

OC Pizza

Solution de gestion du groupe de restaurants OC_Pizza

Dossier d'exploitation

Version 1

Auteur

Pierre Sempéré
Développeur

TABLE DES MATIERES

0 -Versions	3
1 -Introduction	3
1.1.1 -Objet du document.....	3
1.1.2 -Références	3
Pré-requis	4
1.2 -Système	4
1.2.1 -Serveur de Base de données.....	4
1.2.1.1 -Caractéristiques techniques	4
1.2.2 -Serveur Web.....	4
1.2.2.1 -Caractéristiques techniques	4
1.2.3 -Serveur de Fichiers.....	4
1.3 -Bases de données	5
1.4 -Web-services	5
2 -Procédure de déploiement.....	6
2.1 -Déploiement des Batches	6
2.1.1 -Fichiers	6
2.1.2 -Configuration.....	7
2.1.3 -Ressources	8
2.1.4 -Vérifications.....	8
2.2 -Déploiement de l'Application Web.....	8
2.2.1 -Fichiers	8
2.2.2 -Environnement de l'application web	9
2.2.2.1 -Variables d'environnement.....	9
2.2.3 -Répertoire de configuration applicatif	10
2.2.4 -Vérifications.....	10
3 -Procédure de démarrage / arrêt.....	11
3.1 -Base de données	11
3.2 -Batches	11
3.3 -Application web	11
4 -Procédure de mise à jour.....	12
4.1 -Base de données	12
4.2 -Application web	12
5 -Supervision/Monitoring.....	13
5.1 -Supervision de l'application web	13
5.2 -Supervision de l'infrastructure	13
6 -Procédure de sauvegarde et restauration	14
7 -Glossaire	15

0. VERSIONS

Auteur	Date	Description	Version
Pierre Sempéré	26/08/2021	Création du document, éditions majeures.	0.1
Pierre Sempéré	04/09/2021	Éditions majeures	0.2
Pierre Sempéré	06/09/2021	Mise en page et correction des erreurs.	0.3
Pierre Sempéré	21/12/2021	Corrections et révisions finales	1.0

1.1 - Introduction

1.1.1 - Objet du document

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

L'objectif du document est de documenter la marche à suivre pour le bon déploiement de ladite application sur un serveur et la mise en ligne de ce dernier.

1.1.2 - Références

Pour de plus amples informations, se référer :

1. **OpenClassrooms_DA_Python_V1_Projet10**
2. **Dossier de conception technique**
3. **Dossier de conception fonctionnelle**

1- PRÉ-REQUIS

1.2 - Système

1.2.1 - *Serveur de Base de données*

Serveur Ubuntu 20.04 Focal Fossa et PostgreSQL 13

1.2.1.1 - *Caractéristiques techniques*

Ubuntu 20.04 Focal Fossa (LTS)

Processor: 2 X86 64b

RAM: 2GB

Memory: 100GB SSD NVMe

Bandwidth: 200 Mbps

1.2.2 - *Serveur Web*

VPS Scaleway : Serveur Ubuntu 20.04 Focal Fossa

1.2.2.1 - *Caractéristiques techniques*

NginX Version 1.21.1

Processor: 4 X86 64b

RAM: 16GB

Memory: 200GB SSD NVMe

Bandwidth: 500 Mbps

1.2.3 - *Serveur de Fichiers*

Scaleway object storage a capacite de 250 Go (extensible).

1.3 - Bases de données

Les bases de données et schémas suivants doivent être accessibles et à jour :

- **PostgreSQL:** version 13

1.4 - Web-services

Les web services suivants doivent être accessibles et à jour :

- **NginX:** version 1.21.1
- **GUnicorn:** version 20.1.0
- **Sentry:** version 1.3.1
- **NewRelic:** version 1.19.4 (infrastructure agent)
- **TravisCI**

2 - PROCÉDURE DE DÉPLOIEMENT

2.1 - Déploiement des batches

2.1.1 - Fichiers

Les fichiers de l'application `oc_pizza_project` sont construits sous la forme d'une archive `tar.gz` **delivery.tar** contenant les répertoires et fichiers de configuration suivants :

