



OC Pizza

Système de gestion de Pizzerias

Dossier d'exploitation

Version 1

Auteur

Sylvain Rieutor
Développeur

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
3.1.1 -Serveur de Base de données.....	5
3.1.1.1 -Caractéristiques techniques.....	5
3.1.2 -Serveur Web.....	5
3.1.2.1 -Caractéristiques techniques.....	5
3.2 -Bases de données.....	6
3.3 -Web-services.....	6
3.4 -Ressources.....	6
4 -Procédure de déploiement.....	7
4.1 -Déploiement de l'Application Web.....	7
4.1.1 -Composition de l'application.....	7
4.1.2 -Environnement de l'application web.....	7
4.1.2.1 -Variables d'environnement.....	7
4.1.3 -Répertoire de configuration applicatif.....	8
4.1.4 -Déploiement.....	8
4.1.5 -Vérifications.....	13
5 -Procédure de démarrage / arrêt.....	14
5.1 -Base de données.....	14
5.1.1 -Démarrage:.....	14
5.1.2 -Arrêt:.....	14
5.2 -Application web.....	14
5.2.1 -Démarrage:.....	14
5.2.2 -Arrêt:.....	14
6 -Procédure de mise à jour.....	15
6.1 -Base de données.....	15
6.2 -Application web.....	15
6.3 -Redémarrage du serveur.....	15
7 -Supervision/Monitoring.....	16
7.1 -Supervision de l'application web.....	16
8 -Annexes.....	17

1 - VERSIONS

Auteur	Date	Description	Version
SRR	01/05/20	Création du document	1

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier d'exploitation du système de gestion du groupe de pizzeria.

Ce document a pour principal objectif de documenter les informations essentielles pour une bonne utilisation et la maintenance de l'application.

2.2 - Références

Pour de plus amples informations, se référer aux documents suivants :

1. **OC-AF-1:** Dossier d'Analyse fonctionnelle
2. **OC-ST-1:** Dossier de Spécifications Techniques

3 - PRÉ-REQUIS

3.1 - Système

3.1.1 - Serveur de Base de données

Le SGBD utilisé sera PostgreSQL (PSQL), il sera installé lors de la préparation du serveur.
Un utilisateur avec les droits nécessaires sera créé.

3.1.1.1 - Caractéristiques techniques

La base de données sera hébergée dans le même VPS que l'application.
Voir caractéristiques techniques du serveur.
Un utilisateur sera créé et ajouté au groupe SUDO

3.1.2 - Serveur Web

Le serveur web sera préparé avec les applications **Django**, **Nginx** et **Gunicorn** dans leur dernière version.

3.1.2.1 - Caractéristiques techniques

L'application sera déployée sur un serveur VPS (Serveur Privé Virtuel)
Nous recommandons l'offre **Value VPS SSD 1** de l'hébergeur **OVH**.

Processeur : **1vCore**

Mémoire : **2Go**

Stockage : **40 Go NV/Me**

Bande passante : **250 Mbit/s - trafic illimité**

Voir les offres de VPS sur OVH :

<https://www.ovhcloud.com/fr/vps/>

3.2 - Bases de données

L'ORM de Django sera utilisé

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

- **PSQL:** version 10.12

3.3 - Web-services

Un abonnement à un service de paiement en ligne **Paypal**

Voir les informations sur ce service:

<https://www.paypal.com/fr/webapps/mpp/merchant>

3.4 - Ressources

- **Django:** version 3.0.3
- **Nginx:** version 1.14
- **Gunicorn:** version 20.0.4
- **Python:** version 3.8.2
- **Distribution:** LINUX seule
- **Système d'exploitation:** Ubuntu server 18.04

4 - PROCÉDURE DE DÉPLOIEMENT

4.1 - Déploiement de l'Application Web

4.1.1 - Composition de l'application

- **ocpizza** : Dossier contenant les fichiers de configuration (Django)
 - **settings**: Dossier contenant les paramètres
 - **__init__.py**: réglages communs
 - **production.py**: réglages en production
 - **travis.py**: réglages Travis
 - **static** : Dossier contenant les 'static' : images, .CSS, .JS
- **product** : Dossier contenant les fichiers du package 'product'.
- **restaurant** : Dossier contenant les fichiers du package 'restaurant'.
- **user**: Dossier contenant les fichiers du package 'user'.
- **requirements.txt** : ensemble des librairies nécessaires.
- **Docs** : Dossier contenant la documentation

4.1.2 - Environnement de l'application web

4.1.2.1 - Variables d'environnement

Les variables d'environnement nécessaires seront dans les fichiers suivant:

- **secrets** : Variables d'environnement (Nginx)
- **secretsenv**: Commandes d'exportation des variables
voir le contenu de ces fichiers en Annexe

Ces fichiers seront dans le dossier **www**
se référer à la procédure de déploiement.

