



OC Pizza

ERP OC Pizza

Dossier d'exploitation

Version 1.0

**Auteur**  
Etienne BARBIER  
Développeur

# TABLE DES MATIERES

<b>1 - Versions.....</b>	<b>3</b>
<b>2 - Introduction.....</b>	<b>4</b>
2.1 - Objet du document .....	4
2.2 - Références.....	4
<b>3 - Pré-requis .....</b>	<b>5</b>
3.1 - Système .....	5
3.1.1 - Serveur de base de données .....	5
3.1.2 - Serveur d'application .....	5
3.1.3 - Serveur web .....	5
3.2 - Base de données .....	5
3.3 - Web services .....	5
<b>4 - Procédure de déploiement .....</b>	<b>6</b>
4.1 - Déploiement de la base de données.....	6
4.1.1 - Mise à jour des programmes .....	6
4.1.2 - Installation de PostgreSQL sur le serveur de base de données .....	6
4.2 - Déploiement de l'application web .....	7
4.2.1 - Mise à jour des programmes .....	7
4.2.2 - Variables d'environnement de l'application web .....	7
4.2.3 - Fichier de réglages de l'application Django .....	7
4.2.4 - Installation de Python sur le serveur d'application .....	8
4.2.5 - Installation de l'application Django sur le serveur d'application.....	8
4.2.6 - Chargement de la base de données (optionnel).....	8
4.2.7 - Configuration du serveur web NGINX.....	9
4.2.8 - Configuration de Supervisor pour piloter le serveur d'application Gunicorn .....	10
4.2.9 - Vérification .....	10
<b>5 - Procédure de démarrage / arrêt .....</b>	<b>11</b>
5.1 - Base de données .....	11
5.2 - Application web.....	11
<b>6 - Procédure de mise à jour .....</b>	<b>12</b>
6.1 - Base de données .....	12
6.2 - Application web.....	12
<b>7 - Supervision/Monitoring .....</b>	<b>13</b>
7.1 - Supervision de l'application web .....	13
7.2 - Supervision des serveurs.....	13
<b>8 - Procédure de sauvegarde et restauration.....</b>	<b>14</b>
<b>9 - Glossaire .....</b>	<b>15</b>

# 1 - VERSIONS

Auteur	Date	Description	Version
Etienne BARBIER	04/05/2020	Création du document	1.0

## 2 - INTRODUCTION

### 2.1 - Objet du document

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

L'objectif de ce document est de définir les procédures permettant de déployer et maintenir l'application « ERP OC Pizza ».

Les éléments du présent dossier découlent d'échanges de courriels avec OC Pizza, du dossier de conception fonctionnelle et du dossier de conception technique de l'application « ERP OC Pizza ».

### 2.2 - Références

Pour de plus amples informations, se référer :

1. **PDOCPizza\_01\_dossier\_conception\_fonctionnelle** : dossier de conception fonctionnelle de l'application
2. **PDOCPizza\_02\_dossier\_conception\_technique** : dossier de conception technique de l'application

## 3 - PRE-REQUIS

### 3.1 - Système

#### 3.1.1 - Serveur de base de données

Comme décrit dans la section 5.1 du dossier de conception technique de l'application, le serveur de base de données est un serveur Linux Ubuntu (version 18.04 ou LTS ultérieure) sur lequel est installé PostgreSQL (version 12.2 ou ultérieure).

L'adresse IP du serveur de base de données sera notée dans la suite du document comme suit : *IP\_serveur\_base\_de\_données*.

Le schéma de la base de données est indiqué dans le Modèle Physique de Données annexé au dossier de conception technique de l'application. La base de données

#### 3.1.2 - Serveur d'application

Comme décrit dans la section 5.2 du dossier de conception technique de l'application, le serveur d'application est un serveur Linux Ubuntu (version 18.04 ou LTS ultérieure) sur lequel sont installés :

- Python (version 3.6.9 ou ultérieure), le langage de programmation back end du projet
- Gunicorn (version 20.0.4 ou ultérieure), qui traite les requêtes HTTP et les transmet au code Python/Django
- Django (version 3.0.5 ou ultérieure), le framework web Python utilisé pour ce projet
- les autres dépendances de Python nécessaires au projet

Gunicorn pilote l'exécution de l'application web.

