Groupe OC Pizza

Projet de création d'un système informatique pour la gestion de l'ensemble des pizzerias du groupe OC Pizza

Dossier d'exploitation

Version 1.0.0

Auteur
DA SILVA David
Développeur iOS

Table des matières

1. Versions

Auteur	Date	Description	Version
DA SILVA David	02/06/2021	Création du document	0.5.0
DA SILVA David	05/06/21	Ajout de nouvelles sections	0.8.0
DA SILVA David	10/06/2021	Finalisation	1.0.0

2. Introduction

2.1 Objet du document

Le présent document constitue le dossier d'exploitation des applications du groupe OC Pizza. L'objectif du document est de présenter les démarches de déploiement, de mise à jour et de maintenance des applications web et mobiles du système de gestion du groupe OC Pizza.

2.2 Références

Pour de plus amples informations, se référer aux documents :

1. **Projet OC Pizza - specifications_fonctionnelles.pdf** : Dossier de conception fonctionnelle du système de gestion OC Pizza

2. **Projet OC Pizza - specifications_techniques.pdf** : Dossier de conception technique du système de gestion OC Pizza

3. Prérequis

3.1 Système

3.1.1 Serveur de base de données

Serveur de base de données Amazon RDS utilisant MySQL hébergeant le schéma OCPizza.

3.1.1.1 Caractéristiques techniques

MySQL est un serveur de base de données relationnelles SQL qui fonctionne sur de nombreux systèmes d'exploitation (dont Linux, Mac OS X, Windows, Solaris, FreeBSD...) et qui est accessible en écriture par de nombreux langages de programmation, incluant notamment PHP, Java, Ruby, C, C++, .NET, Python ...

L'une des spécificités de MySQL c'est qu'il inclut plusieurs moteurs de bases de données et qu'il est par ailleurs possibles au sein d'une même base de définir un moteur diffèrent pour les tables qui composent la base. Cette technique est astucieuse et permet de mieux

optimiser les performances d'une application. Les 2 moteurs les plus connus étant MyISAM (moteur par défaut) et InnoDB.

La réplication est possible avec MySQL et permet ainsi de répartir la charge sur plusieurs machines, d'optimiser les performances ou d'effectuer facilement des sauvegardes des données.

3.1.2 Serveur Web

Serveur virtuel Amazon EC2 hébergeant l'application web couplée à une instance Amazon RDS (Service de base de données relationnelle géré pour MySQL et d'autres SGBDR).

3.1.2.1 Caractéristiques techniques

- 750 heures par mois d'utilisation de bases de données db.t2.micro (moteurs de base de données applicables)
- 20 Go de stockage de base de données à usage général (SSD)
- 20 Go de stockage pour les sauvegardes et instantanés de base de données

