

OcPizza

Projet expansion OcPizza

Dossier d'exploitation

Version 1.0

**Auteur**

Benjamin Mourgues  
Analyste-programmeur

# TABLE DES MATIÈRES

<b>1 - Versions.....</b>	<b>4</b>
<b>2 - Introduction.....</b>	<b>5</b>
2.1 - Objet du document.....	5
2.2 - Références.....	5
<b>3 - Pré-requis.....</b>	<b>6</b>
3.1 - Système.....	6
3.1.1 - Serveur Web.....	6
3.1.2 - Hébergement.....	6
3.2 - Bases de données.....	6
3.3 - Web-services.....	6
<b>4 - Procédure de déploiement.....</b>	<b>7</b>
4.1 - Déploiement de l'application.....	7
4.1.1 - Variables d'environnement.....	7
4.1.2 - Configuration.....	9
<b>5 - Procédure de démarrage / arrêt.....</b>	<b>10</b>
5.1 - Serveur Linode.....	10
5.2 - Application web.....	10
<b>6 - Procédure de mise à jour.....</b>	<b>11</b>
6.1 - Application web.....	11
<b>7 - Procédure de sauvegarde et restauration.....</b>	<b>12</b>
7.1 - Base de données.....	12
7.2 - Application web.....	12
<b>8 - Supervision/Monitoring.....</b>	<b>14</b>
8.1 - Supervision de l'application web.....	14
8.2 - Supervision du serveur.....	14
<b>9 - Glossaire.....</b>	<b>17</b>

# 1 - VERSIONS

Auteur	Date	Description	Version
Benjamin	08/04/2022	Création du document	1.0

## 2 - INTRODUCTION

### 2.1 - Objet du document

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

Il regroupe les informations nécessaires à l'exploitation de l'application par les équipes techniques d'OC Pizza afin de pouvoir déployer et maintenir l'application en cas de besoins.

### 2.2 - Références

Pour de plus amples informations, vous pouvez vous référer aux documents suivants :

1. **DCT - « P9 Dossier de conception technique.pdf »** : Dossier de conception technique de l'application,
2. **DCF - « P9 Dossier de conception fonctionnelle.pdf »** : Dossier de conception fonctionnelle de l'application.
3. **PV - « P9 Procès verbal de livraison.pdf »** : Procès-verbal de livraison finale.

## 3 - PRÉ-REQUIS

### 3.1 - Système

#### 3.1.1 - Serveur Web

Le serveur web utilisé pour l'application est NGINX 1.21.6 et le WSGI est Gunicorn 20.10.

#### 3.1.2 - Hébergement

L'application web est hébergée sur Linode, un service de location de serveurs. Le plan actuel de paiement est Nanode. Il s'agit d'un serveur avec CPU partagé avec d'autres clients qui convient pour un faible trafic, mais une amélioration est possible en optant pour des plans plus importants.

L'application est accessible à l'adresse <https://www.OCPizza.fr/>

### 3.2 - Bases de données

La base de donnée utilisée sera PostgreSQL, dans sa version 12.9.

### 3.3 - Web-services

Les web services suivants doivent être accessibles et à jour :

- **API Google Maps** : Afin de suivre les itinéraires et la position des livreurs, il sera nécessaire d'utiliser cet API, la clé d'authentification est la suivante : sP3+B+wl3vc0+6gh6CFXH7Qfstsoq+6nWXZO6/40mvgdROI82698LFgwCiLExChdstjY/n0=
- **API Paypal** : Paypal est l'un des moyens de paiement les plus populaire au monde, la clé d'authentification pour l'API est la suivante : WMyFcIEZC6zQIH3jj5M2E6W/RFU15zilKWrLhvodtIFDf9wSGJSsO4PDJ4=