4.1.3 - Répertoire de configuration applicatif

Les différents fichiers de configuration:

- **Gunicorn:**
 - ocpizza-gunicorn.service
 - ocpizza-gunicorn.socket
- **Nginx:** ocpizza

4.1.4 - Déploiement

- **Créer et connecter un utilisateur.**

- **Connection de l'utilisateur root:**

```
ssh root@{{AdresselP}}
```

Adresse ip et mot de passe fournis dans les mail de confirmation de location

- **Création d'un utilisateur:**

```
$ adduser {{NomUtilisateur}}
```

le mot de passe utilisateur sera renseigné ici

- **Ajout de l'utilisateur dans le groupe sudo:**

```
$ usermod -aG sudo {{NomUtilisateur}}
```

- **Connexion de l'utilisateur:**

```
$ ssh {{NomUtilisateur}}@{{AdresselP}}
```

- le mot de passe sera demandé

- **Le Pare-feu:**

- **Configuration:**

- Vérification des apps enregistrées:

```
$ sudo ufw app list
```

- Vérification du statut:

```
$ sudo ufw status
```

- Authorisation d'OpenSSH

```
$ sudo ufw allow openSSH
```

- **Activation:**

```
$ sudo ufw enable
```

- Vérification:

```
$ sudo ufw status
```

- **Mise a jour de la distributin LINUX:**

- Récupérer les mises à jour:

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

- installer les mises à jour:

```
$ sudo apt-get upgrade
```

- **Dépendances du projet:**

- Installation des dépendances:

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

- **Configuration de PSQL:**

- Connexion de l'utilisateur 'root':

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

- Création de la base de donnée:

```
$ CREATE DATABASE {{NomDeLaBase}};
```

- Création de la base de donnée:

```
$ CREATE USER {{NomUtilisateur}} WITH PASSWORD {{MotDePasse}};
```

- Mise à jour des paramètres:

- Paramètres Django:

```
$ ALTER ROLE {{NomUtilisateur}} SET client_encoding TO 'utf8';  
$ ALTER ROLE {{NomUtilisateur}} default_transaction_isolation TO 'read  
committed';  
$ ALTER ROLE {{NomUtilisateur}} SET timezone TO 'UTC';
```

- Privilèges Administrateur:

```
$ GRANT ALL PRIVILEGES ON DATABASE {{NomUtilisateur}} TO  
{{NomDeLaBase}};
```

- **L'environnement:**

- Installation et mise à jour de **pip**:

```
$ sudo -H pip3 install --upgrade pip
```

- installation de virtualenv

```
$ sudo -H pip3 install virtualenv
```

- **Création d'un dossier www:**

```
$ mkdir ~/www
```

- Vérification:

```
$ ls
```

- Modification des permissions

```
$ chmod o-rw www
```

- Vérification des permissions:

```
$ ls -l
```

- **Cloner le repo GitHub**

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

- **L'ennvironnement virtuel:**

- Création:

```
$ virtualenv ocpizzaenv
```

- Activation:

```
$ source ocpizzaenv/bin/activate
```

- installation des librairies:

```
$ pip install django gunicorn psycpg2
```

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

- Récupération des variables:

```
$ source ./secretsenv
```

- **Migrations:**

```
$ cd ocpizza
```

```
$ python manage.py makemigrations
```

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

- **Fichiers statiques:**

```
$ python manage.py collecstatic
```

- **Fichiers de configuration:**

- **Gunicorn:**

```
$ cd /etc/systemd/system
```

- Création du fichier ocpizza-gunicorn.service:

```
$ sudo vi ocpizza-gunicorn.service
```

voir le contenu en annexe

- enregistrer:

```
echap puis :wq
```

- Création du fichier ocpizza-gunicorn.socket:

```
$ sudo vi ocpizza-gunicorn.socket
```

voir le contenu en annexe

- **Nginx:**

```
$ cd etc/nginx/sites-available
```

- Création du fichier ocpizza:

```
$ sudo vi ocpizza
```

voir le contenu en annexe

- **Activation:**

```
$ sudo systemctl start ocpizza-gunicorn
```

```
$ sudo systemctl enable ocpizza-gunicorn
```

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

```
$ sudo systemctl restart nginx
```

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

4.1.5 - Vérifications

Afin de vérifier le bon déploiement de l'application:

- Vérification **Nginx**:

```
$ systemctl status nginx
```

doit retourner **active (running)**

- Vérification **Psq**l:

- vérifier que le dossier de configuration existe:

```
$ stat /etc/postgresql
```

- vérifier le statut:

```
systemctl status postgresql
```

doit retourner **active (exited)**

5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

5.1 - Base de données

La base de données Psql démarre automatiquement avec l'application.