- **oc_pizza_project** : le répertoire contenant le code source du projet Django et toutes ses applications.
- **conf_sql** : le fichier de dump sql permettant d'initialiser la base de donnée.
- **conf_scripts** : des fichiers shell bash à exécuter pour lancer les installations des dépendances.
- **oc_pizza_project/requirements.txt** : fichier texte contenant les modules Python nécessaires à l'application.
- **conf_sqlp/db_dump.sql** : fichier de configuration de la base de données.
- **oc_pizza_project/oc_pizza/settings/production.py** : fichier de réglages propres à l'environnement de production.
- **oc_pizza_project/oc_pizza/settings/sentry_setup.py** : fichier de réglages permettant la configuration de Sentry pour les logs de l'application

2.1.2 - Configuration

Extraire l'archive **delivery.tar** dans le répertoire :

/home

Positionner les droits d'exécution sur les scripts SH de lancement des batches avec la commande `sudo chmod 101 /home/oc_pizza_project/*`.

1. Installation des dependances logicielles:

Executer successivement:

- `cd oc_pizza_project`
- `sudo source conf_scripts/install_dependencies.sh`

Le fichier `install_dependencies.sh` : Ce fichier regroupe toutes les commandes nécessaires à l'installation des logiciels évoqués dans les points **2.4 et 2.5** ainsi que la dernière version de Python et pip.

2. Installation des packages Python:

Executer:

- `pip install -r requirements.txt`

Cette action permet d'installer tous les modules de Python nécessaires au bon fonctionnement de l'application

3. Initialisation de la base de données :

Exécuter successivement :

- `cd oc_pizza`
- `python3 manage.py migrate`
- `python3 manage.py makemigrations`
- `cd /home/ocpizzaproject`
- `psql oc_pizza_db < conf_sql/dump`

Cette série de commandes permettra dans un premier temps d'initialiser la base de données conformément aux modèles définis dans les applications Django, le fichier `dump.sql` remplira les tables de la base de données de lignes prédéfinies pour l'utilisation de l'application par le client.

2.1.3 - Ressources

Les ressources sont les fichiers static et médias qu'on appariera à l'application Django au moyen de la commande :

```
python3 oc_pizza/manage.py collectstatic
```

2.1.4 - Vérifications

Exécuter la commande :

```
Python3 oc_pizza/manage.py tests
```

Les tests unitaires et selenium devraient s'exécuter et ne reporter aucune erreur si les démarches précédentes ont été effectuées correctement.

2.2 - Déploiement de l'Application Web

2.2.1 - Fichiers

L'archive fournie au client contient le répertoire oc_pizza_project qui lui-même contient le code source de l'application web.

2.2.2 - Environnement de l'application web

2.2.2.1 - Variables d'environnement

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

Nom	Obligatoire	Description
DJANGO_SECRET_KEY	Oui	Permet la génération d'éléments impératifs à la sécurité de l'application
DJANGO_SETTINGS_MODULE	Oui	Permet au service Web de localiser convenablement les réglages de l'application au démarrage
DATABASE_URL	Non	Alternative à la présence des identifiants PostgreSQL directement dans le code

Pour créer ces variables, exécuter les commandes :

```
sudo export DJANGO_SECRET_KEY="<voir*>"
```

```
sudo export DATABASE_URL="<voir*>"
```

*Variables de sécurité, leur contenu ne doit être connu que du client.

Pour DJANGO_SETTINGS_MODULE : créer la variable d'environnement
DJANGO_SETTINGS_MODULE=oc_pizza.settings.production

2.2.3 - Répertoire de configuration applicatif

Le répertoire de configuration applicatif se trouve dans /home/oc_pizza_project/oc_pizza, il se nomme settings et contient les fichiers suivants :

- __init__.py : contient les réglages communs aux environnements de production et de développement de l'application, les données contenues ne sont pas sensibles.
- production.py : importe les valeurs de __init__.py et sentry_setup.py, ce fichier est privé et n'est pas visionné par les développeurs une fois la livraison effectuée.
- sentry_setup.py : permet de paramétrer Sentry (le service utilisé pour le monitoring) pour répondre au mieux aux besoins du client.
- travis.py : utilisé par TravisCI par l'équipe de développement en tant qu'outil d'intégration continue.

2.2.4 - Vérifications

Afin de vérifier le bon déploiement de l'application, se rendre à l'adresse du serveur et vérifier que l'application est accessible et fonctionnelle.

