

Loïc ROMERO

Analyste-programmeur

loic.romero1@gmail.com

**Résumé**

*Il s’agit du document contenant les informations dont l'équipe d'exploitation a besoin pour pouvoir assurer une exploitation en règle du système et pouvoir réagir de manière appropriée lorsqu'un problème surgit.*

*Version 1.1 – 03/04/2021*

Dossier d’exploitation

**« Projet 9 : Documentez votre système de gestion de pizzeria »**



**Table des matières**

[1 - Versions 2](#_Toc68358360)

[2 - Introduction 3](#_Toc68358361)

[2.1 - Objet du document 3](#_Toc68358362)

[2.2 - Références 3](#_Toc68358363)

[3 - Pré-requis 4](#_Toc68358364)

[3.1 - Système 4](#_Toc68358365)

[3.1.1 - Nom de domaine et IP 4](#_Toc68358366)

[3.1.2 - Serveur de Base de données 4](#_Toc68358367)

[3.1.3 - Serveur Web 4](#_Toc68358368)

[3.1.4 - Serveur de Fichiers 4](#_Toc68358369)

[4 - Procédure de déploiement 5](#_Toc68358370)

[4.1 - Initialisation 5](#_Toc68358371)

[4.1.1 - Installation 5](#_Toc68358372)

[4.1.2 - Variables d'environnement 5](#_Toc68358373)

[4.2 - Déploiement 6](#_Toc68358374)

[4.2.1 - Application web 6](#_Toc68358375)

[4.2.2 - Application OCPizza 7](#_Toc68358376)

[4.2.3 - Vérifications 7](#_Toc68358377)

[5 - Procédure de démarrage / arrêt 8](#_Toc68358378)

[6 - Procédure de mise à jour 9](#_Toc68358379)

[6.1 - Base de données 9](#_Toc68358380)

[6.2 - Application web 9](#_Toc68358381)

[7 - Supervision/Monitoring 10](#_Toc68358382)

[7.1 - Gestion des logs 10](#_Toc68358383)

[7.2 - Supervision de l’application web 10](#_Toc68358384)

[8 - Procédure de sauvegarde et restauration 11](#_Toc68358385)

[9 - Glossaire 12](#_Toc68358386)

# Versions

|  |  |  |  |
| --- | --- | --- | --- |
| Auteur | Date | Description | Version |
| Loïc ROMERO | 15/03/2021 | Création du document | 1.0 |
| Loïc ROMERO | 03/04/2021 | Relecture | 1.1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Objet du document

Document contenant les informations dont l'équipe d'exploitation a besoin pour pouvoir assurer une exploitation en règle du système et pouvoir réagir de manière appropriée lorsqu'un problème surgit.

## Références

Le présent *document, «***PDOCPizza\_03\_exploitation.pdf »** *est disponible à l’adresse suivante :*

* *[https://github.com/ROL-1/P9-Documentez](https://github.com/ROL-1/P6-Solution-Technique)*

Pour de plus amples informations, se référer également aux éléments suivants :

1. **PDOCPizza\_01\_fonctionnel.pdf – [version 3.1] :** Dossier de conception fonctionnel
2. **PDOCPizza\_02\_technique.pdf – [version 2.1] :** Dossier d’exploitation
3. **PDOCPizza\_04\_PV\_livraison.pdf :** PV de livraison

# Pré-requis

## Système

### Nom de domaine et IP

Il est possible d’accéder à l’application directement depuis l’adresse suivante :

***ocpizza.rol1.fr***

Le sous domaine « ocpizza » du domaine « rol1.fr » est dédié à cette application. Il s’agit d’un hébergement réservé sur le serveur **DigitalOcean**, accessible par le DNS réservé via **OVH**.

Il pointe vers l’adresse IP publique suivante :

159.89.8.181

### Serveur de Base de données

La base de données est hébergée sur le serveur lié au droplet DigitalOcean. C’est une base de données **PostgreSQL**, version 13.2 au moment de la livraison.

