|  |
| --- |
| **OC Pizza**  **Système de gestion**  Dossier d'exploitation  Version 1.0.0 |
| **Auteur**  Damien Gironnet  *Analyste-programmeur* |

Table des matières

[1 - Versions 4](#_Toc56177223)

[2 - Introduction 5](#_Toc56177224)

[2.1 - Objet du document 5](#_Toc56177225)

[2.2 - Références 5](#_Toc56177226)

[3 - Pré-requis 6](#_Toc56177227)

[3.1 - Système 6](#_Toc56177228)

[3.1.1 - Serveur de Base de données 6](#_Toc56177229)

[3.1.2 - Serveur d’application 6](#_Toc56177230)

[3.2 - Bases de données 6](#_Toc56177231)

[3.3 - Web services 7](#_Toc56177232)

[4 - Procédure de déploiement 8](#_Toc56177233)

[4.1 - Ressources & Installation logiciel 8](#_Toc56177234)

[4.2 - Variables d'environnement 8](#_Toc56177235)

[4.3 - Déploiement de la base de données 9](#_Toc56177236)

[4.3.1 - Vérifications 10](#_Toc56177237)

[4.4 - Déploiement du serveur d’application et de l’interface internet 10](#_Toc56177238)

[4.4.1 - Vérifications 13](#_Toc56177239)

[4.5 - Déploiement de l’interface boutique 14](#_Toc56177240)

[5 - Procédure de démarrage / arrêt 15](#_Toc56177241)

[5.1 - Serveur de base de données 15](#_Toc56177242)

[5.2 - Serveur d’application 15](#_Toc56177243)

[6 - Procédure de mise à jour 16](#_Toc56177244)

[6.1 - Serveur de base de données 16](#_Toc56177245)

[6.2 - Serveur d’application 16](#_Toc56177246)

[7 - Supervision/Monitoring 18](#_Toc56177247)

[7.1 - Supervision du serveur de base de données 18](#_Toc56177248)

[7.2 - Supervision du serveur d’application 19](#_Toc56177249)

[8 - Procédure de sauvegarde et restauration 20](#_Toc56177250)

# Versions

|  |  |  |  |
| --- | --- | --- | --- |
| Auteur | Date | Description | Version |
| Damien Gironnet | 13/10/2020 | Création du document | 1.0.0 |

# Introduction

## Objet du document

Le présent document constitue le dossier d’exploitation du système de gestion de pizzérias d’OC PIZZA. Ce document présente à l’équipe technique d’OC PIZZA, les procédures de mise en œuvre et de maintenance du système.

## Références

Pour de plus amples informations, se référer également aux éléments suivants :

1. DCF - 1.0.0 : Dossier de conception fonctionnelle de l'application
2. DCT - 1.0.0 : Dossier de conception technique de l'application
3. PVL - 1.0.0 : Procès-verbal de livraison de l'application

# Pré-requis

## Système

L’ensemble du système D’OC Pizza est hébergé sur un unique serveur Windows Server 2019 dans les locaux du siège de la société de manière à pouvoir superviser et comparer l’activité des différentes pizzérias du groupe.

### Serveur de Base de données

La base de données est une base de type PostgreSQL, hébergée sur le serveur Windows Server 2019 de la société́ OC-Pizza.

