer le fichier de configuration de Supervisor :

```
nano /etc/supervisor/conf.d/oc_pizza-gunicorn.conf
```

Le remplir avec ces informations :

```
[program:oc_pizza-gunicorn]
command = /home/user/venv_oc_pizza/bin/gunicorn oc_pizza.wsgi:application
autostart = true
autorestart = true
environment = DJANGO_SETTINGS_MODULE='oc_pizza.settings.production'
directory = /home/user/oc_pizza
```

Note : le fichier de configuration oc_pizza-gunicorn.conf est aussi remis au client avec le dossier d'exploitation.

4.1.3 - Configuration de Nginx

Rentrer la commande suivante dans le terminal pour créer le fichier suivant :

```
nano /etc/nginx/sites-available/oc_pizza
```

Insérer le code suivant dans le fichier :

```
server {
    listen 80;
    server_name xxx.xxx.xxx.xxx ocpizza.fr www.ocpizza.fr;
    location = /favicon.ico { access_log off; log_not_found off; }
    location /static/ {
        root /home/user/oc_pizza/oc_pizza;
    }
}
```



```
location / {
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_redirect off;
    proxy pass http://127.0.0.1:8000;
}
}
```

Note : le fichier de configuration oc_pizza est aussi remis au client avec le dossier d'exploitation.

Créer le lien symbolique entre le fichier et le dossier *sites-enabled* de Nginx :

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

Relancer Nginx :

```
sudo systemctl restart nginx
```

4.1.4 - Configuration de Redis

Le fichier de configuration de Redis, *redis.conf*, se trouve dans *etc/redis/*. Il est créé par défaut lors de l'installation de Redis. Par défaut, Redis se lancera à chaque démarrage du serveur.

Pour que Redis puisse être supervisé par le système, mettre à jour l'option de *supervised* de « no » à « systemd » dans le fichier *redis.conf* :

```
# If you run Redis from upstart or systemd, Redis can interact with your
# supervision tree. Options:
# supervised no      - no supervision interaction
# supervised upstart - signal upstart by putting Redis into SIGSTOP mode
# supervised systemd - signal systemd by writing READY=1 to $NOTIFY_SOCKET
# supervised auto    - detect upstart or systemd method based on
#                      UPSTART_JOB or NOTIFY_SOCKET environment variables
# Note: these supervision methods only signal "process is ready."
#       They do not enable continuous liveness pings back to your supervisor.
supervised systemd
```

Relancer Redis pour prendre en compte la modification :

```
sudo systemctl restart redis
```

4.1.5 - Lancement de l'application Angular

Dans *home/oc_pizza/angular/dist*, entrer la commande suivante pour préparer la mise en production de l'application Angular :

```
ng build -prod
```

Se connecter au compte OC Pizza de [Firebase](#) :

```
firebase login
```

Initialiser le projet :

```
firebase init
```

Mettre en production :

```
firebase deploy
```

Lancement :

```
firebase open
```

4.2 - Déploiement de la base de données

4.2.1 - Configuration de l'espace serveur de la base de données

4.2.1.1 - Serveur dédié à la base de données

La base de donnée est déployée sur un serveur privé virtuel loué chez Digital Ocean.

4.2.1.2 - Accès au serveur de base de données

Le serveur de base de données sera accessible avec l'API ou l'interface utilisateur mise à disposition par [Digital Ocean](#).

4.2.1.3 - Configuration de la base de données

Note : Digital Ocean fournit un [guide](#) pour configurer sa base de données.

L'interface graphique permet de créer et configurer directement la base de données ainsi que ses utilisateurs.

4.2.1.4 - Insertion des données dans la base de données

Une fois la base de données créée, il faut un insérer le fichier SQL contenant les tables ainsi que leurs données de base nécessaires au bon fonctionnement de l'application.

5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

5.1 - Base de données

La base de donnée est accessible dès qu'elle est créée et configurée sur le serveur Digital Ocean.

5.2 - Application web

5.2.1 - Django

L'application se lance avec la commande suivante dans le terminal, à la racine du projet :

```
gunicorn oc_pizza.wsgi:application
```

Pour l'arrêter, il suffit de presser les touches CTRL+C ou CTRL + BREAK/PAUSE sur le clavier.

Il est à noter que l'application se lancera automatiquement grâce à l'intégration de Supervisor (voir partie [4.1.2. Configuration de Supervisor et de Gunicorn](#)).

5.2.2 - Angular

L'application déployée sur Firebase se lance avec la commande suivante :

```
firebase open
```

Pour désactiver une application, se connecter au compte [Firebase](#) et aller dans les paramètres de l'application.

6 - PROCÉDURE DE MISE À JOUR

6.1 - Base de données

Digital Ocean permet de planifier les dates et heures des mises à jours de la base de données via son interface utilisateur.

6.2 - Application web

Afin d'éviter toute incompatibilité, les versions des librairies Django spécifiées dans le fichier *requirements.txt* doivent être mises à jour manuellement.

7 - SUPERVISION/MONITORING

7.1 - Supervision de l'application web

Digital Ocean met à disposition des outils de [monitoring](#) dans l'interface utilisateur, sans coûts additionnels. Ils sont intégrés par défaut :

- Un tableau de bord donnant des métriques sur le CPU, la mémoire et l'utilisation du disque ainsi que l'espace disponible
- Création d'alertes : il est possible de définir ses propres alertes sur les métriques de son choix et de recevoir une notification quand une métrique dépasse la limite définie

Il est possible de personnaliser les outils de monitoring par l'API.

7.2 - Supervision de la base de données

Digital Ocean met également à disposition des [outils](#) de métriques et d'alertes pour superviser le bon fonctionnement de la base de données. Il est là aussi possible de définir ses propres alertes sur les métriques de son choix et d'être notifié en cas de dépassement des dites métriques.

D'autres fonctionnalités sont accessibles avec une base de données PostgreSQL:

- connexions à la base de données
- taux de réussite de la mémoire cache
- rythme de création, modification, suppression et extraction de données

8 - PROCÉDURE DE SAUVEGARDE ET RESTAURATION

8.1 - Sauvegarde et restauration de l'application web

Digital Ocean génère automatiquement des [sauvegardes](#) toutes les semaines pour un coût supplémentaire de 20%. Ils sont conservés pendant quatre semaines.

Il est possible de consulter la date de procédure de *backup*, déterminée par Digital Ocean, dans le panneau de configuration mis en place dans l'interface utilisateur.

8.2 - Sauvegarde et restauration de la base de données

Le serveur Digital Ocean procède automatiquement à des sauvegardes quotidiennes de la base de données, conservées pendant sept jours.

Pour restaurer une sauvegarde de la base de données, il faut se rendre dans la section *Overview* de sa base de données et cliquer sur le bouton *Action* à droite de la fenêtre pour activer le menu déroulant. Cliquer sur l'option *Restore from backup*.

The screenshot displays the DigitalOcean interface for a database cluster named 'example-database'. The top navigation bar includes 'Overview', 'Insights', 'Logs & Queries', 'Users & Databases', 'Connection Pools', and 'Settings'. The 'Overview' tab is active. The main content area shows the 'DATABASE CLUSTER TOTAL COST' as '\$15 monthly rate' and 'CONNECTION DETAILS' with 'Public network' selected. A dropdown menu labeled 'Actions' is open on the right, showing options: 'Connection details', 'Add a read-only node', 'Fork database cluster', 'Restore from backup' (highlighted in blue), 'Edit tags', 'Move to...', and 'Destroy'.

9 - GLOSSAIRE

repository	dépôt
backup	sauvegarde
VPS	serveur privé virtuel (Virtual Private Server)