

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 1.0 – 15/03/2021*

Dossier d’exploitation

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



**Table des matières**

[1 - Versions 2](#_Toc66718538)

[2 - Introduction 3](#_Toc66718539)

[2.1 - Objet du document 3](#_Toc66718540)

[2.2 - Références 3](#_Toc66718541)

[3 - Pré-requis 4](#_Toc66718542)

[3.1 - Système 4](#_Toc66718543)

[3.1.1 - Serveur de Base de données 4](#_Toc66718544)

[3.1.2 - Serveur Web 4](#_Toc66718545)

[3.1.3 - Serveur de Batches 4](#_Toc66718546)

[3.1.4 - Serveur de Fichiers 4](#_Toc66718547)

[3.2 - Bases de données 4](#_Toc66718548)

[3.3 - Web-services 4](#_Toc66718549)

[3.4 - Autres Ressources 4](#_Toc66718550)

[4 - Procédure de déploiement 5](#_Toc66718551)

[4.1 - Déploiement des Batches 5](#_Toc66718552)

[4.1.1 - Artefacts 5](#_Toc66718553)

[4.1.2 - Variables d'environnement 5](#_Toc66718554)

[4.1.3 - Configuration 5](#_Toc66718555)

[4.1.4 - Ressources 6](#_Toc66718556)

[4.1.5 - Vérifications 6](#_Toc66718557)

[4.2 - Déploiement de l'Application Web 6](#_Toc66718558)

[4.2.1 - Artefacts 6](#_Toc66718559)

[4.2.2 - Environnement de l’application web 6](#_Toc66718560)

[4.2.3 - Répertoire de configuration applicatif 6](#_Toc66718561)

[4.2.4 - DataSources 7](#_Toc66718562)

[4.2.5 - Ressources 7](#_Toc66718563)

[4.2.6 - Vérifications 7](#_Toc66718564)

[5 - Procédure de démarrage / arrêt 8](#_Toc66718565)

[5.1 - Base de données 8](#_Toc66718566)

[5.2 - Batches 8](#_Toc66718567)

[5.3 - Application web 8](#_Toc66718568)

[6 - Procédure de mise à jour 9](#_Toc66718569)

[6.1 - Base de données 9](#_Toc66718570)

[6.2 - Batches 9](#_Toc66718571)

[6.3 - Application web 9](#_Toc66718572)

[7 - Supervision/Monitoring 10](#_Toc66718573)

[7.1 - Supervision de l’application web 10](#_Toc66718574)

[8 - Procédure de sauvegarde et restauration 11](#_Toc66718575)

[9 - Glossaire 12](#_Toc66718576)

# Versions

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

# 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, «***Dossier d’explitation.pdf »** *est disponible à l’adresse suivante :*

* [*https://github.com/ROL-1/P6-Solution-Technique*](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] :** Dossier de conception fonctionnel, disponible à l’adresse suivante :
2. **PDOCPizza\_03\_exploitation.pdf – [VERSION] :** Dossier d’exploitation, disponible à l’adresse suivante :
3. **PDOCPizza\_04\_PV\_livraison.pdf – [VERSION] :** PV de livraison, disponible à l’adresse suivante :

# Pré-requis

## Système

### Nom de domaine

### 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

## AVANT

### Github

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

**home/<user>/ocpizza**

**détailler premier niveau d’arbo**

### Variables d'environnement

Voici les variables d'environnement reconnues par les batches de l’application XXX :

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Valeur** | **Requis** | **Description** |
| XXX\_HOME |  | NON | Répertoire racine de l’installation de l’application |
| 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 |

Définissez les variables d’environnement nécessaires comme ceci...

## Déploiement des Batches

### Artefacts

Les batches de l’application Xxx sont construits sous la forme d'une archive ZIP contenant les répertoires :

* **bin** : les scripts SH de lancement des différents batches
* **conf** : les fichiers de configuration
* **etc/nginx/sites-avaible/ocpizza :** fichier de configuration Nginx
* **etc/supervisor/conf.d/ocpizza-gunicorn.conf :** fichier de configuration de Supervisor

Positionner les droits d'exécution sur les scripts SH de lancement des batches.

