OC PIZZA

SYSTEME DE GESTION DE PIZZERIAS

Dossier d'exploitation

Version 1.0

Auteur
Crispin Pigla

A Faire :

- Remplacer manuellement les valeurs entre <>
- Les valeurs entre {{}} sont à renseigner dans les propriétés du document

IT Consulting & Development
S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Xxxx – SIREN 999 999 999 – Code APE : 6202A

Table des matières

1 - Versions

Auteur	Date	Description	Version
Crispin Pigla	12/12/2020	Création du document	1.0

2 - Introduction

2.1 - Objet du document

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

Les éléments du présents dossiers découlent :

- des discussions avec le client
- du cahier des charges

2.2 - Références

Pour de plus amples informations, se référer :

1. Dossier de conception technique de l'application
2. Dossier de conception fonctionnelle de l'application

3 - Pré-requis

3.1 - Système

3.1.1 - Serveur de Base de données

Le serveur de base de données hébergeant la base est postgresql dans sa version version 13.1

3.1.2 - Serveur Web

Le serveur web utilisé pour ce projet est nginx dans sa version 1.18.0

3.1.3 - Serveur de Fichiers

Le serveur hébergeant l'application est un serveur virtuel de l'hébergeur google cloud platform

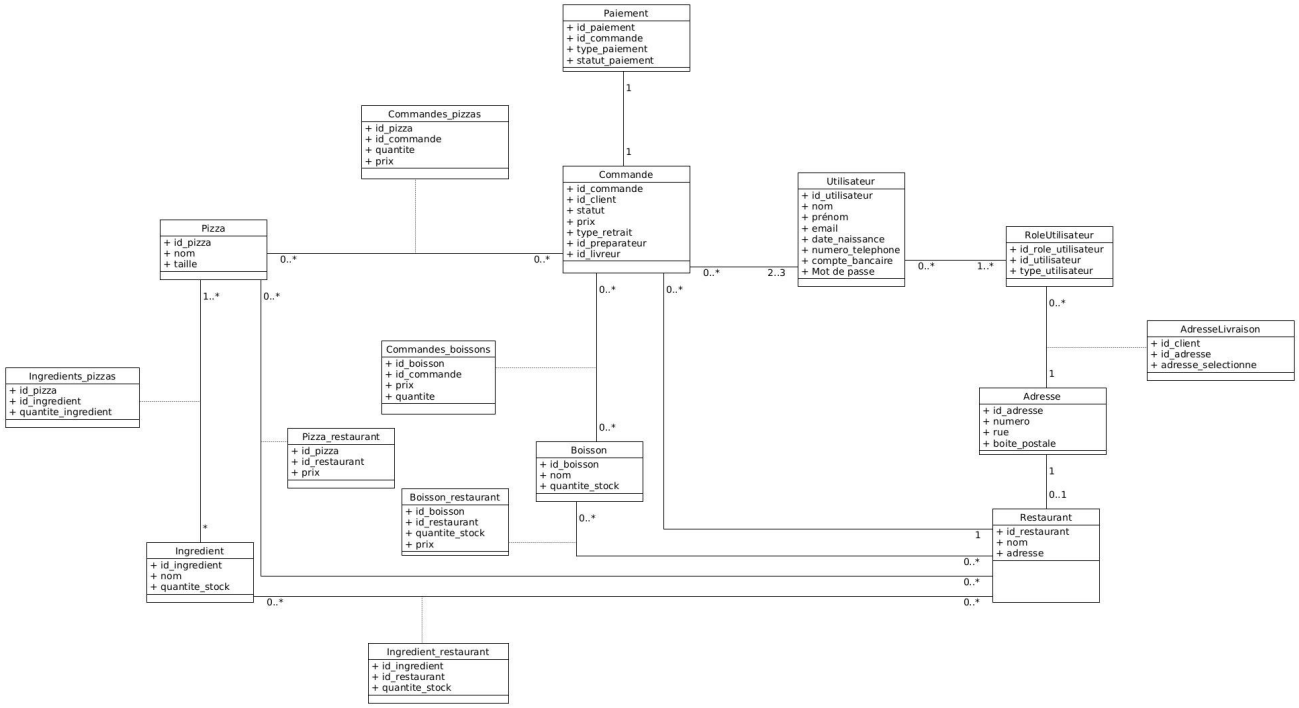
3.1.3.1 - Caractéristiques techniques

- crée le 2 déc. 2020, 15:51:13,
- de type 'e2-micro :2 vCPU, 1 Go de mémoire',
- situé en europe de l'ouest,
- son système d'exploitation est Ubuntu, 20.04 LTS
- l'adresse IP externe est 000.000.000.000