### Serveur Web

**Nginx** est le serveur web virtuel mis à disposition par le droplet DigitalOcean.

Le fichier de configuration se trouve à l’adresse suivante : *etc/nginx/sites-avaible/ocpizza*

Il indique le port http à écouter, le nom de domaine et/ou son adresse ip publique, la localisation de l’application sur le serveur. Il renseigne l’emplacement des fichiers statics. Et redirige le trafic vers l’application django quand un fichier dynamique est demandé.

Le service peut être relancé à l’aide de la commande :

sudo service nginx reload

### Serveur de Fichiers

**Gunicorn** est le serveur HTTP Python pour Unix qui utilise les spécifications WSGI (Web Server Gateway Interface) (cf. 4.3.2).

Ce **serveur** **de** **production** est lancé par la commande :

gunicorn ocpizza.wsgi:application

# Procédure de déploiement

## Initialisation

### Installation

Extraire l'archive **ocpizza.zip** dans le répertoire :

**home/<user>/ocpizza**

On trouvera les éléments suivant à la racine du projet :

* **manage.py :** fichier de configuration de l’utilitaire de ligne de commande Django, il défini la variable d’environnement permettant de pointer vers le fichier de configuration (cf. 4.1.2).
* **pipfile et pipfile.lock :** nécessaires à la création de l’environnement virtuel avec pipenv.
* **README.md :** fichier décrivant le projet.
* **docs :** dossier contenant de la documentation concernant l’application.
* **ocpizza :** package contenant les fichiers de configuration de l’application Django.
* **shopping :** package contenant les modules dédiés à l’interface d’achat (partie client).
* **restaurant :**package contenant les modules dédiés aux employés (stock, gestion des commandes, etc…)
* **administration :** package contenant les modules dédiés à la direction du groupe et aux responsables de point de vente (« dashboard », recettes, etc…)
* **tests :** package contenant les tests.

### Variables d'environnement

Voici les variables d'environnement de l’application « ocpizza » :

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Valeur** | **Requis** | **Description** |
| ENV | PRODUCTION | OUI | Défini qu’il s’agit de l’environnement de production |
| SECRET\_KEY | - | OUI | Clef secrète de production, à ne pas divulguer publiquement. |
| DJANGO\_SETTINGS\_MODULE | ocpizza.settings.production | OUI | Défini à Django où est situé le fichier de configuration |

**La clef SSH privée** permettant de se connecter avec le compte administrateur sera transmise dans un fichier sécurisé au moment de la livraison

## Déploiement

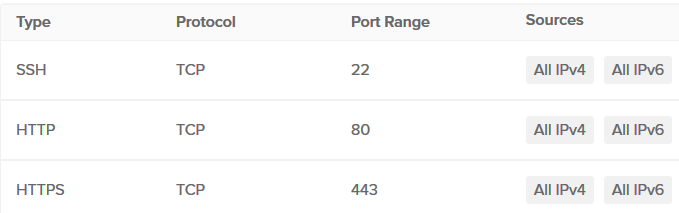
### Application web

Le serveur est mis à disposition par les services de **DigitalOcean,** dans un espace « droplet » (cf 5.2). Il est accessible à **l’adresse IP publique** :

159.89.8.181

**La clef SSH privée** permettant de se connecter avec le compte administrateur sera transmise dans un fichier sécurisé au moment de la livraison.

**Parfeu :** des règles de parfeu ont étés créées pour permettre les connections selon les usages :



#### Datasources

La base de données est également hébergée sur le serveur lié au droplet DigitalOcean. C’est une base de données **PostgreSQL**.

Les informations de connections à la base de données sont disponibles dans le module : *purbeurre/settings/\_\_init\_\_.py et purbeurre/settings/production.py*

#### Fichier *purbeurre/settings/\_\_init\_\_.py*

Ce module contient les informations complètes permettant la configuration de ce projet Django pour le développement ainsi que la clef secrète de développement.