# 4 - PROCÉDURE DE DÉPLOIEMENT

## 4.1 - Déploiement de l'application

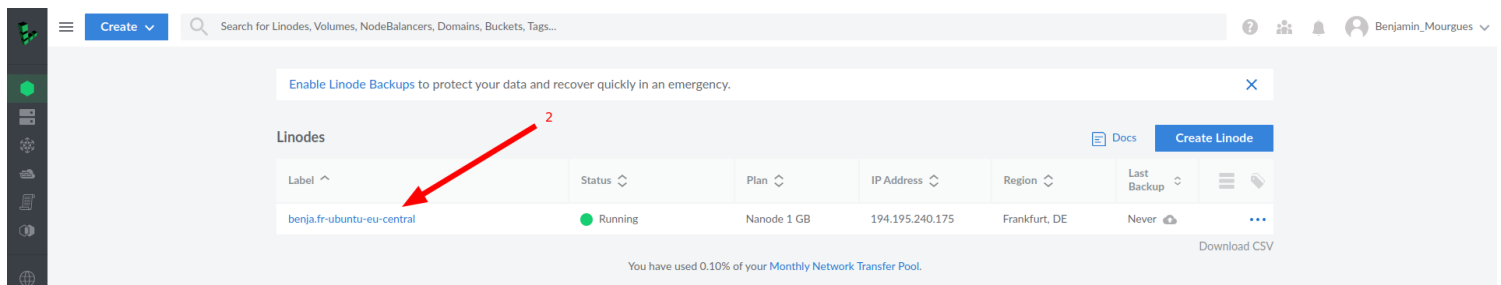
### 4.1.1 - Variables d'environnement

Voici les variables d'environnement à insérer dans Linode :

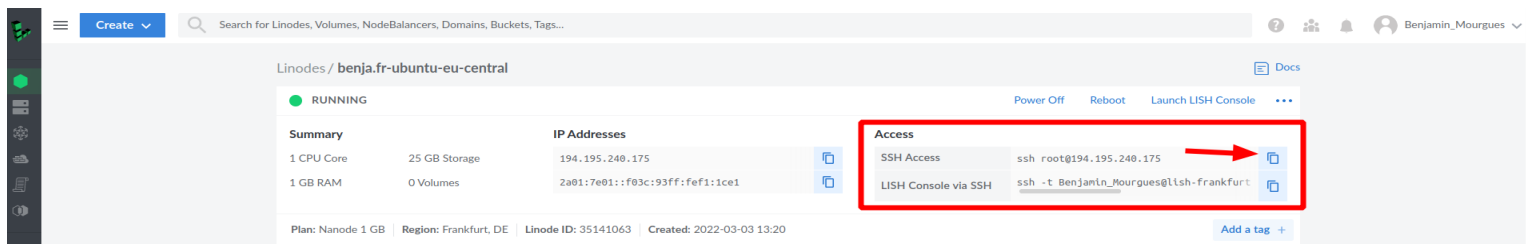
Nom	Valeur	Description
ENV	Production	Permet de signaler à l'application que l'environnement est celui de production. La valeur « dev » aurait pour effet d'activer le mode « debug » qu'il est essentiel d'avoir désactivé tant que l'application est accessible au public.
DJANGO_SECRET_KEY	TbUTzf2B6StO2bM847McfQrcEeab+CmYWwgxQgmFRlly++lYuD0CUQ==	Cette clé est à garder précieusement et secrète, elle est utilisée pour signer de nombreux objets et mots de passe dans l'application.
POSTGRES_USER	Ocpizza	Votre nom d'utilisateur pour accéder à la base de données
POSTGRES_PASSWORD	lpKYaAjpWUBnxlYXUj+eTYkUlfjMnR4gZ7vEAjfjXj2qHkbFV5CZS43AhRfzXWRXDC13gRw=	C'est le mot de passe à utiliser pour s'identifier et accéder à la base de donnée, combiné avec le nom d'utilisateur POSTGRES_USER
POSTGRES_PORT	5432	This is the default port to which PostgreSQL will listen. Make sure this port is available.
POSTGRES_DATABASE	ocpizza_db	Nom de la base de donnée à laquelle l'application web se connectera
POSTGRES_HOST	194.195.241.175	Adresse du serveur hébergeant votre base de donnée. Si vous hébergez sur le même serveur, vous pouvez inscrire « localhost »
MAPS_API_KEY	sP3+B+wI3vc0+6gh6CFXH7Qfstsoq+6nWXZO6/40mvgdROI82698LFgwCiLExCdstjY/n0=	Clé pour accéder à l'API google maps
PAYPAL_API_KEY	WMyFcIEZC6zQlHy3jj5M2E6W/RfUI5ziLKWrLhvodtIFDf9wSGJSsO4PDJ4=	Clé pour accéder à l'API Paypal

Pour définir une variable d'environnement, utilisez l'interface de Linode comme ceci :

1. Rendez vous sur <https://cloud.linode.com/linodes>
2. Cliquez sur votre serveur



3. Copiez l'une des deux commandes, et entrez là dans le terminal



4. Vous êtes connecté à votre serveur. Vous pouvez maintenant enregistrer votre variable d'environnement en utilisant la commande suivante :  
``export NOM_VARIABLE_ENVIRONNEMENT=VALEUR``

```
root@benja:~# export HELLO="bonjour"
root@benja:~# echo $HELLO
bonjour
root@benja:~#
```