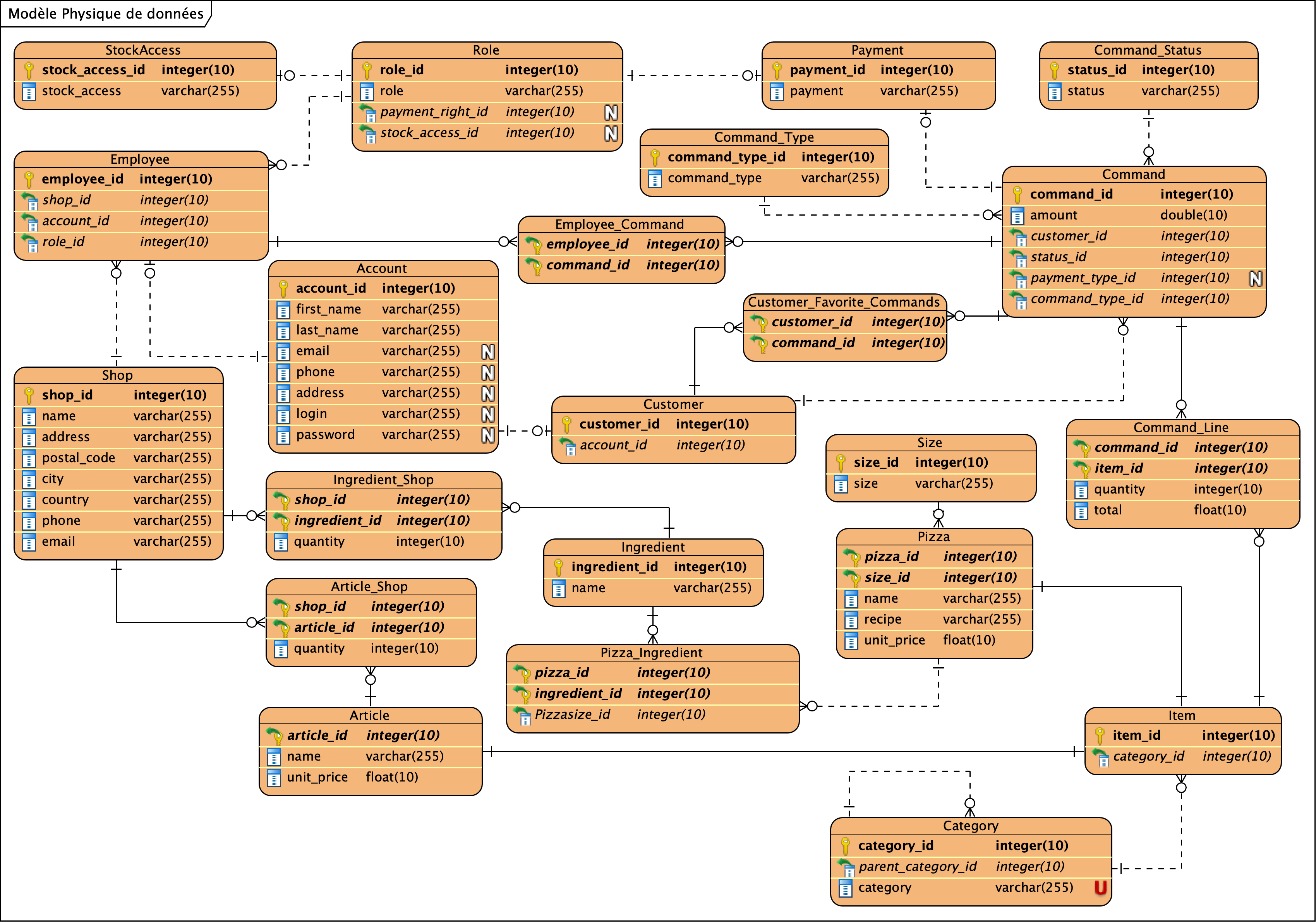
### Serveur d’application

Le serveur est un serveur d’application GlassFish qui regroupe à la fois l’application internet dans un conteneur web et la logique métier dans un conteneur métier EJB qui sert à la fois pour l’application web et pour l’application Java/Swing destiné aux employés dans les points de vente.

## Bases de données

Le document ci-dessous représente le modèle physique de donnée qui regroupe les différentes tables de la base de données ainsi que leurs cardinalités.

Les scripts de création des tables ont été́ fournis à l’équipe de déploiement à l’adresse <https://github.com/itconsulting/ocpizza/scripts>.



## Web services

Le système utilise pour le paiement en ligne un Web Service externe APIGEE qui est une plateforme Google de gestion D’apis.

# Procédure de déploiement

Le code source de l’application se trouve à l’adresse <https://github.com/itconsulting/ocpizza/source>.