….

...

### Configuration

Voici les différents fichiers de configuration :

* **xxx.yyy** : fichier de configuration des logs
* **zzz.ttt** : fichier de configuration de l'application...
* **aaa** : fichier de configuration de la ressources XXX
* ...

#### Fichier xxx.yyy

…

#### Fichier zzz.ttt

...

#### Fichier ...

### Ressources

...

### Vérifications

Afin de vérifier le bon déploiement des batches, faire ceci…

## Déploiement de l'Application Web

### Artefacts

...

### Environnement de l’application web

#### Variables d’environnement

Le serveur d'application JOnAS doit être exécuté avec la variable d'environnement suivante définie au démarrage. Elle est nécessaire afin de récupérer le répertoire contenant les fichiers de configuration de l'application :

**-Dcom.ocpizza.apps.conf=$home\_application\_conf\_directory**

INFO : il ne faut pas mettre de « / » à la fin de la valeur de la variable et ne pas utiliser d'espace dans le chemin.

### Répertoire de configuration applicatif

Le répertoire de configuration applicatif doit être créé sur le système de fichier et définit de la façon suivante :

**$home\_application\_conf\_directory/applicationX**

… fichiers de configuration… :

* ...

#### Fichier xxx.yyy

...

### DataSources

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

Le fichier de drivers **postgresql (postgresql-9.2.x.)** doit être déposé dans le répertoire :

**$home\_server/lib/ext**

...

### Ressources

...

### Vérifications

Afin de vérifier le bon déploiement de l’application, faire ceci…

# Procédure de démarrage / arrêt

## Base de données

## Application web

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

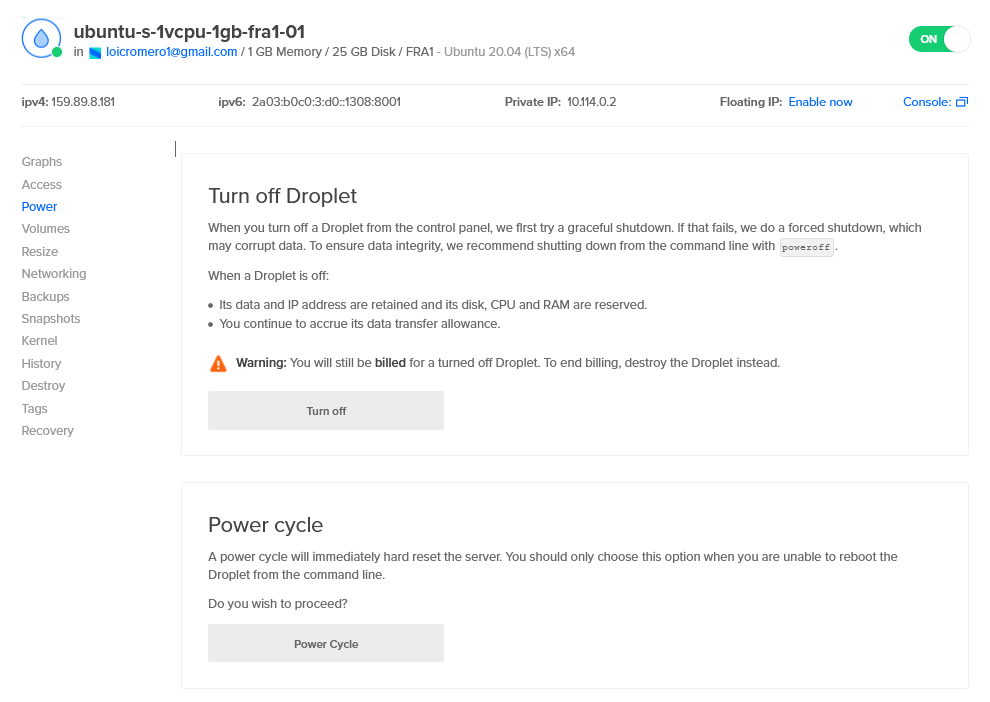
Grâce au bouton « ON/OFF » ou avec le bouton « Turn off ».

