



{{OC Pizza}}

{{OC Pizza-Manager}}

Dossier d'exploitation

Version {{1.0.0}}

{{Entreprise}}

<<https://openclassrooms.com>>

<10 QUAI DE LA CHARENTE 75019 PARIS> – <[01 80 88 80 30](tel:0180888030)> –
<openclassrooms@gmail.com>

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



Auteur

{{Mr Cottenceau Thomas}}

{{Développeur Full Stack Python}}

{{Entreprise}}

<<https://openclassrooms.com>>

<10 QUAI DE LA CHARENTE 75019 PARIS> – <[01 80 88 80 30](tel:0180888030)> –
<openclassrooms@gmail.com>

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

TABLE DES MATIÈRES

1 Versions	4
2 Introduction	5
2.1 - Objet du document	5
2.2 - Références	5
3 Pré-requis	6
3.1 Système	6
3.1.1 Serveur de Base de données.....	6
3.1.2 Serveur Web	6
4 Procédure de déploiement	7
4.1 Déploiement de l'application web	7
4.1.1 Protocole	7
4.1.2 Variables d'environnement.....	7
4.1.3 Le premier push.....	8
4.1.4 DataSources.....	8
4.1.5 Vérifications.....	8
5 Procédure de démarrage / arrêt	9
5.1 Application web	9
6 Procédure de mise à jour	10
6.1 Base de données.....	10
6.2 Application web	10
7 Supervision/Monitoring	11
7.1 Supervision de l'application web	11
8 Procédure de sauvegarde et restauration.....	12
8.1 Protocole.....	12

1 - VERSIONS

Auteur	Date	Description	Version
Cottenceau Thomas	02/03/2021	Création du document	1.0.0

2 - INTRODUCTION

2.1 - Objet du document

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

Ce document a pour vocation d'être le protocole du déploiement, modification et sauvegarde de l'application.

2.2 - Références

Pour de plus amples informations, se référer :

1 DCT - OC Pizza-Manager : Dossier de conception fonctionnelle

2 DCT - OC Pizza-Manager : Dossier de conception technique

3 - PRÉ-REQUIS

3.1 Système

3.1.1 Serveur de Base de données

Le serveur de la base de données PostgreSQL sera géré par le serveur web Heroku qui fournit une clé donnant accès à la base de données par le biais d'une variable d'environnement.

3.1.1.1 Caractéristiques techniques

Le serveur de base de données hébergeant le schéma 'ocpizza_manager_db' ce fait grâce à une option 'Add-ons' disponible sur Heroku donnant accès à une base de données PostgreSQL pour 1.7 Go de mémoire à 200 dollars par mois qui pour l'instant suffira amplement.

3.1.2 Serveur Web

L'application sera hébergée sur Heroku.

3.1.2.1 Caractéristiques techniques d'Heroku, les 'Dynos'

Le code de l'application est exécuté sur la plate-forme dans des structures appelées 'dynos'. Ce sont des conteneurs d'exécution gérés qui reposent sur un système d'exploitation Linux. Ces conteneurs exécutent les processus qui permettent d'exécuter le code de l'application. Ce sont de simples mécanismes qui permettent d'isoler les processus d'exécution les uns des autres. Ils séparent votre code et votre configuration des autres. Ainsi, votre application conserve ses propres caractéristiques.

4 - PROCÉDURE DE DÉPLOIEMENT

4.1 Déploiement de l'application web

4.1.1 Protocole

Le déploiement sur Heroku se fait simplement, néanmoins certaines étapes sont nécessaires avant de déployer quoi que ce soit. La première des choses est bien entendu la création d'un compte. Ensuite il faut installer l'outil 'Heroku Cli' afin de pouvoir exécuter les commandes propres au déploiement en le téléchargeant directement sur google. On le trouve à la rubrique 'Download and install' en prenant la version compatible avec votre Os. Une fois dans le terminale au repertoire du projet sur votre machine connectez vous en tapant 'heroku login'. Une page s'ouvre et vous invite à confirmer votre connexion. Une fois fait il est temps de créer l'application en question en tapant la commande 'heroku apps:create le_nom_que_vous_lui_donnez' (ce nom doit être unique sur Heroku afin de ne pas créer de confusion) cette opération crée non seulement l'application mais aussi un remote lié au repo Github de l'application afin de pouvoir bénéficier des commandes git sur Heroku.

4.1.2 Variables d'environnement

Le moment est venu de créer nos variables d'environnement nécessaires au fonctionnement de l'application avec la commande suivante 'heroku config:set le_nom_de_la_variable=sa_valeur'.