## Ressources & Installation logiciel

pour pouvoir déployer l’application, il faut installer les deux serveurs :

* Pour le serveur de base de données **PostgreSQL**, suivez les instructions à l’adresse <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads> ;
* Pour le serveur d’application **GlassFish**, suivez les instructions à l’adresse <https://javaee.github.io/glassfish/download>;

Il faut aussi installer l’outil de gestion et d'automatisation de production **Maven.** Pour cela, suivez les instructions disponibles à l’adresse <https://maven.apache.org/users/index.html>

## Variables d'environnement

Voici les variables d'environnement nécessaire au déploiement de l’application OC Pizza :

|  |  |  |
| --- | --- | --- |
| Nom | Obligatoire | Description |
| JAVA\_HOME | oui | Répertoire racine contenant le Java Development Kit (JDK) |
| MAVEN\_HOME | oui | Répertoire racine de l’outil Maven |
| POSTGRESQL\_HOME | oui | Répertoire racine du serveur PostgreSQL |
| GLASSFISH\_HOME | oui | Répertoire racine du serveur d’application |

Une fois les variables définies, il faut maintenant pour accéder aux commandes en ligne de Maven, PostgreSQL et GlassFish, ajouter au chemin global (PATH) les chemins **%MAVEN%\bin**, **%POSTGRESQL\_HOME%\bin** et **%GLASSFISH\_HOME%\bin**.

## Déploiement de la base de données

Les scripts nécessaires au déploiement de la base de données se trouve à l’adresse <https://github.com/itconsulting/ocpizza/scripts>.

Pour commencer, il faut créer la base de données

createdb ocpizza

Ensuite il faut créer l’utilisateur

createuser -s postgres

Maintenant déplacez-vous dans le répertoire contenant les scripts et tapez la commande suivante

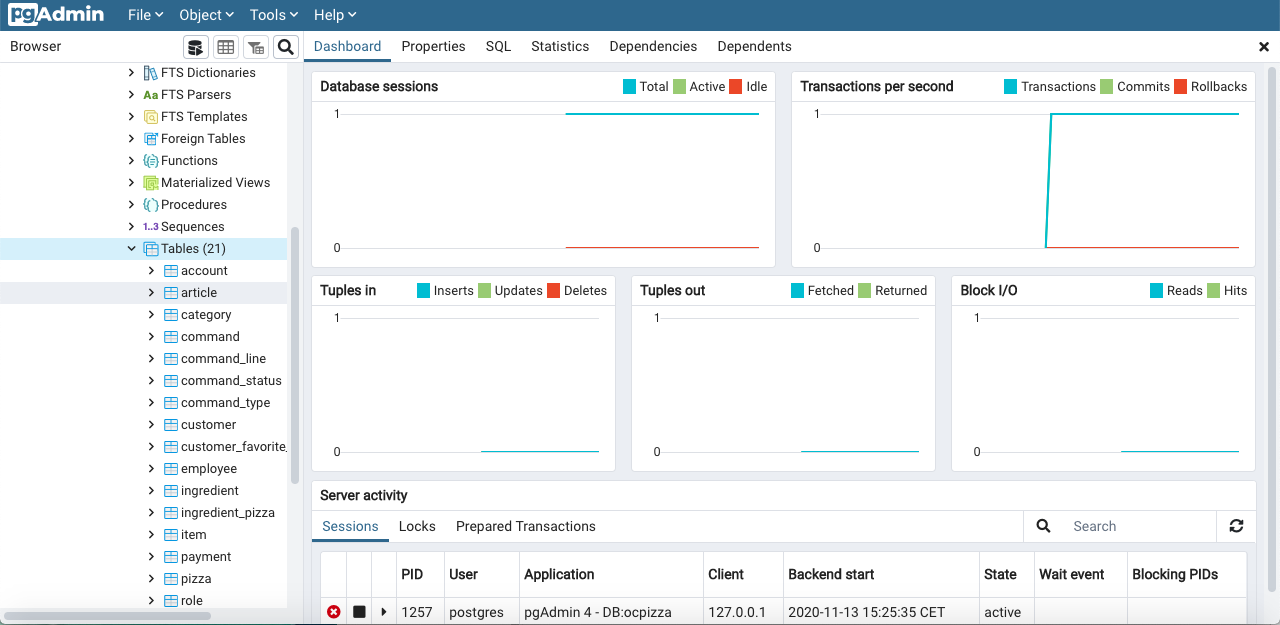
psql -d ocpizza < postgres\_create\_db.sql

### Vérifications

Démarrez pgAdmin, l’interface graphique de PostgreSQL.