3.2 - Bases de données

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

- **postgresql :** version 13.1
- shémas de la base de données :



3.3 - Web-services

Les web services suivants doivent être accessibles et à jour :

- **nginx :** version 1.18.0
- **supervisor** : version 4.1.0

3.4 - Autres Ressources

- **pipenv**: version 2018.11.26

4 - Procédure de déploiement

4.1 - Déploiement de l'Application Web

4.1.1 - Artefacts

L'application OC pizzerias est construite sous la forme d'un dossier contenant les répertoires :

- **oc_pizza_projet** : contenant les différents modules de l'ensemble du projet
- **oc_pizza_application_authentification** : contenant les différents modules de l'application d'auhtentification
- **oc_pizza_application_client** : contenant les différents modules de l'application réservé au client
- **oc_pizza_application_pizzaiolo**:contenant les différents modules de l'application réservé au pizzaiolo
- **oc_pizza_application_livreur** : contenant les différents modules de l'application réservé au livreur
- **oc_pizza_application_administrateur** : contenant les différents modules de l'application réservé aux administrateur

et les fichiers:

- **manage.py** : permettant de gérer l'application
- **newrelic.ini** : contenant les configurations newrelic de l'application
- **Pipfile, Pipfile.lock, requirements.txt** : contenant les différentes dépendances du projet
- **README.md** : contenant des informations sur le projet
- **supervisor.log, supervisor0.log, supervisor1.log** : contenant les messages et erreurs renvoyés par supervisor

4.1.2 - Téléchargement des sources sur le serveur et installation des packages

- Mettre à jour les logiciels du serveur avec la commande **sudo apt-get update**
- Installer python avec la commande **sudo apt-get install python3**
- Installer postgresql avec la commande **sudo apt install postgresql**
- Installer pipenv avec la commande **pip install pipenv**
- Installer nginx avec la commande **sudo apt install nginx**
- Installer supervisor avec la commande **sudo apt-get install supervisor**
- Creer un dossier qui contiendra l'application ('**dossier0**')
- **Puller** l'application depuis le dépot distant avec la commande '**git pull https://github.com/crispinpigla/substitutor_foods_website.git**'

- Se rendre à l'adresse du projet '**<<adresse vers le dossier dossier0>>**' et installer les dépendences en entrant la commande '**pipenv install**'

4.1.3 - Configuration du serveur

- Créer un fichier de configuration de supervisor ('**substitutor-gunicorn.conf**') à l'adresse '**/etc/supervisor/conf.d**' et insérer le contenu suivant :

```
[program:pure_beurre-gunicorn]
environment = ENV="PRODUCTION",NEW_RELIC_CONFIG_FILE=newrelic.ini
directory = <<adresse vers le dossier dossier0>>
command = pipenv run newrelic-admin run-program gunicorn -w 3 pure_beurre_django.wsgi:application
user = crispinpigla
autostart = true
autorestart = true
logfile= <<adresse vers le dossier dossier0>>/supervisor.log
stderr_logfile= <<adresse vers le dossier dossier0>>/supervisor0.log
stdout_logfile= <<adresse vers le dossier dossier0>>/supervisor1.log
```

ou **<<adresse vers le dossier dossier0>>** est l'adresse vers le dossier '**dossier0**' contenant le projet

- Créer un fichier de configuration nginx ('**purebeurre**') à l'adresse '**/etc/nginx/sites-enabled**' et insérer le contenu suivant :

```
server {
    listen 80;
    server_name <<adresse ip du projet>>;
    root <<adresse vers le dossier dossier0>>;
    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;
    }
}
```

ou **<<adresse vers le dossier dossier0>>** est l'adresse vers le dossier '**dossier0**' contenant le projet et **<<adresse ip du projet>>** est l'adresse ip du serveur du projet

- Ajouter l'adresse ip du projet à la liste **ALLOWED_HOSTS** du fichier '**<<adresse vers le dossier dossier0>>**/pure_beurre_django/settings.py'

4.1.4 - Environnement de l'application web

4.1.4.1 - Variables d'environnement

Le serveur d'application doit être exécuté avec la variable d'environnement suivante définie au démarrage. Elle est nécessaire afin de récupérer le répertoire contenant les fichiers de configuration de l'application :

ENV=PRODUCTION

INFO : il ne faut pas mettre de « / » à la fin de la valeur de la variable et ne pas utiliser d'espace dans le chemin.

