|  |
| --- |
| Résultat de recherche d'images pour "dessin pizza"OC PIZZA  Système de gestion pizzeria  Dossier d'exploitation  Version 1.0 |
| **Auteur**  Sébastien Fantinati  Analyste programmeur |

Table des matières

1 - Introduction 5

1.1 - Objet du document 5

1.2 - Références 5

2 - Pré-requis 6

2.1 - Système 6

2.1.1 - Serveur de Base de données 6

2.1.1.1 - Caractéristiques techniques 6

2.1.2 - Serveur Web 6

2.1.2.1 - Caractéristiques techniques 6

2.1.3 - Serveur de Batches 6

2.1.4 - Serveur de Fichiers 6

2.2 - Bases de données 7

2.3 - Web-services 7

2.4 - Autres Ressources 7

3 - Procédure de déploiement 8

3.1 - Déploiement de l'Application Web 8

3.1.1 - Composition de l’application web 8

3.1.2 - Environnement de l’application web 8

3.1.2.1 - Variables d’environnement 8

3.1.3 - Répertoire de configuration 8

3.1.4 - Déploiement 9

3.1.5 - Ressources 9

3.1.6 - Vérifications 9

4 - Procédure de démarrage / arrêt 10

4.1 - Base de données 10

4.2 - Application web 10

5 - Procédure de mise à jour 11

5.1 - Base de données 11

5.2 - Application web 11

6 - Supervision/Monitoring 12

6.1 - Supervision de l’application web 12

7 - sauvegarde et restauration 13

7.1 - Sauvegarde base de données 13

7.2 - Restauration base de données 13

8 - Glossaire 14

Versions

|  |  |  |  |
| --- | --- | --- | --- |
| Auteur | Date | Description | Version |
| SF | 13/10/2019 | Création du document | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Objet du document

Le présent document constitue le dossier d’exploitation d’un système de gestion de gestion de pizzeria.

L’objectif du document est de fournir à l’équipe technique de la société OC PIZZA les informations essentielles pour une bonne utilisation de l’application, ainsi que les instructions à suivre pour le déploiement et la maintenance de celle-ci.

## Références

Pour de plus amples informations, se référer :

1. **P9 - DCT - 1.1:** Dossier de conception technique de l'application
2. **P9 - DCF - 1.1 :** Dossier de conception fonctionnelle de l’application.

# Pré-requis

## Système

L’application est hébergé sur un serveur dédié.  
L’application est liée au nom de domaine : « https://oc-pizza.fr»

### Serveur de Base de données

Le SGBD utilisé sera Mysql est sera préalablement installé lors de la préparations du serveur.

Un utilisateur avec les plein pouvoirs sur la base de données sera être créé.

#### Caractéristiques techniques

Le déploiement de la solution sera hébergé sur serveur dédié dimensionné pour recevoir le système et qui sera capable de gérer les connexions multiples.

Afin d’assurer la pérennité du système et de répondre dans les meilleurs délais, le serveur sera équipé de disque SSD de 256Go.

Afin d’éviter toute perte de données les SSD seront paramétrés en RAD1 afin d’avoir une tolérance de panne de 1 pour 1. Toutes les données du premier disque seront copiées sur le second.

Le serveur sera équipé d’un processeur XEON E2640 et de 16 Go de RAM.

### Serveur Web

Le serveur web sera préparé avec les applications Nginx, Django et Gunicorn dans leur dernières version.

#### Caractéristiques techniques

Idem que le serveur de base de données

### Serveur de Batches

Néant

### Serveur de Fichiers

Néant

## Bases de données

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

* **Mysql :** version 8.0.16

## Web-services

Les web services suivants doivent être accessibles et à jour :

* **Nginx :** version 1.17.3
* **Django:** version 2.2.6
* **Gunicorn:** version 19.9.0

## Autres Ressources

Python 3.7

Le système d’exploitation installé sera Ubuntu server 19.04.

# Procédure de déploiement

## Déploiement de l'Application Web

### Composition de l’application web

L’application est construite sous la forme d'une archive ZIP contenant les répertoires suivants: 

* **OCPIZZA** : Contient les fichiers de configuration de l’application et de Django. 
* **Procfile** : Contient les instructions pour démarrer l’application 
* **requirements.txt** : contient les librairies nécessaires pour que l’application fonctionne. 
* **Static** : Contient les fichiers .CSS et .JS de l’application. 
* **Templates** : Contient les fichiers HTML de l’application. 
* **Ventes** : Contient les fichiers liés au package « Ventes » de l’application. 
* **Production** : Contient les fichiers liés au package « Production » de l’application. 
* **Docs** : Contient la documentation de l’application.