L'adresse IP du serveur d'application sera notée dans la suite du document comme suit : *IP\_serveur\_application*.

#### 3.1.3 - Serveur web

Comme décrit dans la section 5.3 du dossier de conception technique de l'application, le serveur web est physiquement le serveur d'application, sur lequel est également installé NGINX (version 1.17.10 ou ultérieure).

### 3.2 - Base de données

La base de données utilisée doit être accessible et à jour. Elle est paramétrée depuis l'application Django.

### 3.3 - Web services

L'ERP OC Pizza utilise des web services : API de paiement, API de GPS, API de téléphonie (service de SMS).

Le dernier service tiers présenté dans le diagramme de déploiement (section 5 du dossier de conception technique) est le service de messagerie : celui-ci est inclus dans l'application Django, il ne nécessite donc pas de web service.

Les web services suivants doivent être accessibles pour garantir le bon fonctionnement de l'application :

- API de paiement : [PayPal](#)
- API de GPS : [Directions API \(Google Maps\)](#)
- API de téléphonie (service de SMS) : [Digitaleo](#)

## 4 - PROCEDURE DE DEPLOIEMENT

### 4.1 - Déploiement de la base de données

#### 4.1.1 - Mise à jour des programmes

Mettre à jour les programmes actuels du système en exécutant la commande suivante sur le serveur de base de données :

```
sudo apt-get update
```

#### 4.1.2 - Installation de PostgreSQL sur le serveur de base de données

Sur le serveur de base de données, installer PostgreSQL en exécutant la commande suivante :

```
sudo apt-get install libpq-dev postgresql postgresql-contrib
```

Accéder à la console de PostgreSQL en exécutant la commande suivante :

```
sudo -u postgres psql
```

Dans cette console, créer la base de données, l'utilisateur et lui attribuer les privilèges nécessaires :

```
CREATE DATABASE erp_oc_pizza_database;  
CREATE USER db_user WITH PASSWORD 'db_password';  
ALTER ROLE db_user SET client_encoding TO 'utf8';  
ALTER ROLE db_user SET default_transaction_isolation TO 'read committed';  
ALTER ROLE db_user SET timezone TO 'Europe/Paris';  
GRANT ALL PRIVILEGES ON DATABASE erp_oc_pizza_database TO db_user;  
\q
```

**Attention** à bien remplacer **db\_user** et **db\_password** par les bonnes valeurs (cf. sections 4.2.2 et 4.2.7) !

## 4.2 - Déploiement de l'application web

### 4.2.1 - Mise à jour des programmes

Mettre à jour les programmes actuels du système en exécutant cette commande sur le serveur d'application :

```
sudo apt-get update
```

### 4.2.2 - Variables d'environnement de l'application web

Voici les variables d'environnement nécessaires à l'application ERP OC Pizza :

Nom	Description
SECRET_KEY	Clé secrète nécessaire au fonctionnement de l'application Django
DATABASE_NAME	Nom de la base de données PostgreSQL Ici : erp_oc_pizza_database
DATABASE_USER	Nom d'utilisateur de la base de données Exemple <b>à ne pas utiliser en production</b> : db_user
DATABASE_PASSWORD	Mot de passe de l'utilisateur DATABASE_USER Exemple <b>à ne pas utiliser en production</b> : db_password
DATABASE_HOST	Hôte de la base de données Ici : <i>IP_serveur_base_de_données</i> (remplacer par l'adresse IP réelle)
DATABASE_PORT	Port de connexion à la base de données Ici : 5432
DJANGO_SETTINGS_MODULE	Indication du fichier de réglages Django pour la production Ici : erp_oc_pizza.settings.production

Ces variables sont définies dans le fichier de configuration de Supervisor (cf. section 4.2.7).