#### 4.1.2 - Configuration

Des manipulations préliminaires existent afin de configurer Linode, je vous encourage à suivre le tutoriel de Linode eux-même :

\* <https://www.linode.com/docs/guides/getting-started/>

et par la suite celui de Digital Ocean afin de mettre en place Postgres, NGINX et Unicorn:

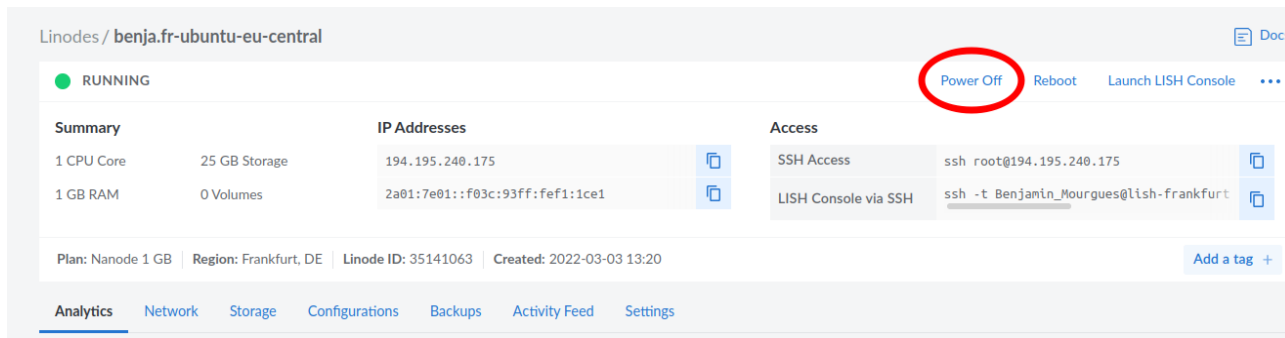
\* <https://www.digitalocean.com/community/tutorials/how-to-set-up-django-with-postgres-nginx-and-unicorn-on-ubuntu-20-04-fr>

Suite à quoi entrez les variables d'environnement dans votre serveur, vous pouvez également utiliser la librairie dotenv et placer les variables d'environnement dans un fichier .env à la racine du dossier contenant l'application.

# 5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

## 5.1 - Serveur Linode

Vous pouvez éteindre totalement le serveur qui héberge votre site web en passant par le site de Linode :



Ce faisant, vous rendez impossible de se connecter au serveur avec la commande SSH, le serveur est totalement hors ligne.

Le même bouton permet ensuite de le rallumer.

## 5.2 - Application web

L'application peut être isolée par exemple en arrêtant le service NGINX, qui sert d'interface entre votre application Django et le web.

Pour ce faire, connectez vous à votre serveur en SSH (en suivant les instructions décrites dans la section 4.1.1), et utilisez la commande suivante :

```
`$ sudo systemctl stop nginx`
```

Cela aura pour effet de rendre votre site inaccessible, mais vous pourrez toujours opérer des modifications sur votre serveur si vous le souhaitez.

Pour le redémarrer, il vous suffira d'entrer la commande suivante:

```
`$ sudo systemctl restart nginx`
```



# 6 - PROCÉDURE DE MISE À JOUR

## 6.1 - Application web

Pour mettre l'application web à jour, il vous faudra éditer le code se trouvant dans le répertoire racine de l'application, sur le serveur.

Cela peut se faire en utilisant un système de contrôle de version tel que Git, et un hébergeur tel que GitHub pour le code. Cela est recommandé pour permettre une traçabilité des changements et revenir à une version antérieure en cas de problèmes.

Dans l'exemple de git, il vous faudra vous connecter au serveur en utilisant la console, vous rendre dans le dossier de l'application (`$ cd /chemin/de/l/application/`) et d'utiliser la commande suivante:

```
| $ git fetch && git checkout <nom de la branche à déployer> && git pull
```

# 7 - PROCÉDURE DE SAUVEGARDE ET RESTAURATION

## 7.1 - Base de données

Avant toute chose, il semble important de rappeler qu'avant toute modification et de manière régulière il est nécessaire de faire une sauvegarde de la base de données.

