



# OC Pizzas

## oc-pizzas.com

Dossier d'exploitation

Version 1.0.5

**Auteur**

Gino Ladowitch

*Directeur technique*

**Gino Ladowitch**  
<siteWebEntreprise>

<AdresseEntreprise> – <Téléphone> – <Email>

S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Xxxx – SIREN 999 999 999 – Code  
APE : 6202A

# TABLE DES MATIÈRES

<b>1 - Introduction.....</b>	<b>3</b>
1.1 - Objet du document.....	3
1.2 - Références.....	3
<b>2 - Infrastructure.....</b>	<b>4</b>
2.1 - Serveurs.....	4
2.1.1 - Serveur d'application.....	4
2.1.2 - Serveur Base de données.....	4
2.1.3 - Serveur Web & Fichiers.....	4
<b>3 - Procédure de déploiement.....</b>	<b>5</b>
3.1 - Schéma général.....	5
3.2 - Création de la base de données.....	6
3.3 - Copier les fichiers de l'application.....	6
3.4 - Installer les modules python nécessaires.....	6
3.5 - Gestion des fichiers statiques.....	6
3.6 - Fichier de configuration de production.....	6
3.7 - Installation et configuration de NGINX.....	7
3.7.1 - Installation de NGINX.....	7
3.7.2 - Configuration de NGINX.....	7
3.8 - Installation de Gunicorn.....	7
3.9 - Installation et configuration de Supervisor.....	8
3.9.1 - Installation de Supervisor.....	8
3.9.2 - Configuration de Supervisor.....	8
3.9.3 - Lancer le processus.....	8
3.9.4 - DataSources.....	8
<b>4 - Procédure de démarrage / arrêt.....</b>	<b>9</b>
4.1 - Installation et configuration de Supervisor.....	9
4.1.1 - Installation de Supervisor.....	9
4.1.2 - Configuration de Supervisor.....	9
4.2 - Lancer le processus.....	9
<b>5 - Procédure de mise à jour.....</b>	<b>10</b>
5.1 - Base de données.....	10
<b>6 - Supervision/Monitoring.....</b>	<b>11</b>
6.1 - Supervision de l'application web.....	11
<b>7 - Procédure de sauvegarde et restauration.....</b>	<b>12</b>

# 1 - INTRODUCTION

## 1.1 - Objet du document

Le présent document constitue le dossier d'exploitation de l'application Python/Django oc-pizzas.

## 1.2 - Références

Pour de plus amples informations, se référer :

1. **OC Pizzas - Dossier de conception technique** : Dossier de conception technique de l'application
2. **OC Pizzas - Dossier de conception fonctionnelle** : Dossier de conception fonctionnelle de l'application

## 2 - INFRASTRUCTURE

### 2.1 - Serveurs

Les serveurs d'application,

#### 2.1.1 - *Serveur d'application*

Serveur physique ou virtuel hébergeant l'application web.

Nous préconisons un server Scaleway ARM64-128GB, doté d'une grande puissance de calcul et d'un disque SSD qui devrait garantir un fonctionnement optimal même durant les pics de fréquentation.

Système : Ubuntu Server 18.04.1 LTS

Logiciels :

- Python version 3.6 ou supérieure
- Python PIP
- Environnement virtuel conforme au fichier requirements.txt du dépôt
- NGINX version 1.15.5

#### 2.1.2 - *Serveur Base de données*

Serveur de base de données hébergeant la base de données de l'application.

Système : Linux.

Logiciels :

- PostgreSQL v11 (supportée jusqu'au 09/11/2023)

#### 2.1.3 - *Serveur Web & Fichiers*

Serveur web répondant aux requêtes et hébergeant et servant les fichiers statiques de l'application.