Vous pouvez vérifier à l’adresse suivant <http://127.0.0.1:50022/browser/>.

Vous devriez obtenir le résultat suivant dans la base de données **ocpizza** :



## Déploiement du serveur d’application et de l’interface internet

Pour créer le domaine **ocpizza** dans GlassFish il faut taper la commande ci-dessous (le port d'admin est 4848)

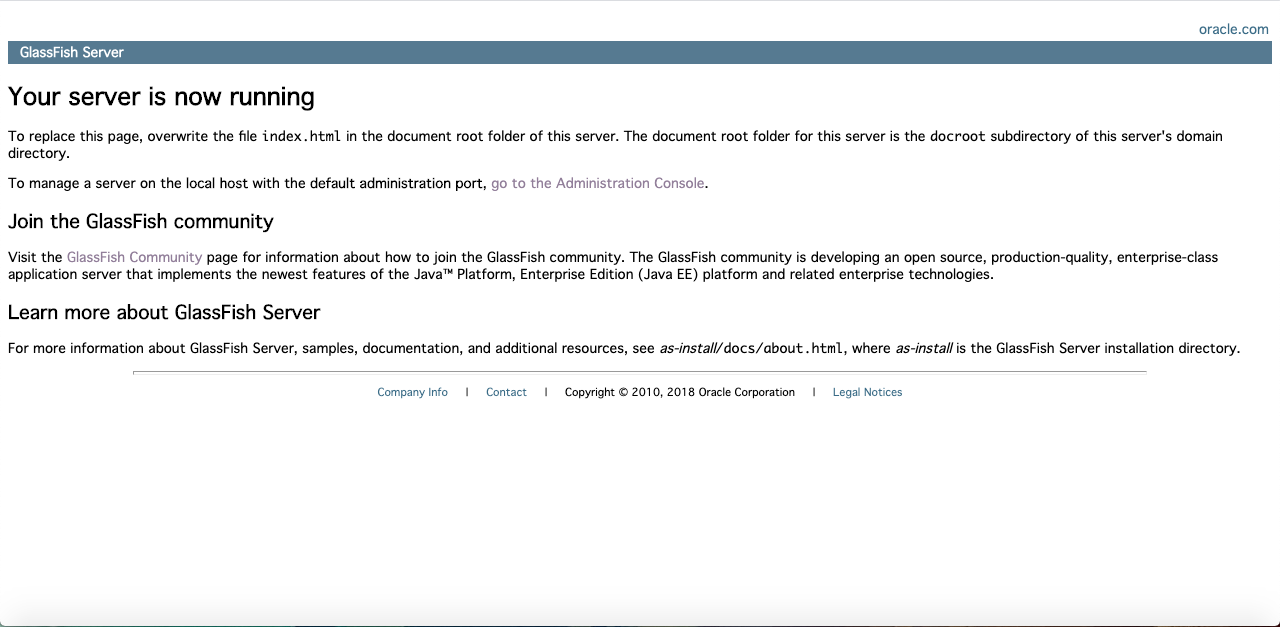
asadmin create-domain --adminport 4848 --instanceport 8080 --nopassword=true ocpizza

Maintenant pour lancer le domaine qui vient d’être créé tapez la commande suivante