3.1.3 Serveur de Batches

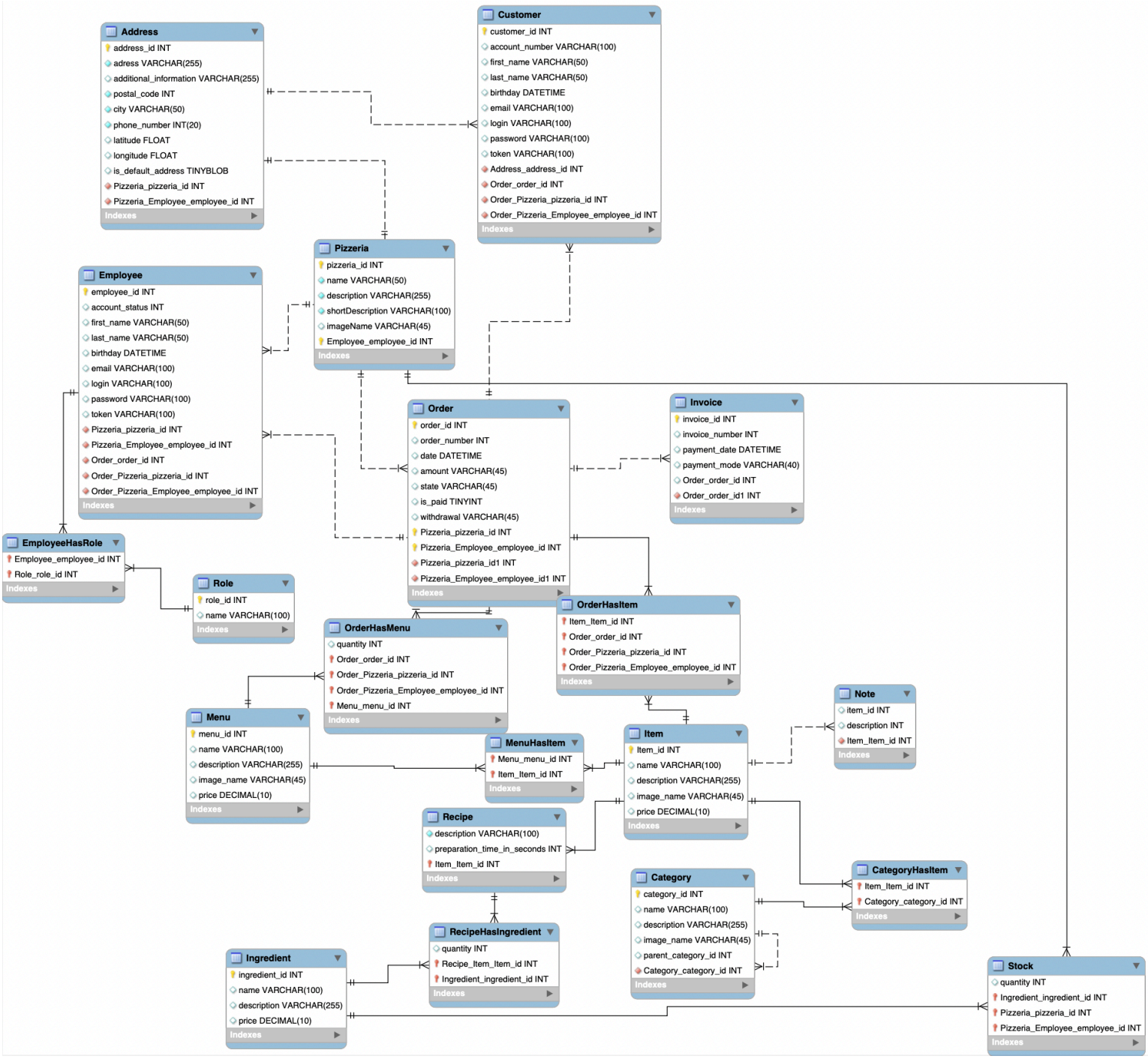
Utilisation d'AWS Batch pour la sauvegarde régulière des données du site notamment. AWS Batch permet aux développeurs d'exécuter aisément et efficacement plusieurs centaines de milliers de taches de calcul par batch sur AWS. AWS Batch alloue dynamiquement une quantité optimale et un type de ressources de calcul (par exemple, des instances optimisées pour la mémoire ou CPU) en fonction du volume et des besoins en ressources spécifiques des tâches par lots soumises.

3.1.4 Serveur de Fichiers

Le serveur de fichiers utilise un serveur Nginx en version 1.16.1 configuré pour servir les fichiers statiques des applications et pour rediriger les requêtes vers Django.

3.2 Bases de données

Une seule base de données est nécessaire au fonctionnement de l'application. Le schéma de la de données s'intitule "OCPizza". Ci-dessous le modèle physique de données en détails :



3.3 Web-services

Les web services suivants doivent être accessibles et à jour :

- POST /createAccount
- PUT/account/{id}
- DELETE/account/{id}
- POST /login
- POST /resetPassword
- GET /restaurants
- GET /products
- GET /products/{id}
- POST /orders
- GET /orders/{id}
- PUT/orders/{id}/{status}
- POST /payment