Ainsi nous créons la clé secrète de l'application comme suit:

```
heroku config:set OCPIZZA_MANAGER_KEY=4jdsksnkgndklnd59sknsk6bv6fv
```

La variable définissant l'environnement de production:

```
heroku config:set ENV=PRODUCTION
```

Et enfin la variable contenant le chemin donnant accès aux settings de l'application:

```
heroku config:set DJANGO_SETTINGS_MODULE=ocpizza_manager.settings
```

Pour ce qui est de la variable d'environnement de la base de données allez sur votre compte Heroku et selectionnez votre application. Ensuite ajoutez un 'Add-ons' directement dans 'Ressources' puis entrez dans le formulaire 'Add-ons' le nom 'Heroku Postgres' et ajoutez le. Vous aurez un point d'accès à une base de données sous forme d'une variable consultable dans les 'Settings' ou vous y verrez les variables créées précédemment.



4.1.3 Le premier push

Pour deployer il vous suffit de procéder de la même manière que vous le feriez sur Github. Tout d'abord il est important de configurer le Procfile nécessaire au lancement de l'application. Ouvrez le et écrivez y la ligne suivante:

```
web: gunicorn ocpizza_manager.wsgi
```

Ceci précise à Heroku que nous utilisons le server gunicorn pour lancer l'application sur Heroku.

On reviens au terminal et on réalise notre premier push sur Heroku comme suit:

```
git add *
```

```
git commit -m "déploiement heroku"
```

```
git push Heroku main (ou master selon votre branche principale)
```

4.1.4 DataSources

L'accès à la base de données sur Heroku se fait dans le terminale en tapant les commandes suivantes:

```
Heroku run bash
```

Maintenant que vous êtes connecté à la base de données sur Heroku il ne reste plus qu'à exécuter les commandes habituelles comme 'python manage.py migrate' pour créer les migrations par exemple.

4.1.5 Vérifications

Afin de vérifier le bon déploiement de l'application, revenez à votre compte Heroku sélectionnez votre application et cliquez sur l'onglet 'Overview', dans la rubrique 'Latest activity' vous devriez voir la mention 'Deployed'. Si elle n'apparaît pas c'est qu'une erreur c'est produite et à empêchée le déploiement. Pour mieux comprendre ce qu'il s'est produit il faut consulter les Logs dans l'onglet 'More' et ensuite 'View Logs'. Sélectionnez 'All process' afin de voir toutes les opérations. Vous pouvez aussi consulter ceci dans votre terminal éventuellement.



5 PROCÉDURE DE DÉMARRAGE / ARRÊT

5.1 Application web

Lors d'une migration en base de données importante ou pour une autre raison il est possible de mettre en pause l'application. Pour ce faire taper la commande suivante dans votre terminal:

```
heroku maintenance:on Maintenance mode enabled
```

Une fois les migrations ou modifications effectuées tapez :

```
heroku maintenance:off Maintenance mode disabled
```

Pour supprimer l'application du compte Heroku. **Attention Ceci est irréversible !** Accédez à votre tableau de bord Heroku. Sélectionnez l'application que vous souhaitez supprimer. Faites défiler vers le bas de la page des paramètres pour cette application. Appuyez sur le bouton rouge 'Supprimer l'application'.

6 PROCÉDURE DE MISE À JOUR

6.1 Base de données

Tout d'abord si vos modifications sont assez conséquentes vous pouvez mettre en pause l'application avec la méthode montrée précédemment de mise pause. Afin de mettre la base de données à jour il suffit de faire un push après modification comme expliqué précédemment et ensuite à l'aide du bash Heroku de migrer comme suit:

```
heroku run bash (pour activer l'invite de commande de Heroku)
```

```
heroku manage.py migrate
```

6.2 Application web

L'application est mise à jour dès qu'un push sur Heroku est effectué (voir procédure expliquée précédemment).

7 SUPERVISION/MONITORING

7.1 Supervision de l'application web

Pour s'assurer que l'application web fonctionne bien des tests fonctionnels ont été mis en place. Sinon il reste tout simplement la consultation de l'application sur le web qui est aussi un indicateur.

8 PROCÉDURE DE SAUVEGARDE ET RESTAURATION

8.1 Protocole

La sauvegarde de l'application se fait sur le cloud Github. Chaque développeur salarié de la société Openclassrooms chargé de travailler sur l'application importe le projet en local sur sa machine par commande 'git clone <https://urlducheminduprojet.com>'. Il crée ensuite une nouvelle branche portant un nom faisant référence à la nouvelle fonctionnalité demandée en tapant 'git branch lenomdelabranche' se met dessus en tapant 'git checkout lenomdelabranche'. Chaque nouvelle fonctionnalité commence par l'écriture d'un test en suivant le principe du TDD. Avant d'être enfin inclus sur la branche master. Vient ensuite un contrôle d'éventuels conflits de versions entre son apport et celui de ses collègues en faisant un 'git fetch'. Une fois ceci prit en considération, la et seulement la, la procédure de déploiement est lancée. Si une Nouvelle fonctionnalité est demandée on recrée une nouvelle branche et ainsi de suite.