#### Fichier *purbeurre/settings/production.py*

Ce module contient les informations permettant de se connecter au serveur de production, à la base de données de production et **contient la clef secrète de production.**

Ce module sert également à faire appel et configurer l’outil de remontée et suivi des logs avec Sentry.

***Ce fichier n’est pas suivi sur git.***

#### Fichier *purbeurre/settings/travis.py*

Ce module sert uniquement à l’intégration continue (CI) pendant le développement et contient les informations permettant de se connecter à la base de données créée spécifiquement pour les tests d’intégration continue.

### Application OCPizza

#### Fichiers statics

Les **fichiers statics** sont contenus dans le package : *ocpizza/staticfiles*

Ils sont générés par la commande :

python ocpizza/manage.py collecstatic

#### Nginx

Nginx est le **serveur** **web** (cf. 4.3.2). Le fichier de configuration se trouve à l’adresse suivante : *etc/nginx/sites-avaible/ocpizza*

Il indique le port http à écouter, le nom de domaine et/ou son adresse ip publique, la localisation de l’application sur le serveur. Il renseigne l’emplacement des fichiers statics. Et redirige le trafic vers l’application django quand un fichier dynamique est demandé.

Le service peut être relancé à l’aide de la commande :

sudo service nginx reload

#### Gunicorn

Gunicorn est le serveur HTTP Python pour Unix qui utilise les spécifications WSGI (Web Server Gateway Interface) (cf. 4.3.2).

Ce **serveur** **de** **production** est lancé par la commande :

gunicorn ocpizza.wsgi:application

#### Supervisor

Supervisor est le service chargé de s’assurer que le serveur fonctionne et de le relancer au besoin. Le fichier de configuration se trouve à l’adresse suivante :  
*etc/supervisor/conf.d/ocpizza-gunicorn.conf.*

Il contient la variable d’environnement nécessaire à DJANGO :

DJANGO\_SETTINGS\_MODULE = ‘ocpizza.settings.production’

C’est ce service qui est en charge de lancer la commande de lancement du serveur (cf. 6.3.2.3).

Le processus peut être lancé avec les commandes suivantes :

sudo supervisorctl reread

sudosupervisorctl update

### Vérifications

Afin de vérifier le bon déploiement de l’application, se rendre à l’adresse suivante :

*www.ocpizza.rol1.fr*

# Procédure de démarrage / arrêt

Il est possible de **désactiver** l’intégralité du droplet directement depuis l’interface **DigitalOcean.**

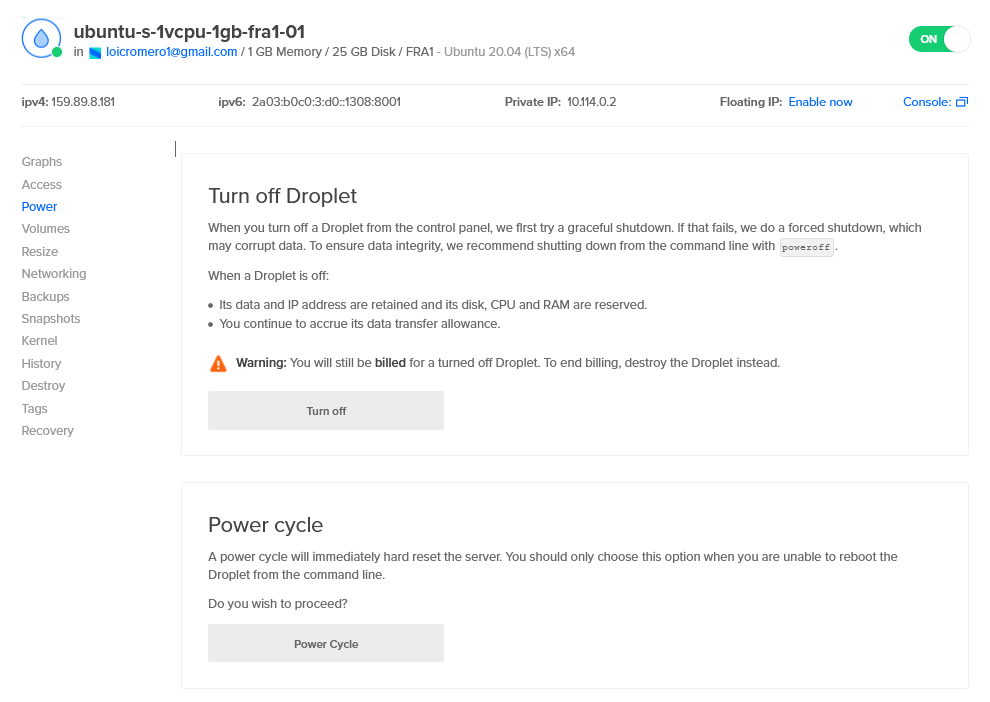
Grâce au bouton « ON/OFF » ou avec le bouton « Turn off ».

