

OCPizza

OC PIZZA

**Pour une application web de vente de pizza en ligne
avec sa logistique**

Dossier d'exploitation
Version 1.0



Angelique F.
Responsable Dev

OCPizza

TABLE DES MATIÈRES

1.	Versions	3
2.	Introduction	4
2.1.	Objet du document	4
2.2.	Références	4
3.	Pré-requis	5
3.1.	Système	5
3.2.	Bases de données.....	5
3.3.	Web-services.....	5
3.4.	Contenu de l'application web	5
4.	Procédure de déploiement.....	6
4.1.	Déploiement de l'application web	6
4. 1.1.	Déploiement	6
4. 1.2.	Vérifications.....	6
5.	Procédure de démarrage / arrêt.....	7
6.	Supervision/Monitoring	8
7.	Procédure de sauvegarde et restauration.....	9
7.1.	Sauvegarde	9
7.1.1.1.	Création d'une sauvegarde.....	9
7.1.1.2.	Planification d'une sauvegarde	10
7.2.	Restauration d'une sauvegarde	10
8.	Glossaire	11

OCPizza

1. VERSIONS

Auteur	Date	Description	Version
Angelique Fourny	02/04/2021	Création du document	1.0

OCPizza

2. INTRODUCTION

2.1. Objet du document

Le présent document constitue le dossier d'exploitation de l'application **OCPizza**

Ce document est un support pour l'application **OCPizza**. Avec les procédures de démarrage, d'arrêt et de surveillance de l'application.

2.2. Références

Pour de plus amples informations, se référer :

1. **PROJET_08_Dossier_fonctionnelle** : Dossier de conception fonctionnelle de l'application
2. **PROJET_08_Dossier_conception_technique** : Dossier de conception technique de l'application
3. **PROJET_08_Proces_verbal** : Procès-verbal de livraison finale

Les documents sont consultables sur Github sur PDOCPizza_fourny_angelique :

https://github.com/fangel19/OC_Pizza_P03_Complet_Tech.git

OCPizza

3.PRÉ-REQUIS

3.1.Système

L'application web OCPizza est hébergée sur  HEROKU (ocpizza.fr)

Technologie utilisé pour la base de donnée :

-  python™ : 3.9.3 et son Framework  : 3.1

Technologie utilisé pour le site web :

-  ANGULAR : 11

3.2.Bases de données

Le SGBD (Système de Gestion de Base de Données) utilisé est  PostgreSQL version 11

Cette base sera gérée par et  mongoDB hébergée sur  HEROKU

3.3.Web-services

Les web services suivants doivent être accessibles et à jour :

-  Firebase (outil, API, Analytics)
-  stripe (paiement en ligne)

3.4.Contenu de l'application web OCPizza

L'application OCPizza est sous la forme d'une archive ZIP contenant les répertoires :

- **Interface client** : contient les fichiers liés au package « Interface client » de l'application.
- **Interface pro** : contient les fichiers liés au package « Interface utilisateurs du restaurant » de l'application.
- **Images** : les fichiers images de l'application.
- **Fichier SQL** : sauvegarde des différents fichiers SQL de base.
- **Doc projet** : les différentes documentations, présentations générales et techniques du projet

OCPizza

4. PROCÉDURE DE DÉPLOIEMENT

4.1. Déploiement de l'application web OCPizza

4.1.1. Déploiement



Afin de mettre en place la solution, il y a une procédure de déploiement à respecter. L'hébergeur du serveur étant Heroku, un simple push (via la commande GIT) suffit.

Pour cela, il faut se rendre à la racine du projet livré et d'effectuer les commandes suivantes :

- « git add . »
- « git commit -m "first deploy" »
- « git push »

Pour plus d'informations sur les méthodes de déploiement, consulter le lien suivant :

<https://devcenter.heroku.com>

4.1.2. Vérifications

Afin de vérifier le bon déploiement de l'application, il suffira de :

Tester manuellement

Faire un push (Staging heroku)

Pour plus d'info suite au Staging vous pouvez aussi consulté les log

Exemple :

Un Dump de notre Base de Données :

https://github.com/fangel19/OC_Pizza_P02_Spe_Techniques/blob/main/PGESTpizza_fourny_angélique/SQL/dump.sql



5. PROCÉDURE DE DÉMARRAGE / ARRÊT

Le démarrage de l'application web se fera lors des prochains push

Avec  **HEROKU** il est possible de mettre en mode maintenance ce qui permet de désactiver l'accès à l'application

<https://devcenter.heroku.com/articles/maintenance-mode>

Mode de maintenance

 Anglais — [日本語語に切り替](#)

 Dernière mise à jour le 09 mars 2020

☰ Table des matières

- Utilisation
- Mode de maintenance et bancs d'essai
- Personnalisation de votre page de maintenance

Si vous devez désactiver temporairement l'accès à votre application Heroku (par exemple, pour effectuer une migration importante), vous pouvez activer le mode de maintenance intégré d'Heroku. En mode maintenance, votre application sert une **page de maintenance** statique à tous les visiteurs.