5.1.1 - Démarrage:

```
$ sudo service postgresql start
```

- vérifier le statut:

```
$ systemctl status postgresql
```

doit retourner **active (exited)**

5.1.2 - Arrêt:

```
$ sudo service postgresql stop
```

5.2 - Application web

5.2.1 - Démarrage:

Nécessite un démarrage des services Gunicorn et Nginx:

```
$ sudo systemctl start purbeurre-gunicorn
```

```
$ sudo systemctl start nginx
```

5.2.2 - Arrêt:

Nécessite un arrêt des services Gunicorn et Nginx:

```
$ sudo systemctl stop purbeurre-gunicorn
```

```
$ sudo systemctl stop nginx
```

6 - PROCÉDURE DE MISE À JOUR

Toutes modifications apportées à Django nécessite un redémarrage

```
$ sudo systemctl restart purbeurre-gunicorn
```

Toutes modification apportées à Nginx nécessite un redémarrage

```
$ sudo systemctl restart nginx
```

6.1 - Base de données

- **Migrations:**

```
$ cd ocpizza  
$ python manage.py makemigrations  
$ python manage.py migrate
```

```
$ sudo systemctl restart purbeurre-gunicorn
```

6.2 - Application web

- **Fichiers statiques:**

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

```
$ sudo systemctl restart purbeurre-gunicorn
```

6.3 - Redémarrage du serveur

```
$ sudo reboot now
```

Ou directement depuis le tableau de bord OVH

7 - SUPERVISION/MONITORING

7.1 - Supervision de l'application web

OVH fournit un monitoring permettant de contrôler l'utilisation:

- Du **processeur**
- De la **RAM**
- Du **trafic**

Ces informations seront disponibles sur le site OVH

<https://www.ovh.com/manager/dedicated/#/iaas/vps/NomDuVps.ovh.net/monitoring>

Possibilité de suivre des *logs* avec une application de traçage

exemple avec l'application **sentry** les logs seront consultables ici:

<https://sentry.io/organizations/ocpizza/issues/?project={{NuméroProjet}}>

Les log peuvent permettre de tracer des événements comme :

- Une nouvelle inscription
- Un panier abandonné

8 - ANNEXES

Gunicorn:

Fichier purbeurre-gunicorn.service:

```
[Service]  
EnvironmentFile=/home/ocpizza/www/secrets  
User=ocpizza  
Group=www-data  
WorkingDirectory=/home/ocpizza/www/ocpizza  
ExecStart=/home/ocpizza/www/purbeurreenv/bin/gunicorn \  
--access-logfile - \  
--workers 3 \  
--bind unix:/run/ocpizza-gunicorn.sock \  
ocpizza.wsgi:application  
  
[Install]  
WantedBy=multi-user.target
```

Fichier purbeurre-gunicorn.socket:

```
[Unit]  
Description=gunicorn socket  
  
[Socket]  
ListenStream=/run/ocpizza-gunicorn.socket  
  
[Install]  
WantedBy=sockets.target
```

Nginx:

Fichier purbeurre:

```
server {  
    server_name NomDuServeur;  
  
    location /staticfiles/ {  
        root /home/ocpizza/www/ocpizza/ocpizza;  
    }  
  
    location / {  
        include proxy_params;  
        proxy_pass http://unix:/run/ocpizza-gunicorn.sock;  
    }
```

Environnement:

Fichier secrets:

```
SECRET_KEY='CléSecrèteDjango'  
DB_NAME='NomDeLaBase'  
DB_USER='NomUtilisateur'  
DB_PASSWORD='MotDePasseUtilisateur'  
DJANGO_SETTINGS_MODULE='ocpizza.settings.production'
```

Fichier secretsenv:

```
export SECRET_KEY='CléSecrèteDjango'  
export DB_NAME='NomDeLaBase'  
export DB_USER='NomUtilisateur'  
export DB_PASSWORD='MotDePasseUtilisateur'  
export DJANGO_SETTINGS_MODULE='ocpizza.settings.production'
```