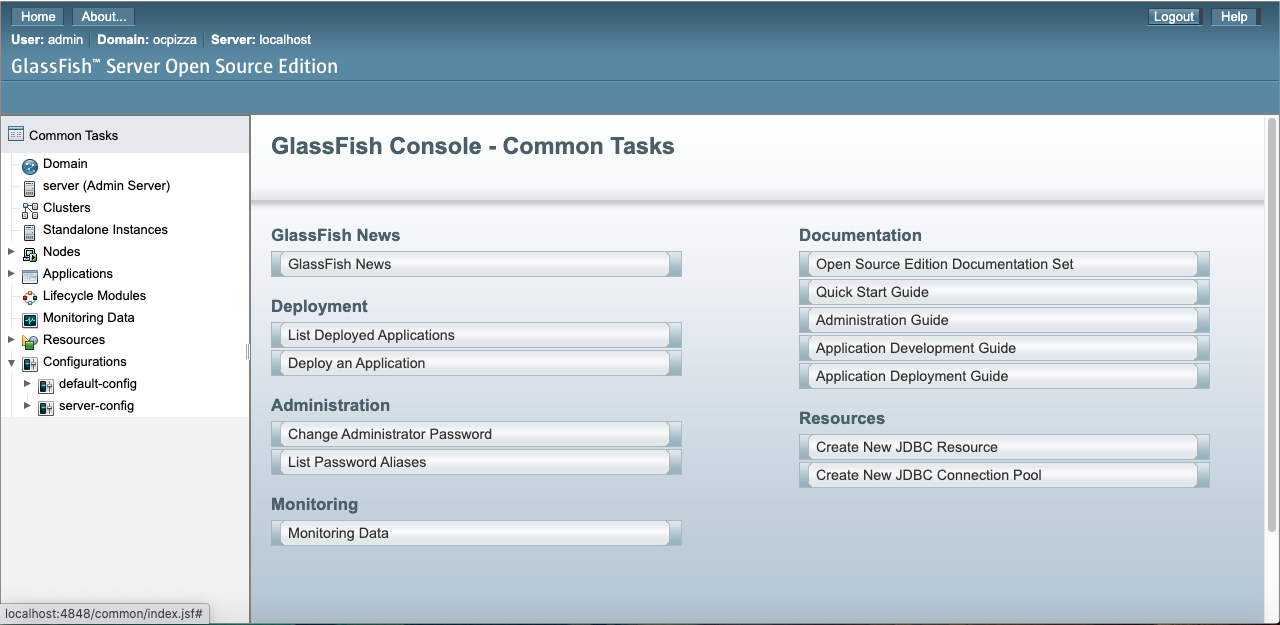
asadmin start-domain ocpizza

Vérifier que le serveur fonctionne en allant sur :

<http://localhost:8080/>



<http://localhost:4848/> (pour la console d'administration)



Si votre serveur ne démarre pas, consultez le fichier

%GLASSFISH\_HOME%\glassfish\domains\ocpizza\logs\server.log

Pour construire l'application. déplacez-vous à la racine du code source puis taper