Lorsque le mode maintenance est actif, le code d'erreur **H80** est renvoyé dans vos journaux.



Les tâches de planification peuvent toujours s'exécuter pendant que votre application est en mode maintenance. Soyez attentif si vous avez des tâches configurées pour s'exécuter.



6. SUPERVISION/MONITORING

La solution mise en place pour est Firebase Performance Monitoring

Pour les avantages les plus notables de l'application de Firebase performance monitoring que vous devez connaître :

- Des applications efficaces et réactives
- Il est plus facile d'obtenir un contexte de performance avec des traces
- Suivre le comportement du réseau
- Comprendre l'origine du problème

OCPizza

7. PROCÉDURE DE SAUVEGARDE ET RESTAURATION

7.1. Sauvegarde

Une sauvegarde de notre base de données (un dump) à été effectué lors de la création de notre base sur

 PostgreSQL . Nous pourrons faire une sauvegarde sur  HEROKU automatique, restauration...

7.1.1.1. Création d'une sauvegarde

By default, `pg:backups` operates against your primary database, identified by the `DATABASE_URL` config var:

```
$ heroku pg:backups:capture --app sushi
Hit Ctrl-C at any time to stop watching progress; the backup will
continue running. Stop a running backup with heroku pg:backups:cancel.

HEROKU_POSTGRESQL_BLACK (DATABASE_URL) ----backup--> b251

Running... done
```

Si vous avez plusieurs bases de données sur votre application, vous pouvez choisir celle à sauvegarder en spécifiant le nom de la base de données :

```
$ heroku pg:backups:capture HEROKU_POSTGRESQL_PINK
Hit Ctrl-C at any time to stop watching progress; the backup will
continue running. Stop a running backup with heroku pg:backups:cancel.

HEROKU_POSTGRESQL_PINK ----backup--> b252

Running... done
```

You can use the flag `--verbose` to see logs as your backup progresses. If you need to stop a backup for any reason, use the `cancel` command:

```
$ heroku pg:backups:cancel
Canceled backup b252
```

OCPizza

7.1.1.2. Planification d'une sauvegarde

En plus des sauvegardes déclenchées manuellement, vous pouvez planifier des sauvegardes automatiques régulières. Ceux-ci s'exécuteront quotidiennement sur la base de données spécifiée.

```
$ heroku pg:backups:schedule DATABASE_URL --at '02:00 America/Los_Angeles' --app sushi
```

The `--at` option uses a 24 hour clock to indicate the hour of the day that you want the backup to be taken. The `--at` option also accepts a timezone in either the full TZ format (America/Los_Angeles) or the abbreviation (PST) but we recommend you use the [full TZ format](#).

7.2. Restauration d'une sauvegarde

Pour restaurer une sauvegarde, utilisez la commande `restore` :

```
$ heroku pg:backups:restore b101 DATABASE_URL --app sushi
```

This will restore backup id `b101` to the specified database URL in the app `sushi`. Note: you can omit the backup id and the target database to restore the latest backup to `DATABASE_URL`, otherwise both backup id and target database must be provided.

Vous pouvez également restaurer à partir d'une sauvegarde sur une autre application (de l'application `sushi` à l'application `sushi-staging`) :

```
$ heroku pg:backups:restore sushi::b101 DATABASE_URL --app sushi-staging
```

Ou à partir d'une URL accessible au public :

```
$ heroku pg:backups:restore 'https://s3.amazonaws.com/me/items/mydb.dump' DATABASE_URL
```

Possibilité avec Django-Cron : Permet de créer des tâches « Cron » pouvant s'exécuter à intermédiaire régulier, comme une mise à jour de la base de données vis-à-vis d'une source externe.

OCPizza

8.GLOSSAIRE

API	En informatique, une interface de programmation d'application ou interface de programmation applicative est un ensemble normalisé de classes, de méthodes, de fonctions et de constantes qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.
CRON	CRON est un programme qui permet aux utilisateurs des systèmes Unix d'exécuter automatiquement des scripts, des commandes ou des logiciels à une date et une heure spécifiée à l'avance, ou selon un cycle défini à l'avance.