4. Procédure de déploiement

4.1 Applications Web

4.1.1 Comprendre et savoir démarrer un serveur EC2

EC2 signifie Elastic Compute Cloud (Service de calculs élastiques dans le cloud).
Amazon EC2 présente un environnement informatique vraiment virtuel, vous permettant d'utiliser des interfaces de service Web pour lancer des instances avec une variété de systèmes d'exploitation, les charger avec votre environnement d'applications personnalisées, gérer les autorisations d'accès à votre réseau, et exécuter votre image en utilisant autant ou aussi peu de systèmes que vous le désirez

Afin de mieux comprendre comment cela fonctionne et créer une instance pour héberger chacune de nos applications web voici les procédures à suivre dans l'ordre :

- Introduction aux services Amazon EC2
- Choisir le bon type d'instance
- Lancer une instance
- Se connecter à l'instance
- Héberger une page web
- Configurer une IP élastique

4.2 Déploiement de l'Application Web

4.2.1 Téléchargement des sources depuis le serveur

Récupérer les sources depuis le serveur à l'aide d'un git clone puis créer un nouvel environnement virtuel. Une fois créé l'activer et installer les dépendances de l'application :

git clone https://url/to/git/repository

```
sudo apt install virtualenv
virtualenv env -p python3
source env/bin/activate
pip install -r path/to/requirements.txt
```

4.2.2 - Environnement de l'application web

4.2.2.1 - Variables d'environnement

Le serveur d'application JOnAS* 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 :

-Dcom.ocpizza.apps.conf=$home_application_conf_directory

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

4.2.3 - Répertoire de configuration applicatif

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

$home_application_conf_directory/settings/

Les fichiers de configuration suivants sont disponibles :

- init__.py : fichier de configuration principal chargé par défaut
- production.py : fichier de configuration pour la production. Charge les identifiants de base et des variables propre à l'environnement de production
- travis.py : fichier de configuration à destination de l'outil d'intégration continue

4.2.3.1 - centengine.cfg

Fichier de configuration principal est généralement se trouvant dans le répertoire /etc/centreon-engine/.

4.2.3.2 - nagios.cfg

Contient un certain nombre de directives qui affectent la manière dont Nagios fonctionne.

4.2.4 - Vérifications

Afin de vérifier le bon déploiement de l'application, se connecter sur le compte Amazon et vérifier que les instance EC2 et RDS sont bien actives.

4.3 Applications Android

La procédure à suivre afin de publier sur le Google Play les applications Android du groupe OC Pizza est la suivante :

- Créer un compte Google Publisher (paiement unique de €25)
- Compiler et tester les applications
- Publier les applications sur le Play Store

Pour plus d'informations à ce sujet lire en détails cet article.

4.4 Applications iOS

La procédure à suivre afin de publier sur l'App Store les applications iOS du groupe OC Pizza est la suivante :

1. Créer un compte Apple Developer (paiement annuel de 99€)
2. Générer le certificat de distribution
3. Déclarer les applications via Apple Developer
4. Générer le provisioning profile
5. Créer les applications sur iTunes Connect
6. Télécharger Xcode et envoyer l'IPA
7. Soumettre sur l'App Store

5. Procédure de démarrage / arrêt

5.1 - Applications web

5.1.1 - Gunicorn

Gunicorn est un serveur http python pour distribution unix utilisant les spécifications WSGI* permettant de déployer un serveur de production.
Il est nécessaire d'installer Gunicorn via pip install si la librairie n'est pas dans le requirements.txt du projet puis démarrer le serveur Gunicorn pour vérification avec la commande suivante, en se situant dans le dossier du projet :

gunicorn <app_name>.wsgi:application

Naviguer ensuite vers l'url du serveur.
Plus d'informations se rendre ici **:** https://docs.gunicorn.org/en/stable/deploy.html