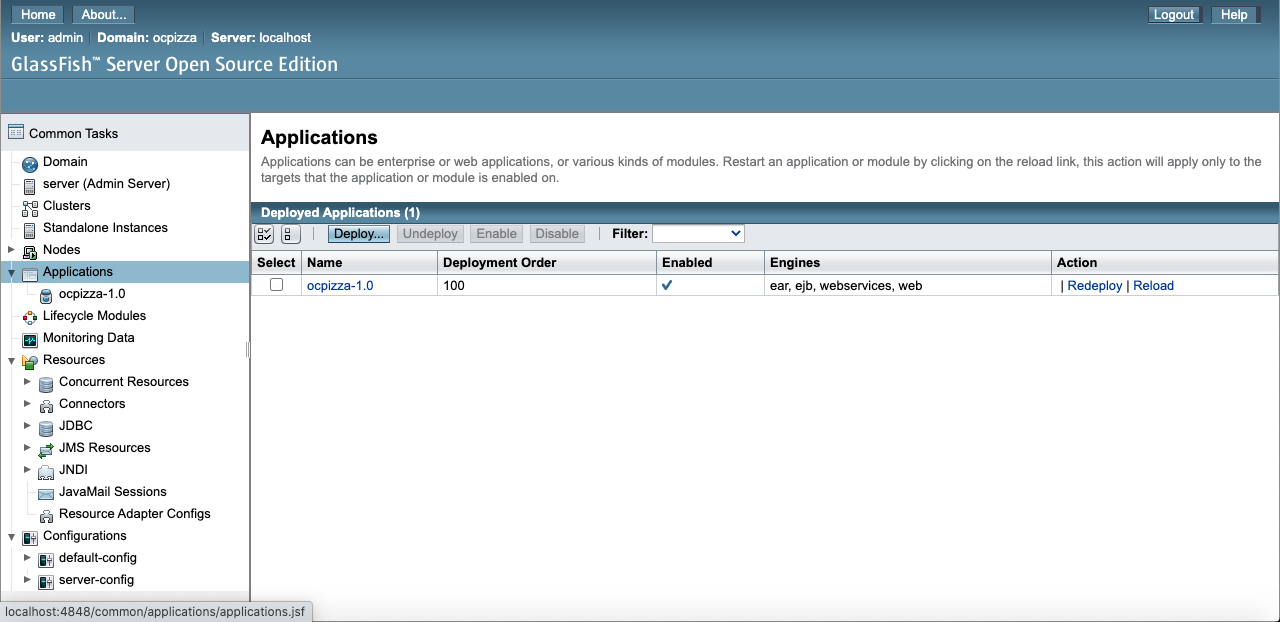
mvn clean package

Enfin, pour déployer l’application sur le serveur GlassFish tapez la commande

asadmin –port 4848 deploy –force ear\target\ocpizza-1.0.ear

### Vérifications

Vous obtenez alors l’aperçu suivant dans la partie « applications » de la console d’administration.



Vous pouvez désormais accéder au site à l’adresse <http://locahost:8080/ocpizza>

## Déploiement de l’interface boutique

L’interface boutique est une application développée en Java/Swing. Chaque boutique est équipée de tablette et d’ordinateur Windows sur lesquels vont être déployé cette interface.

Pour installer l’interface boutique, vous devez, sur chaque terminal, importer le code source du projet disponible à l’adresse <https://github.com/itconsulting/ocpizza/source> , installer **Maven** (de la même manière que pour le serveur physique d’OC Pizza) et créer la variable d’environnement associée.

Une fois ceci fait, pour exécuter et déployer l'application Swing sur les différents terminaux, déplacez-vous dans le répertoire « Swing » du code source puis tapez la commande suivante :

mvn exec:java -Dexec.mainClass="com.itconsulting.ocpizza.client.ui.OCPizzaFrame"

Vous accédez ainsi à l’interface boutique qui vous permettra de gérer les stocks et les commandes de la boutique.

# Procédure de démarrage / arrêt

## Serveur de base de données

Pour démarrer le serveur PostgreSQL en ligne de commande, exécuter la commande suivante :

pg\_ctl -D "%POSTGRESQL\_HOME\data" start

Pour stopper le serveur PostgreSQL en ligne de commande, exécuter la commande suivante :

pg\_ctl -D "%POSTGRESQL\_HOME\data" stop

## Serveur d’application

Pour démarrer le serveur GlassFish en ligne de commande, exécuter la commande suivante :

asadmin start-domain ocpizza

Pour stopper le serveur GlassFish en ligne de commande, exécuter la commande suivante :