Sur Linode, il existe un simple bouton qui permet de faire la copie du serveur de la base de donnée :

Linodes / benja.fr-ubuntu-eu-central

**RUNNING** Power Off Reboot Launch LISH Console ...

Summary		IP Addresses	Access
1 CPU Core	25 GB Storage	194.195.240.175	SSH Access ssh root@194.195.240.175
1 GB RAM	0 Volumes	2a01:7e01::f03c:93ff:fef1:1ce1	LISH Console via SSH ssh -t Benjamin_Mourgues@lish-frankfurt

Plan: Nanode 1 GB | Region: Frankfurt, DE | Linode ID: 35141063 | Created: 2022-03-03 13:20

Analytics Network Storage Configurations **Backups** Activity Feed Settings

**Backups**

Three backup slots are executed and rotated automatically: a daily backup, a 2-7 day old backup, and an 8-14 day old backup. To enable backups for just **\$2.00 per month**, click below.

[Enable Backups](#)

Vous pouvez aussi indépendamment mettre en place un dump et sauvegarder cette copie séparément avec une tâche cron par exemple. Ne sauvegardez jamais les backup sur le même serveur que la base de donnée, car vous courrez le risque de perdre toutes les données en même temps sur le serveur physique est détruit (cf. Accident OVH Janvier 2021)

Pour la procédure plus complète, je vous renvoie à la documentation de PostgreSQL :

<https://www.postgresql.org/docs/12/backup.html#BACKUP-DUMP-RESTORE>

## 7.2 - Application web

L'application web est versionnée par l'outil de contrôle de version Git.

Pour revenir à un état antérieur, il conviendra d'utiliser la commande :

```
$ git checkout <sha1 du commit>
```

# 8 - SUPERVISION/MONITORING

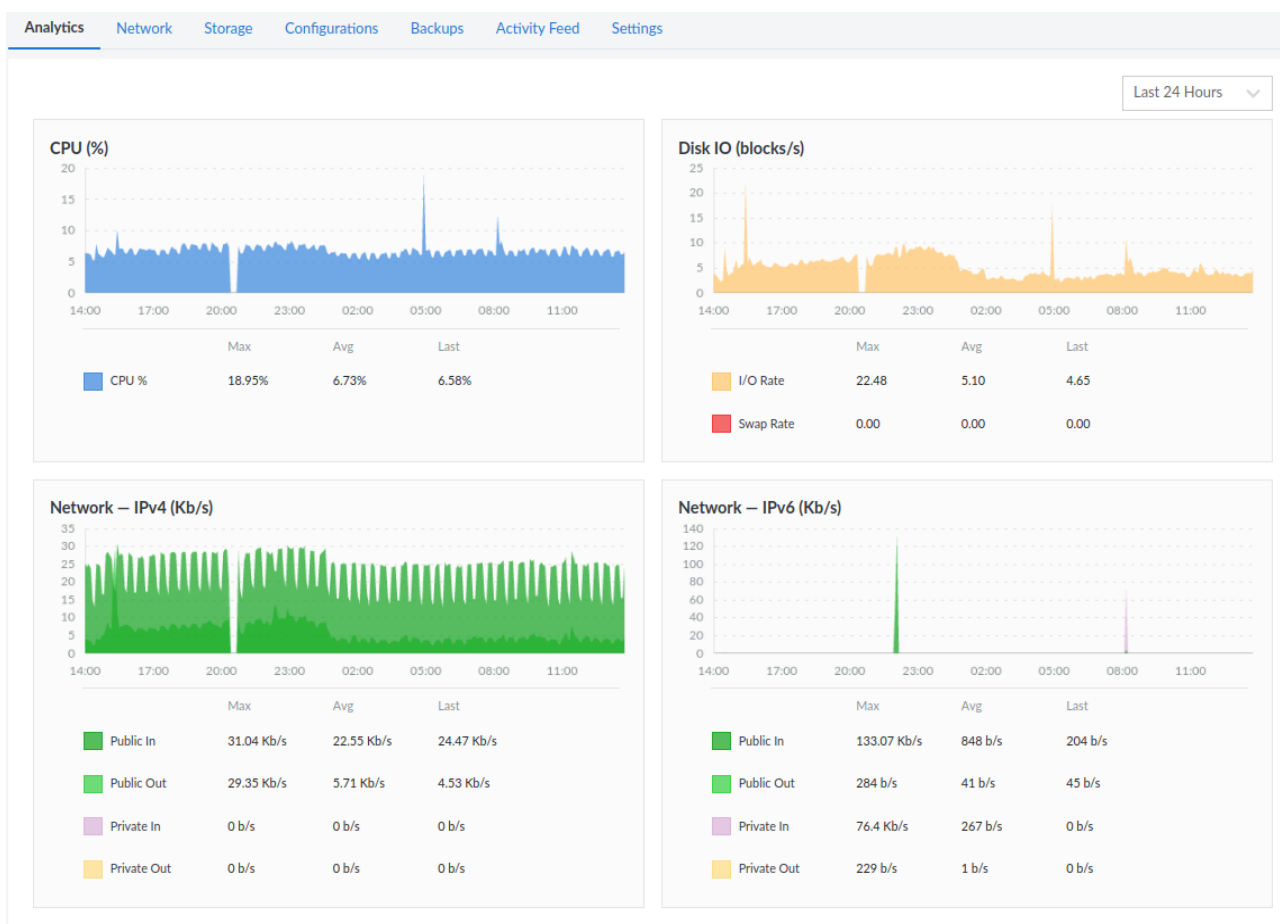
## 8.1 - Supervision de l'application web