5.1.2 - Supervisor

Supervisor est un logiciel de supervision de processus permettant de démarrer ou redémarrer un processus via une commande.
Il est nécessaire d'installer Supervisor puis d'ajouter un fichier .conf dans /etc/supervisor/conf.d/ et d'y inscrire le processus à surveiller/démarrer/redémarrer :

sudo apt-get install supervisor

[program:<processus_name>]
command = /absolute_path_to/gunicorn <app_name>.wsgi:application user = <username>
directory = /absolute_path_to/project_dir
autostart = true // exécute la commande au premier démarrage du serveur
autorestart = true // exécute la commande si le processus est interrompu

// On indique à Django où trouver le module de settings
environment = DJANGO_SETTINGS_MODULE='<app_name>.settings.production'

Plus d'informations se rendre ici : https://docs.gunicorn.org/en/stable/deploy.html

5.2 - Application Android

Aucune procédure de démarrage / d'arrêt n'est disponible sur Android.
Ce qui peut éventuellement être fait est de développer une configuration qui peut être mise à jour coté serveur afin d'indiquer si l'application est en maintenance ou non. Dans le cas où le JSON* récupéré et parsé par l'application indique "true" alors une vue spéciale de maintenance est affichée sur le device Android de l'utilisateur.

5.3 - Application iOS

Aucune procédure de démarrage / d'arrêt n'est disponible sur iOS.
Ce qui peut éventuellement être fait est de développer une configuration qui peut être mise à jour coté serveur afin d'indiquer si l'application est en maintenance ou non. Dans le cas où le JSON récupéré et parsé par l'application indique "true" alors une vue spéciale de maintenance est affichée sur le device iOS de l'utilisateur.

6. Procédure de mise à jour

Afin de garantir une homogénéité dans les différentes applications du système de gestion du groupe OC Pizza un gitflow a été mis en place. Pour plus d'informations sur le sujet merci de consulter cette page web :

Worflow Gitflow

6.1 - Base de données

Concernant la mise à jour de la base de données il n'y a rien à faire car le service Amazon RDS se charge automatiquement de cette tâche.

Pour plus d'informations se rendre sur cette page :

https://openclassrooms.com/fr/courses/4810836-decouvrez-le-cloud-avec-amazon-web-servic es/5018441-sauvegarder-et-restaurer-la-base-de-donnees

6.2 - Application web

L'outil d'intégration continue Travis est utilisé afin de mettre automatiquement à jour le code en production lors de modifications ou de nouvelles fonctionnalités. Un fichier de configuration .travis.yml et un fichier de settings travis.py sont disponibles dans les sources de l'application.

Pour plus d'information sur l'utilisation de Travis CI, se référer aux lien suivants :

https://docs.travis-ci.com/user/tutorial/ https://docs.travis-ci.com/user/languages/python/

6.3 - Application Android

Pour les deux applications Android nous allons utiliser GitLab CI/CD couplé à fastlane* pour l'intégration continue et le déploiement continu afin de minimiser les tâches manuelles et automatiser un maximum les mises à jour ainsi que les vérifications de code (non régression, absence d'erreurs, etc.).

Pour plus d'informations à ce sujet merci de se référer aux liens suivants :

https://about.gitlab.com/blog/2018/10/24/setting-up-gitlab-ci-for-android-projects/
https://about.gitlab.com/blog/2019/01/28/android-publishing-with-gitlab-and-fastlane/

6.4 - Application iOS

Pour les deux applications iOS nous allons utiliser GitLab CI/CD couplé à fastlane pour l'intégration continue et le déploiement continu afin de minimiser les tâches

manuelles et automatiser un maximum les mises à jour ainsi que les vérifications de code (non régression, absence d'erreurs, etc.).

Pour plus d'informations à ce sujet merci de se référer aux liens suivants :

https://about.gitlab.com/blog/2016/03/10/setting-up-gitlab-ci-for-ios-projects/
https://about.gitlab.com/blog/2019/01/28/android-publishing-with-gitlab-and-fastlane/

7. Supervision/Monitoring

7.1 - Supervision de l'application web

Deux outils de supervisions vont être mis en place afin de superviser le système de gestion du groupe OC Pizza :

1. Sentry pour la partie application web et le framework Django
2. Nagios* / Centreon pour la partie matériel, espace de stockage, consommation RAM, etc.