### 4.2.3 - Fichier de réglages de l'application Django

La variable d'environnement DJANGO\_SETTINGS\_MODULE définit le fichier de réglages de l'application Django à prendre en compte. Les autres variables d'environnement définies dans la section précédente sont utilisées dans le fichier désigné par DJANGO\_SETTINGS\_MODULE dont voici un extrait :

```
import os

SECRET_KEY = os.environ.get('SECRET_KEY')
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': os.environ.get('DATABASE_NAME'),
        'USER': os.environ.get('DATABASE_USER'),
        'PASSWORD': os.environ.get('DATABASE_PASSWORD'),
        'HOST': os.environ.get('DATABASE_HOST'),
        'PORT': os.environ.get('DATABASE_PORT'),
    }
}
```

#### 4.2.4 - Installation de Python sur le serveur d'application

Sur le serveur d'application, installer Python en exécutant la commande suivante :

```
sudo apt-get install python3-pip python3-dev
```

#### 4.2.5 - Installation de l'application Django sur le serveur d'application

Se placer au niveau du répertoire `/home` de l'utilisateur linux :

```
cd ~
```

Importer le répertoire du projet depuis le dépôt Git :

```
git clone https://github.com/etienne86/erp_oc_pizza.git
```

Installer l'utilitaire d'environnements virtuels **venv** :

```
sudo apt-get install python3-venv
```

Activer l'environnement virtuel en exécutant les commandes suivantes :

```
cd erp_oc_pizza
python3 -m venv venv
. venv/bin/activate
```

Installer les dépendances de l'application :

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

Générer les fichiers statiques :

```
./manage.py collectstatic
```

Lancer les migrations :

```
./manage.py migrate
```

Créer un super utilisateur pour l'interface d'administration de l'application Django :

```
./manage.py createsuperuser
```

#### 4.2.6 - Chargement de la base de données (optionnel)

Si nécessaire, charger un dump de la base de données **ou** exécuter une commande personnalisée d'initialisation de la base de données :

```
./manage.py loaddata chemin/vers/le/dump/nom_du_dump.json
```

**ou**

```
./manage.py commande_personnalisee
```

**Attention** à bien remplacer `chemin/vers/le/dump/nom_du_dump.json` ou `commande_personnalisee` par la bonne valeur !



#### 4.2.7 - Configuration du serveur web NGINX

Installer NGINX sur le serveur d'application :

```
sudo apt-get install nginx
```

Ajouter le fichier de configuration :

```
cd /etc/nginx/  
sudo touch sites-available/erp_oc_pizza
```

Créer un lien symbolique vers le fichier *sites-enabled* :

```
sudo ln -s /etc/nginx/sites-available/erp_oc_pizza/etc/nginx/sites-enabled
```

Éditer le fichier :

```
sudo vi sites-available/erp_oc_pizza
```

Voici le contenu à insérer dans le fichier :

```
server {  
  
    listen 80; server_name IP_serveur_application;  
    root /home/linux_user/erp_oc_pizza/  
  
    location /static {  
        alias /home/linux_user/erp_oc_pizza/staticfiles/  
    }  
  
    location / {  
        proxy_set_header Host $http_host;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_redirect off;  
        if (!-f $request_filename) {  
            proxy_pass http://127.0.0.1:8000;  
            break;  
        }  
    }  
}
```

**Attention** à bien remplacer **IP\_serveur\_application** et **linux\_user** par les bonnes valeurs !

Recharger le service pour prendre en compte les modifications

```
sudo service nginx reload
```

#### 4.2.8 - Configuration de Supervisor pour piloter le serveur d'application Gunicorn

Installer Supervisor :

```
sudo apt-get install supervisor
```

Configurer Supervisor :

```
sudo vi /etc/supervisor/conf.d/erp_oc_pizza-gunicorn.conf
```