**Les données peuvent être altérées par cette méthode.** Il est recommandé d’utiliser la ligne de commande suivante, pour éviter la procédure forçant l’extinction brutale :

poweroff

**L’activation** s’effectue par les mêmes boutons ou la commande suivante :

poweron



# Procédure de mise à jour

## Base de données

L’installation de la base de données s’effectue par la commande personnalisée créée avec Django et le module manage.py :

python manage.py database

Cette commande sollicite automatiquement les modules nécessaires à la récolte des données depuis l’api. Si la base de données est déjà remplie elle va supprimer toutes les données en dehors des comptes utilisateurs.

Pour une simple mise à jour de la base de données il est possible d’utiliser la commande :

python manage.py update

Cette commande n’efface pas les données et n’ajoutera que les nouvelles informations.

Cette commande est lancée de manière automatique par une tâche Cron sa fréquence est :

Hebdomadaire, le dimanche, à minuit.

Tâche dans le fichier *crontab* :



## Application web

La mise à jour de l’application web pourra faire l’objet d’un échange avec l’équipe de IT Consulting & Developement au regard de l’accompagnement contractuel ou par le biais d’une nouvelle négociation de prestation.

# Supervision/Monitoring

## Gestion des logs

Les modules ont été créés en y incorporant la librairie sentry-sdk. Installable via :

pip install sentry-sdk

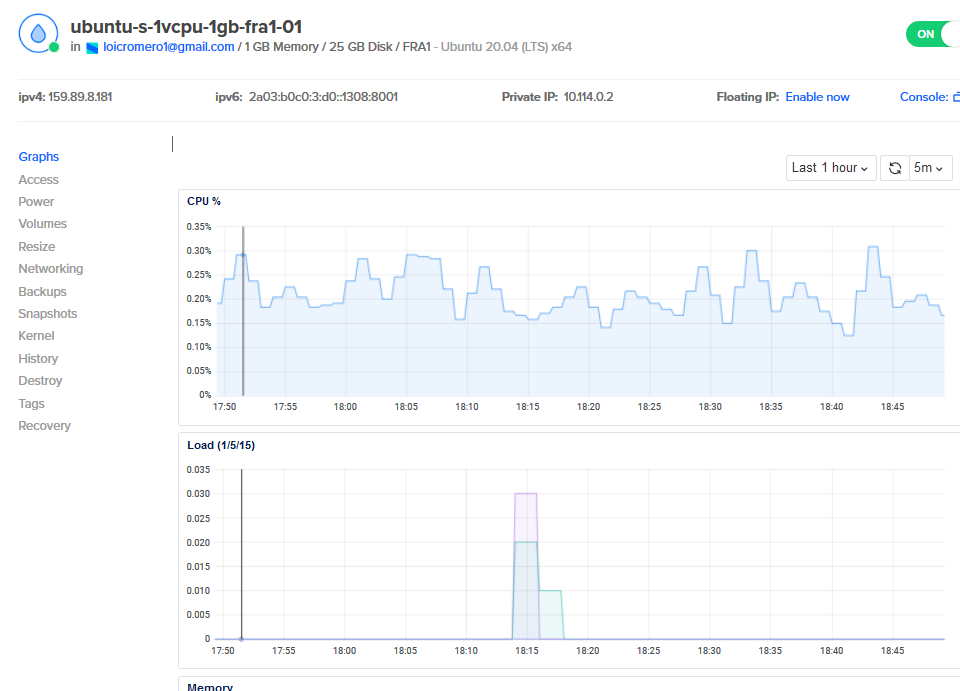
Cela nous a permis de créer des logs captés par l’application **Sentry**. Cette plateforme de **monitoring** permet de tracker les erreurs en temps réel. Des logs ont également étés créés au niveau « info » pour permettre de suivre l’exécution de certaines tâches (mise à jour hebdomadaire de la base de données ».