Le fichier Procfile contient les instructions exécutées par le serveur pour démarrer l’application.   
Le contenu de ce fichier doit a minima être le suivant pour que l’application fonctionne :

**web: gunicorn ocpizza.run :app**

### Environnement de l’application web

#### Variables d’environnement

Pour passer Django en productions les variables d'environnement suivante sont nécessaire :

|  |  |  |
| --- | --- | --- |
| VARIABLE | VALEUR | commandes |
| ENV | PRODUCTION | export ENV=PRODUCTION |
| SECRET\_KEY | g5wdqq6$^o!k@o5o61ngjqu-03i9e#i(zg(!)6\*t)4vn#pap+# | export SECRET\_KEY=g5wdqq6$^o!k@o5o61ngjqu-03i9e#i(zg(!)6\*t)4vn#pap+# |

### Répertoire de configuration

Voici les différents fichiers de configuration : 

* **Procfile** : fichier contenant les commandes utilisées par le serveur pour le déploiement de l’application
* **requirements.txt** : fichier contenant la liste des librairies à installer sur le serveur pour que l’application fonctionne correctement. 
* **OCPIZZA/pizzapp/settings.py** : fichier de configuration l’application Django.

### Déploiement

Le déploiement peut être effectué en suivant les étapes ci-dessous:

1. Cloner depuis notre dépôt Git l’archive de votre application.
2. Editer le fichiers settings.py et modifier les paramètres de connexion à la base de données.
3. Lancement des migrations avec la commande ./***manage.py migrate***
4. Lancement de de la création des fichiers statics avec la commande ./***manage.py collectstatic (répondre yes à la question)***
5. Créer un super utilisateur pour administrer l’application avec la commande ***./manage.py createsuperuser***
6. Configurer un vhost sur Nginx comme suis:

erver {

listen 80;

listen [::]:80;

server\_name <nom de domaine>;

root /home/user/pur\_beurre/;

index index.html;

location /static {

alias /home/user/pur\_beurre/staticfiles;

}

location / {

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_redirect off;

proxy\_pass http://127.0.0.1:8000;

}

}

1. Lancer le serveur avec la commande ***gunicorn pur\_beurre.wsgi:application***

### Ressources

Néant

### Vérifications

Vérifier que le service Nginx fonctionne avec la commande ***systemctl status nginx.*** Doit retourner: active (running).

Vérifier que le service Mysql fonctionne avec la commande ***systemctl status mysql.*** Doit retourner: active (running).

# Procédure de démarrage / arrêt

L’arrêt du système s’effectue en ligne de commande.  
Le démarrage s’effectue automatiquement en même temps de le démarrage serveur.

## Base de données

Pour arrêter la base de données proprement exécuter la commande ***sudo systemctl stop mysql***

## Application web

Pour arrêter le serveur web proprement exécuter la commande ***sudo systemctl stop nginx***

Pour stopper django

# Procédure de mise à jour

La mise à jour de l’application nécessite de passer celle-ci en mode maintenance.

## Base de données

Importer les nouveaux models dans le dossiers correspondants puis executer les commandes suivante:

1. ./***manage.py makemigration***
2. ./***manage.py migrate***

## Application web

Remplacer les fichiers existant par les nouveaux

# Supervision/Monitoring

## Supervision de l’application web

Pour superviser le serveur, utiliser l’application sepervisor.

1. Créer un nouveau service dans le dossier /etc/supervisor/conf.d/non\_du\_service.
2. Ajouter les ligne suivante:   
    [program:disquaire-gunicorn]  
    command = /home/user/env/bin/gunicorn pur\_beurre\_project.wsgi:application  
    user = user  
    directory = /home/user/disquaire  
    autostart = true  
    autorestart = true  
    environment =
3. Execter les commandes ***sudo supervisorctl reread, sudo supervisorctl update et sudo supervisorctl status***
4. Faire de même pour les service que vous souhaitez monitorer

# sauvegarde et restauration

## Sauvegarde base de données

Pour effectuer la sauvegarde de la base de données exécuter la commande suivante:

***mysqldump -u <nom\_utilisateur> -p<mot\_de\_ passe> <nom\_de\_la\_base> > OCPIZZA\_<date>.sql.gz***

***(***Replacer les argument <> par vos valeurs ***)***

## Restauration base de données

Pour effectuer la restauration de la base de données exécuter la commande suivante:

***mysql -u <nom\_utilisateur> -p<mot\_de\_ passe> <nom\_de\_la\_base> < OCPIZZA\_<date>.sql.gz***

***(***Replacer les argument <> par vos valeurs ***)***

# Glossaire

|  |  |
| --- | --- |
| **SGBD** | Système de gestion de base de données |
| **DJANGO** | Framework permettant la réalisation d’application web en Python. |
| **NGINX** | Serveur web |
| **CLI** | Command Line |
| **MYSQL** | SGBD |
| **GUNICORN** | Serveur web HTTP WSGI |