3 - PROCÉDURE DE DÉMARRAGE / ARRÊT

3.1 - Base de données

Le service PostgreSQL est actif dès son installation et ne nécessite pas de procédure de lancement spécifique. Les commandes :

`sudo service postgresql stop` : permet d'arrêter le service

`sudo service postgresql start` : permet de relancer le service

3.2 - Batches

Batch inclus permettant de sauvegarder et restaurer la BDD automatiquement

Une tâche Cron doit être mise en place de cette façon pour effectuer une sauvegarde tous les jours à minuit :

1. Ouvrir le Crontab avec la commande : `crontab -e`
2. Entrer après les commentaires signalés par des # la commande :
`0 0 * * * pg_dump oc_pizza > dumpsql`
3. Sauvegarder les changements et quitter crontab

3.3 - Application web

Supervisor assure l'ininteruption du service, ainsi que son démarrage et arrêt manuel :

Les commandes importantes sont les suivantes :

- `sudo supervisorctl reread` : permet d'effectuer une relecture du fichier de configuration <confname>
- `sudo supervisorctl update` : permet de mettre à jour le service, à utiliser après `reread`
- `sudo supervisorctl restart` : permet de redémarrer les services assurés par supervisor, à utiliser après `update` pour que les mises à jour du fichier <confname> soient appliquées.
- `sudo supervisorctl` : permet de voir si le service `oc_pizza_gunicorn` est actif.
- `sudo supervisor stop oc_pizza_gunicorn` : permet d'arrêter le service et donc l'application web
- `sudo supervisor start oc_pizza_gunicorn` : permet de lancer manuellement le service et donc l'application web.

4 - PROCEDURE DE MISE A JOUR

4.1 - Base de données

La base de données sera mise à jour lors de l'utilisation de l'application, lors de son utilisation par les clients et le personnel d'OC PIZZA.

Si des changements sont nécessaires, il sera fourni un dump.sql contenant les changements demandés par le client. Il devra exécuter la commande

```
- psql oc_pizza_db < 'nom_du_fichier_sql'
```

4.2 - Application web

Les éventuelles mises à jour de l'application web s'effectueront au par le même moyen que la livraison. Une archive sera fournie au client, ses instructions d'installation dépendront du type de modification apportée. Certaines modifications pourront nécessiter le redémarrage de certains services, ainsi, il sera fourni au client un document texte contenant la marche à suivre en fonction de la mise à jour.

5 - SUPERVISION/MONITORING

5.1 - Supervision de l'application web

Sentry sera utilisé comme package de monitoring de l'application Django. Différents niveaux de retours peuvent être configurés pour que les retours de Sentry soient le plus pertinents possibles.

5.2 - Supervision de l'infrastructure

New Relic sera utilisé pour le monitoring de l'infrastructure. Cela nous permettra de consulter en permanence les ressources utilisées, les logs de l'infrastructure et d'autres informations pertinentes comme l'état de la bande passante du serveur.

6 - PROCÉDURE DE SAUVEGARDE ET RESTAURATION

L'application est développée en utilisant git, un problème majeur, que ce soit dans l'environnement de développement ou de production, n'empêcherait pas la récupération du code et son redéploiement.

Une tâche Cron sera configurée afin d'effectuer une sauvegarde journalière de la base de données. Ce dump sera ensuite versionné en utilisant Git. En cas de problème majeur affectant la base de données, ce dump pourrait être utilisé pour restaurer la base de données.

Scaleway offre la possibilité de faire une sauvegarde de l'instance pour que la configuration de l'utilisateur puisse être récupérée en cas de problème majeur.

Ces trois mesures de sécurité permettraient :

- Redéployer le code et toutes les données contenues dans la base de données si l'infrastructure pose un problème
- Restaurer le contenu de la base de données à une version antérieure si celle-ci se retrouvait corrompue ;
- Récupérer le code de l'application si un problème majeur y contraignait
- Redéployer l'infrastructure sur une autre instance Scaleway s'il y avait une interruption de service.

7 - GLOSSAIRE

Batch	Le <i>traitement en batch</i> consiste à exécuter des action répétitives contenant des volumes importants de données. La méthode par lots permet aux utilisateurs de traiter des données lorsque des ressources informatiques sont disponibles, avec peu ou pas d'intervention de leur part. Source : https://www.talend.com/fr/resources/batch-processing/
Cron	Utilitaire permettant l'automatisation de tâches prédéfinies et l'exécution de ces tâches à des instants prédéfinis par l'utilisateur.
Monitoring	Processus permettant de surveiller l'activité d'un processus, ici on surveille l'activité de l'application (Sentry) et de l'infrastructure (New Relic)
Log	Fichier stockant des données horodatées d'évènements divers.