1. L'outil Sentry est utilisé afin de superviser l'application.

La librairie sentry_sdk à était ajouté au projet ainsi qu'une configuration dans les settings de production. Afin de bénéficier de cet outil il faudra modifier le dns dans la configuration après avoir créé un compte sur le site.

Lien vers la documentation Sentry ici :
https://docs.sentry.io/platforms/python/guides/django/

2. Centreon est un logiciel de supervision informatique édité par la société du même nom.
L'ensemble des solutions Centreon reposent sur un environnement de base, totalement open source, baptisé OSS pour Open Source Software. Cette solution donne :

- La possibilité d'avoir une vue synthétique de la supervision de son système d'informations
- La visualisation de graphiques de performances
- Des rapports de disponibilités des ressources supervisées : hôtes, services et groupes de ressources (disponible via IHM et exportables en csv)
- Une interface de configuration intuitive pour les différents objets et fichiers de configurations des ordonnanceurs
- La possibilité d'administrer chaque paramètre de l'interface web
- La possibilité de mettre en place des accès restreints aux ressources et pages de l'interface, configurables de manières fines (via des LCA : Liste de Contrôles d'Accès)
- La possibilité de suivre des logs d'utilisation de la solution (logs de modifications des ressources)

- La possibilité de construire un tableau de bord à l'aide de widgets graphiques (carte Google Maps, listage des ressources, graphiques de performance...)
- La possibilité de développer des modules additionnels pour étendre les fonctionnalités de la solution.

Lien vers la documentation Centreon ici : https://docs.centreon.com/current/fr/

8. Procédure de sauvegarde et de restauration

8.1 Sauvegarde de la base de données

Il est bon de savoir que par défaut votre instance Amazon RDS est sauvegardée tous les jours sur les 7 derniers jours de manière automatisée. Toutefois, si vous voulez lancer une sauvegarde manuellement (avant une opération importante par exemple), il vous suffit de sélectionner le serveur et de demander à "Prendre un instantané". Afin de sauvegarder notre instance de base de données suivre la procédure se trouvant ici.

8.2 - Restauration de la base de données

Afin de restaurer notre instance de base de données suivre la procédure se trouvant ici.

8.3 - Sauvegarde et restauration des applications web et mobiles

Pour la sauvegarde il est à noter que l'ensemble du code est versionné sur GitLab et qu'à chaque mise à jour en production que ça soit un hotfix ou une évolution de mise à jour un tag est créé afin de pouvoir faire une restauration en cas de problème. Le fait également d'utiliser Gitflow permet de sécuriser et de faciliter les reverts en cas d'éventuel problème en production. Pour la partie web utilisant une instance Amazon EC2 il est possible d'effectuer des sauvegardes quotidiennes. Afin de sauvegarder et restaurer les instances Amazon EC2 suivre la procédure suivante :

- Sauvegarder et restaurer une instance EC2

9. GLOSSAIRE

Nagios	Nagios est une application permettant la surveillance système et réseau. Elle surveille les hôtes et services spécifiés, alertant lorsque les systèmes ont des dysfonctionnements et quand ils repassent en fonctionnement normal. C'est un logiciel libre sous licence GPL.
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.
fastlane	Fastlane est un outil open source qui permet de faire du Continuous Delivery sous iOS et Android. Il permet d'automatiser un certain nombre d'étapes de notre build, comme, par exemple, la gestion des provisioning profiles, la compilation, la génération de captures d'écran et la publication sur les stores ou sur des plates-formes telles que Fabric.
WSGI	La Web Server Gateway Interface (WSGI) est une spécification qui définit une interface entre des serveurs et des applications web pour le langage Python.
JSON	JavaScript Object Notation est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée comme le permet XML par exemple.
JOnAS	JOnAS, qui signifie Java Open Application Server, était un serveur d'applications open source multiplate-forme, et multi bases de données, permettant de bâtir des services applicatifs métier.