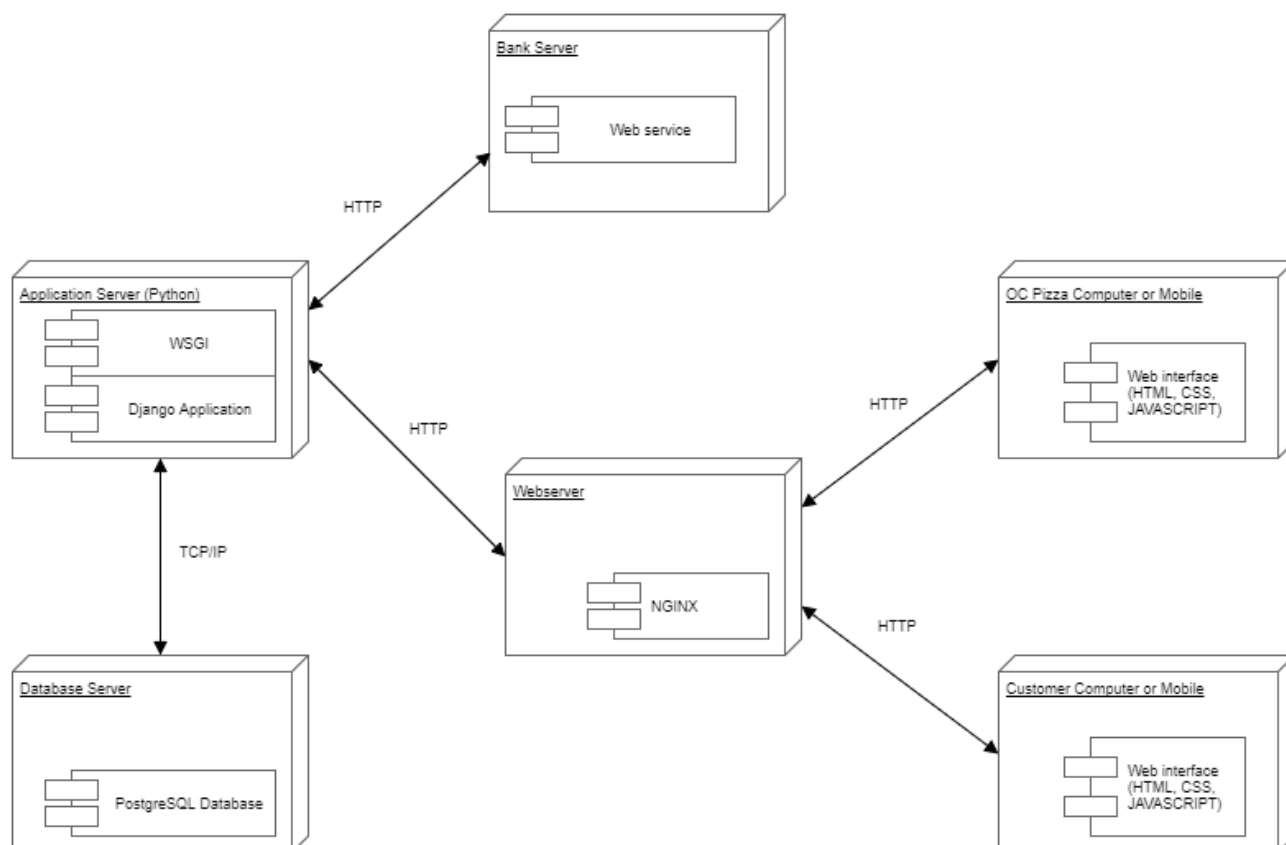
Système : Linux.

Logiciels : NGINX version 1.15.5

- uWSGI 2.0.15 ou supérieure

# 3 - PROCÉDURE DE DÉPLOIEMENT

## 3.1 - Schéma général



## 3.2 - Création de la base de données

Une base de donnée doit être créée sur le serveur de base de données, ainsi qu'un utilisateur ayant tous les privilèges sur cette base.

## 3.3 - Copier les fichiers de l'application

Le dépôt doit être cloné sur le serveur, dans un répertoire appartenant à un utilisateur ayant des droits administrateurs que le système.

## 3.4 - Installer les modules python nécessaires

Un environnement virtuel Python doit être créé sur le serveur puis activé.

Dans cet environnement virtuel installer toutes les dépendances listées dans le fichier requirements.txt

## 3.5 - Gestion des fichiers statiques

Lancer la commande `manage.py collectstatic` pour rassembler les fichiers statiques dans le dossier indiqué dans la variable `STATIC_ROOT` définies dans `settings/__init__.py`.

## 3.6 - Fichier de configuration de production

Un fichier nommé `production.py` doit être créé dans le dossier `settings` de l'application. Il **ne doit en aucun être versionné** car contenant des données sensibles.

Les variables suivantes doivent être renseignées :

```
ALLOWED_HOSTS = [<domaine>]
```

```
SECRET_KEY = <à générer>
```

```
DEBUG = False
```

```
DATABASES = {
```

```
    'default': {
```

```
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
```

```
        'NAME': '<nom de la base de données>',
```

```
        'USER': '<nom de l'utilisateur de la base de données>',
```

```
        'PASSWORD': '<mot de passe de la base de données>',
```

```
        'HOST': 'localhost',
```

```
        'PORT': '5432',
```

```
}  
}
```

## 3.7 - Installation et configuration de NGINX

### 3.7.1 - Installation de NGINX

Le serveur NGINX doit être installé suivant le protocole habituel de la distribution Linux.

### 3.7.2 - Configuration de NGINX

Créer un fichier de configuration NGINX qui va rediriger le trafic vers le dossier du projet en suivant ce modèle (pour les fichiers statics il convient d'indiquer le chemin du serveur web en première option) :

```
server {  
    location /static {  
        alias /home/<nom utilisateur>/<chemin du projet>/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;  
        }  
    }  
}
```

## 3.8 - Installation de Gunicorn

Gunicorn est normalement installé avec les modules requis listés dans requirements.txt.