**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

?

## Application web

?

# Supervision/Monitoring

## Gestion des logs

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

pip install sentry-sdk

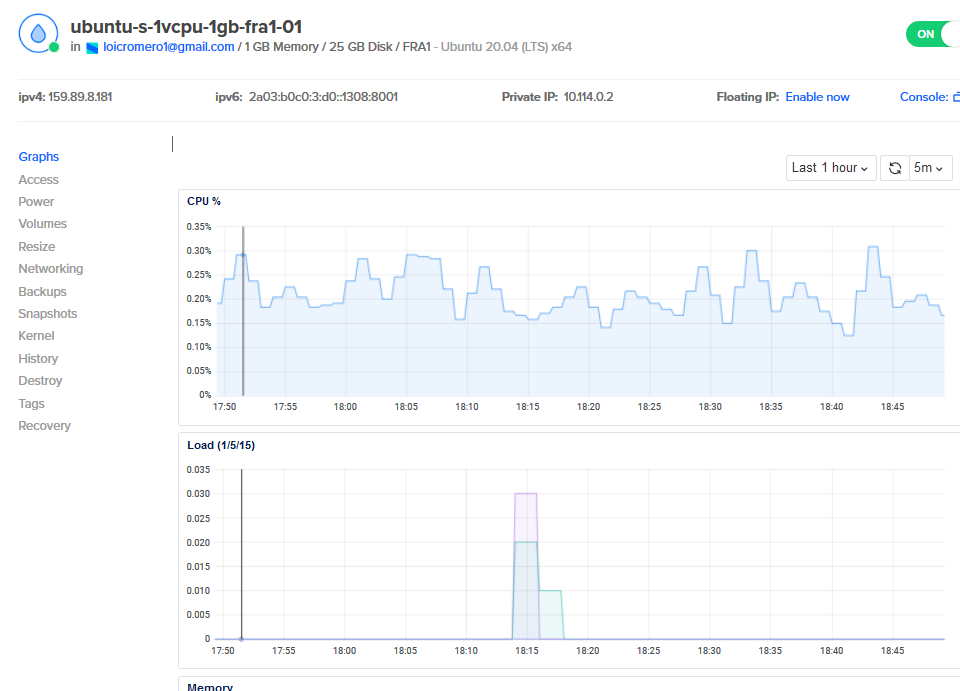
Cela nous a permis de créer des logs captés par l’application **Sentry**. Cette plateforme de **monitoring** permet de tracker les erreurs en temps réel. Des logs ont également étés créés au niveau « info » pour permettre de suivre l’exécution de certaines tâches (mise à jour hebdomadaire de la base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

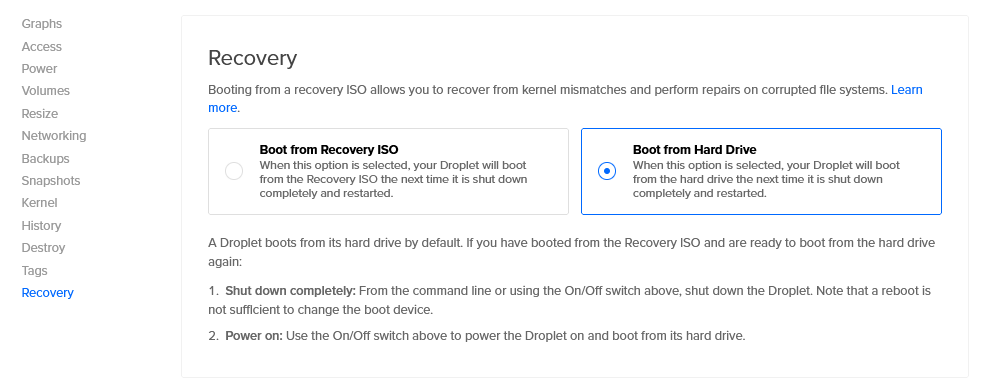
Cela donne accès à des **graphiques** permettant de suivre l’état du serveur : Disk Usage, CPU usage, Bandwidth, Memory, etc…



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 |