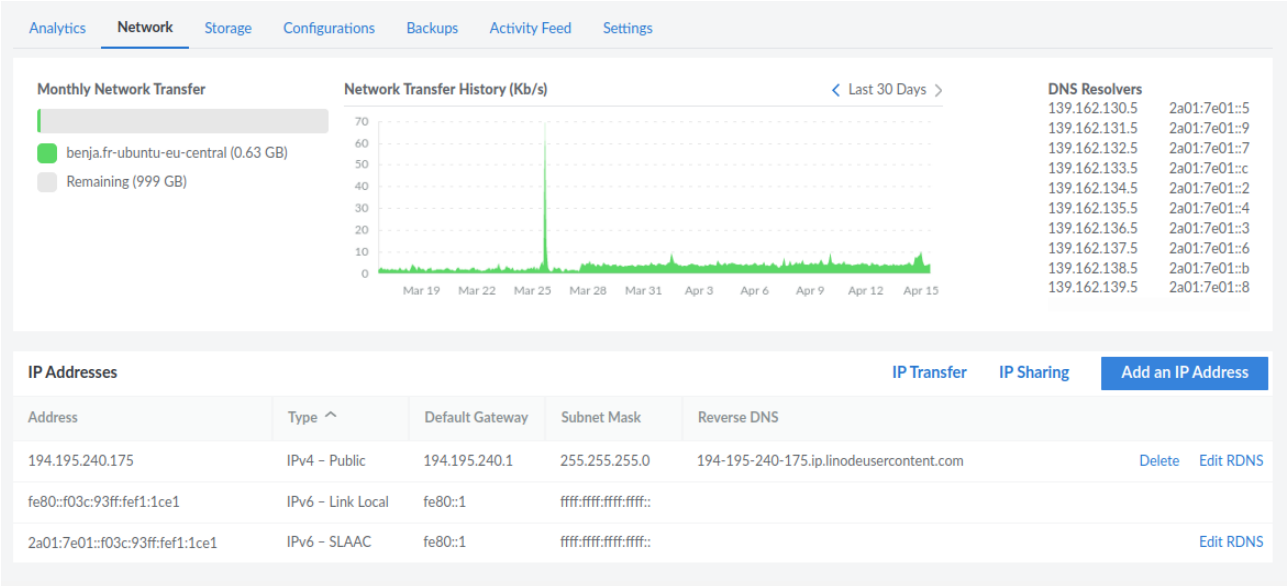
Afin de tester que l'application web est toujours fonctionnelles, prenez soin de vérifier que pour toute mise à jour déployée, un outil de d'intégration continue comme TravisCI valide le passage des tests automatisés, il est par conséquent très important de maintenir une couverture de test importante sur l'ensemble de l'application, et plus particulièrement sur les applications critiques (qui sont votre cœur de métier).

Le projet comprend une installation de Sentry, un service qui vous permettra d'avoir une remontée plus rapide et détaillée des erreurs qui peuvent survenir si certaines passent les mailles du test automatisé. (<lien du dashboard et identifiants>)

## 8.2 - Supervision du serveur

Linode permet un monitoring du serveur sur les performances matérielles et réseau :





## 9 - GLOSSAIRE

<b>API</b>	Interface de programmation
<b>Git</b>	Logiciel de contrôle de version
<b>CPU</b>	Processeur
<b>WSGI</b>	Web Server Gateway Interface
<b>SSH</b>	Secure Shell / Terminal sécurisé
<b>Dump</b>	Cliché mémoire
<b>Cron</b>	Abréviation de <i>Cron Job</i> , Tâche planifiée