## Supervision de l’application web

**DigitalOcean** intègre des outils de monitoring. Ils ont étés activés par la commande :

curl -sSL https://repos.insights.digitalocean.com/install.sh | sudo bash

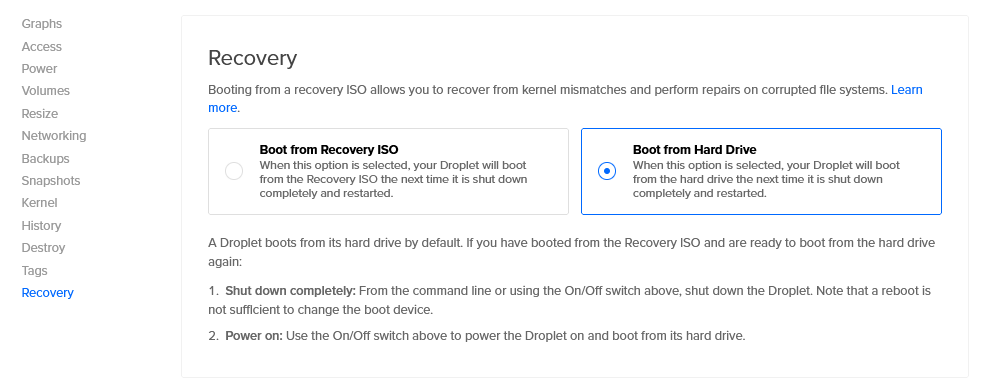
Cela donne accès à des **graphiques** permettant de suivre l’état du serveur : Disk Usage, CPU usage, Bandwidth, Memory, etc…



Dans la rubrique « Monitoring » il est également possible de créer des alertes sur ces mêmes critères, avec « alert policies ». Les alertes seront transmises à l’adresse mail référencé.

# Procédure de sauvegarde et restauration

**DigitalOcean** intègre un outil de restauration : « Recovery ».



Le processus est en trois étapes :

* 1. Charger l’ISO de restauration (après avoir éteint le droplet).
  2. Préparer le droplet pour la restauration.
  3. Restaurer les données.

Ces étapes sont détaillées dans le lien suivant. Elles varient selon la cause nécessitant une restauration.

<https://www.digitalocean.com/docs/droplets/resources/recovery-iso/>

# Glossaire

|  |  |
| --- | --- |
| AI | « Auto Incremental » : attribut auto-incrémenté d’une table SQL. |
| BDD | Base De Données |
| Composant | Un composant UML est une unité autonome fournissant un service précis. Sa vocation est d’être réutilisable |
| Employés | Désigne généralement à la fois, les employés, les responsables de points de vente et la direction du groupe. |
| FK | « Foreign Key » : clef secondaire d’une table SQL. |
| Framework | Désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d’une partie d'un logiciel. |
| PK | « Primary Key » : clef primaire d’une table SQL. |
| Pizzaïolo | Préparateur de pizza |