Voici le contenu à insérer dans le fichier :

```
[program:erp_oc_pizza-gunicorn]
command=/home/linux_user/erp_oc_pizza/venv/bin/gunicorn erp_oc_pizza.wsgi:application
user = linux_user
directory = /home/linux_user/erp_oc_pizza
autostart = true
autorestart = true
environment = SECRET_KEY="votre_clé_secrète", DATABASE_NAME="erp_oc_pizza_database",
DATABASE_USER="db_user", DATABASE_PASSWORD="db_password", DATABASE_HOST="
IP_serveur_base_de_données", DATABASE_PORT="5432",
DJANGO_SETTINGS_MODULE="erp_oc_pizza.settings.production"
```

**Attention** à bien remplacer ces expressions par les bonnes valeurs !

Exécuter les commandes suivantes :

```
sudo supervisorctl reread
sudo supervisorctl update
```

Le serveur tourne désormais en production avec Gunicorn.

#### 4.2.9 - Vérification

Afin de vérifier le bon déploiement de l'application, exécuter la commande suivante :

```
sudo supervisorctl status
```

## 5 - PROCEDURE DE DEMARRAGE / ARRET

### 5.1 - Base de données

Le serveur de base de données ne doit jamais être arrêté.

### 5.2 - Application web

L'application web doit être en permanence disponible. Cependant, si un problème survient, Supervisor est paramétré pour relancer le serveur d'application Unicorn (cf. section 4.2.7) avec la ligne suivante, dans le fichier de configuration de Supervisor :

```
autorestart = true
```

## 6 - PROCEDURE DE MISE A JOUR

### 6.1 - Base de données

La mise à jour de la base de données est réalisée dans le cadre de l'utilisation de l'application : par exemple, le niveau de stock des ingrédients des pizzas peut être mis à jour lors des inventaires, à l'aide de la fonctionnalité prévue (cf. section 5.2.2.5 du dossier de conception fonctionnelle).

### 6.2 - Application web

La mise à jour de l'application web est réalisée lors de la mise à disposition d'une mise à jour sur le dépôt Git, par exemple dans le cas de l'ajout d'une nouvelle fonctionnalité. Dans ce cas, exécuter les commandes suivantes pour mettre à jour l'application web :

```
cd ~/erp_oc_pizza  
git pull origin master
```

Cette étape nécessite impérativement le fait d'avoir initialement exécuté les commandes de la section 4.2.5 lors de l'installation de l'application, et notamment la commande suivante :

```
git clone https://github.com/etienne86/erp_oc_pizza.git
```

## 7 - SUPERVISION/MONITORING

### 7.1 - Supervision de l'application web

L'application web est paramétrée pour être supervisée avec le logiciel Sentry vers lequel remontent les erreurs, avertissements et logs d'information, que Sentry nomme les *issues*. Les *issues* remontées dans Sentry durant les sept derniers jours sont consultables en suivant ce lien :

<https://sentry.io/organizations/erp-oc-pizza/issues/?project=0000000&statsPeriod=7d>

Il est possible de personnaliser le reporting et de configurer des alertes mail.

### 7.2 - Supervision des serveurs

Afin de suivre le bon fonctionnement du serveur de base de données et du serveur d'applications, il y a plusieurs possibilités en fonction de l'hébergeur utilisé :

- Utiliser un agent fourni par l'hébergeur le cas échéant (c'est par exemple le cas de DigitalOcean et Amazon Web Services) pour surveiller le niveau d'utilisation de CPU et de mémoire vive
- Utiliser un service externe de monitoring (exemple : New Relic)

## 8 - PROCEDURE DE SAUVEGARDE ET RESTAURATION

Afin de sauvegarder ou restaurer des données du serveur d'application et du serveur de base de données, il est nécessaire de suivre les procédures mises à disposition par l'hébergeur de chaque serveur.

## 9 - GLOSSAIRE

<b>CPU</b>	Littéralement <i>Central Process Unit</i> , ou « unité centrale de traitement », ou plus communément « processeur ». C'est le composant de la machine (ici, du serveur) qui exécute les instructions élémentaires (les opérations de plus bas niveau) issues des programmes informatiques.
<b>Hébergeur</b>	C'est une société de prestations de services qui met à disposition tout ou partie de serveurs, ainsi que des services tels que la supervision du serveur, la sauvegarde de données, etc.