## 3.9 - Installation et configuration de Supervisor

### 3.9.1 - Installation de Supervisor

Supervisor doit être installé suivant le protocole habituel de la distribution Linux.

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

### 3.9.2 - Configuration de Supervisor

Créer un fichier oc-pizzas-gunicorn.conf en suivant ce modèle :

```
[program:oc-pizzas-gunicorn]
```

```
command = /home/<chemin du projet>/env/bin/gunicorn disquaire_project.wsgi:application
```

```
user = <nom utilisateur>
```

```
directory = /home/<chemin du projet>
```

```
autostart = true
```

```
autorestart = true
```

```
environment = SECRET_KEY=<clé générée>
```

```
environment = DJANGO_SETTINGS_MODULE='oc_pizzas.settings.production'
```

### 3.9.3 - Lancer le processus

Pour lancer le processus, utiliser les commandes suivantes :

```
sudo supervisorctl reread
```

```
sudo supervisorctl update
```

Pour vérifier le processus, utiliser la commandes suivante :

```
sudo supervisorctl status
```

Pour relancer le processus, utiliser la commandes suivante :

```
sudo supervisorctl restart oc-pizzas-gunicorn
```

### 3.9.4 - DataSources

Les accès aux bases de données doivent se configurer à l'aide des fichiers...



## 4 - PROCÉDURE DE DÉMARRAGE / ARRÊT

### 4.1 - Installation et configuration de Supervisor

#### 4.1.1 - Installation de Supervisor

Supervisor doit être installé suivant le protocole habituel de la distribution Linux.

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

#### 4.1.2 - Configuration de Supervisor

Créer un fichier oc-pizzas-gunicorn.conf en suivant ce modèle :

```
[program:oc-pizzas-gunicorn]
command = /home/<chemin du projet>/env/bin/gunicorn disquaire_project.wsgi:application
user = <nom utilisateur>
directory = /home/<chemin du projet>
autostart = true
autorestart = true
environment = SECRET_KEY=<clé générée>
environment = DJANGO_SETTINGS_MODULE='oc_pizzas.settings.production'
```

### 4.2 - Lancer le processus

Pour lancer le processus, utiliser les commandes suivantes :

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

Pour vérifier le processus, utiliser la commandes suivante :

```
sudo supervisorctl status
```

Pour relancer le processus, utiliser la commandes suivante :

```
sudo supervisorctl restart oc-pizzas-gunicorn
```

# 5 - PROCÉDURE DE MISE À JOUR

## 5.1 - Application

Dans le cadre du contrat de maintenance nous nous engageons à effectuer les interventions nécessaires dans les cas suivants :

- mises à jour de sécurité de l'application Django
- bug de l'application oc-pizzas ou de ses composants

Suivant la durée estimée et l'heure de chaque intervention, un serveur de secours pourra être déployé.

## 5.2 - Base de donnée

Dans le cadre du contrat de maintenance nous nous engageons à effectuer les interventions nécessaires dans les cas suivants :

- mises à jour de sécurité de PostgreSQL

Suivant la durée estimée et l'heure de chaque intervention, un serveur de secours pourra être déployé.

## 5.3 - Front-office

Dans l'hypothèse – peu probable – d'une faille critique avec Javascript, nous effectuerons les mises à jour nécessaires dans les mêmes conditions que décrites ci-dessus.

# 6 - SUPERVISION/MONITORING

## 6.1 - Logs de l'application

Afin de tester que l'application web est toujours fonctionnelle nous allons mettre en place différentes solutions techniques :

1. système natif de logging Django
2. rapport des loggings dans le service en ligne Sentry
3. alertes natives de Django par emails aux administrateurs en cas d'erreurs 500 et 400

## 6.2 - Monitoring du serveur

La surveillance du serveur sera assurée par l'installation de Prometheus et Grafana, outils open source de monitoring serveur.

# 7 - PROCÉDURE DE SAUVEGARDE ET RESTAURATION

## 7.1 - Sauvegardes

Une tâche CRON effectuera une sauvegarde quotidienne de la base de données et enverra cette sauvegarde sous la forme d'une archive cryptée sur un deuxième serveur, utilisé uniquement pour le stockage de ses sauvegardes et non accessible autrement qu'en ssh.

Ce serveur aura également une copie de l'application complète et de sa configuration.

En cas d'urgence ou de serveur principal totalement hors service pour une durée indéterminée, il serait possible de rapidement relancer l'application et faire une redirection DNS vers ce serveur pour rétablir le service (toutefois avec des performances moindres).

## 7.2 - Restauration

Conformément au contrat de maintenance, en cas de bug ou attaque sur le serveur, les données pourront être rétablies sous deux heures depuis les données disponibles dans le serveur de sauvegardes, avec des mots de passes réinitialisés pour les comptes d'administrations des différents composants (application, base de données, et compte utilisateur du serveur si nécessaire) .