4.1.5 - Création de la base de données et génération des tables

- Créer un utilisateur et la base de donné du projet tel qu'indiqué dans le **Readme** du projet
- Dans le répertoire du projet, exécuter la commande **./manage.py makemigrations** pour mettre à jour les migrations
- Dans le répertoire du projet, exécuter la commande **./manage.py migrate** pour mettre à jour les tables de la base de donnée **(après avoir installer l'application)**

4.1.6 - Nginx et supervisor

- Mettre à jour nginx avec la commande '**sudo service nginx reload**'
- Lancer le superviseur avec la commande '**sudo supervisorctl reread**' puis la commande '**sudo supervisorctl update**'

4.1.7 - Répertoire de configuration applicatif (dossier contenant les settings du projet)

Le répertoire de configuration applicatif doit être créé sur le système de fichier et définit de la façon suivante :

$home_application_conf_directory/oc_pizza/settings.py

4.1.7.1 - Fichier oc_pizza/settings.py

Fichier contenant les différents paramètres du projet

4.1.8 - DataSources

Les accès aux bases de données doivent se configurer à l'aide du fichier **settings.py**

Aucun fichier de drivers **postgresql (postgresql-13.1)** n'est à installer.

4.1.9 - Vérifications

Afin de vérifier le bon déploiement de l'application, faire ceci:

- se rendre dans le répertoire principal (le répertoire contanant le fichier

‘manage.py’)

- Exécuter la commande : **./manage.py test tests**

5 - Procédure de démarrage / arrêt

5.1 - Base de données

- Pour démarrer la base de données, exécuter la commande : **sudo service postgresql start**
- Pour arrêter la base de données, exécuter la commande : **sudo service postgresql stop**

5.2 - Application web

- Démarrer le superviseur en éxécutant la commande '**sudo service supervisor start**'

6 - Procédure de mise à jour

6.1 - Base de données

- Dans le répertoire du projet, exécuter la commande **./manage.py makemigrations** pour mettre à jour les migrations
- Dans le répertoire du projet, exécuter la commande **./manage.py migrate** pour mettre à jour les tables de la base de donnée **(après avoir installer l'application)**

6.2 - Application web

- Dans le répertoire du projet, exécuter la commande **git pull** https://github.com/crispinpigla/substitutor_foods_website.git

7 - Supervision/Monitoring

7.1 - Monitoring de l’application web

7.1.1 - Intégration sentry

Sentry est intétgrer au projet.

Pour mettre à jour les paramètres de sentry :

- Créer un nouveau projet django dans un compte sentry
- Dans le fichier ‘**<<adresseversledossier dossier0>>**/pure_beurre_django/settings.py’ mettre à jour les paramètres de l’appel de la méthode **sentry.init** par ceux retournés par sentry
- Vérifier que sentry est installé dans l’environement virtuel
- Vérifier que les erreurs du projet apparaissent bien sur la page sentry du projet

7.1.2 - Intégration NewRelic

NewRelic est intétgrer au projet.

Pour mettre à jour les paramètres de NewRelic :

- Créer un nouveau projet django dans un compte Newrelic
- Dans le répertoire du projet (‘**<<adresseversledossier dossier0>>**’), remplacer le fichier de configuration NewRelic par celui téléchargé lors de la création du projet NewRelic
- Vérifier que newrelic est installé dans l’enviroment virtuel
- Vérifier que les informations du projet apparaissent bien dans l’onglet APM de Newrelic

7.2 - Supervision de l’application web

Exécuter la commande **sudo supervisorctl status** et vérifier que **RUNNING** doit etre indiqué dans le résultat affiché.

Les erreurs rencontrées par le superviseur sont indiquées dans le fichier ‘**<<adresse vers le dossier dossier0>>**/supervisor0.log’

8 - Procédure de sauvegarde et restauration

8.1 - Exporter la base de données

Exécuter dans le répertoire principal du projet la commande **./manage.py dumpdata oc_pizza_application_authentification,oc_pizza_application_client, oc_pizza_application_pizzaiolo,oc_pizza_application_livreur, oc_pizza_application_administrateur > purebeurre_dump.json** pour exporter la base de donnée dans un fichier **oc_pizza_dump.json**

8.2 - Importer la base de données

Exécuter dans le répertoire principal du projet la commande **./manage.py loaddata oc_pizza_dump.json** pour importer dans la base de données un fichier **oc_pizza_dump.json**

9 - Glossaire

Monitoring	Opération qui consiste à surveiller l'application et le serveur qui héberge l'application