asadmin stop-domain ocpizza

# Procédure de mise à jour

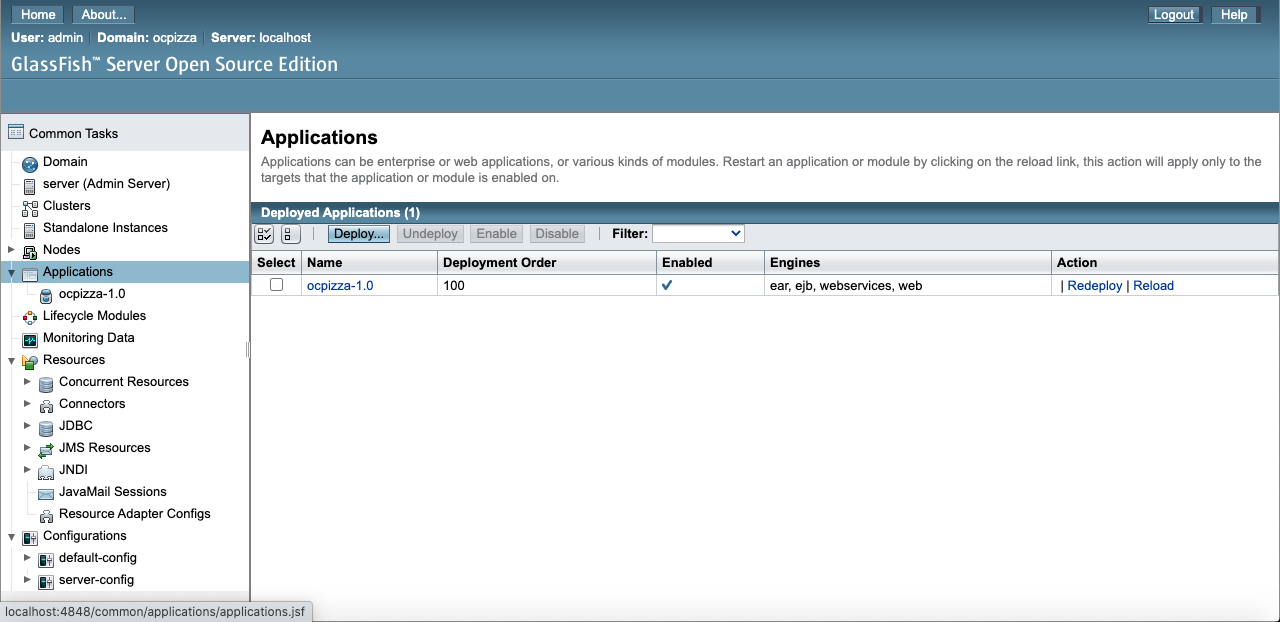
## Serveur de base de données

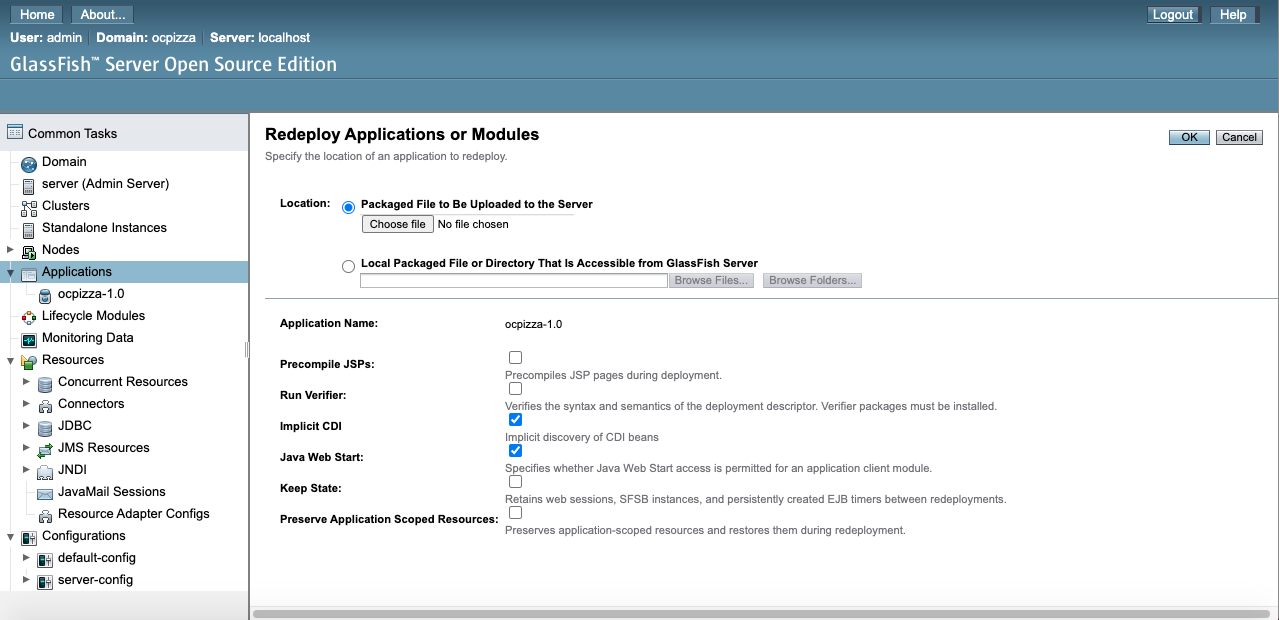
En cas de nécessité́ de mise à jour de la base de données, se connecter à la console d’administration, et exécuter le script qui sera fourni pour ce faire : *script\_update\_ddMMYYYY.sql*.

Avant toute mise à jour, lancer une sauvegarde de la base.

## Serveur d’application

En cas de nécessité́ de mise à jour de l’application, se connecter à la console d’administration, cliquer sur l’onglet « Applications », cliquez sur « redeploy », sélectionnez le fichier .ear dans « choose file ».





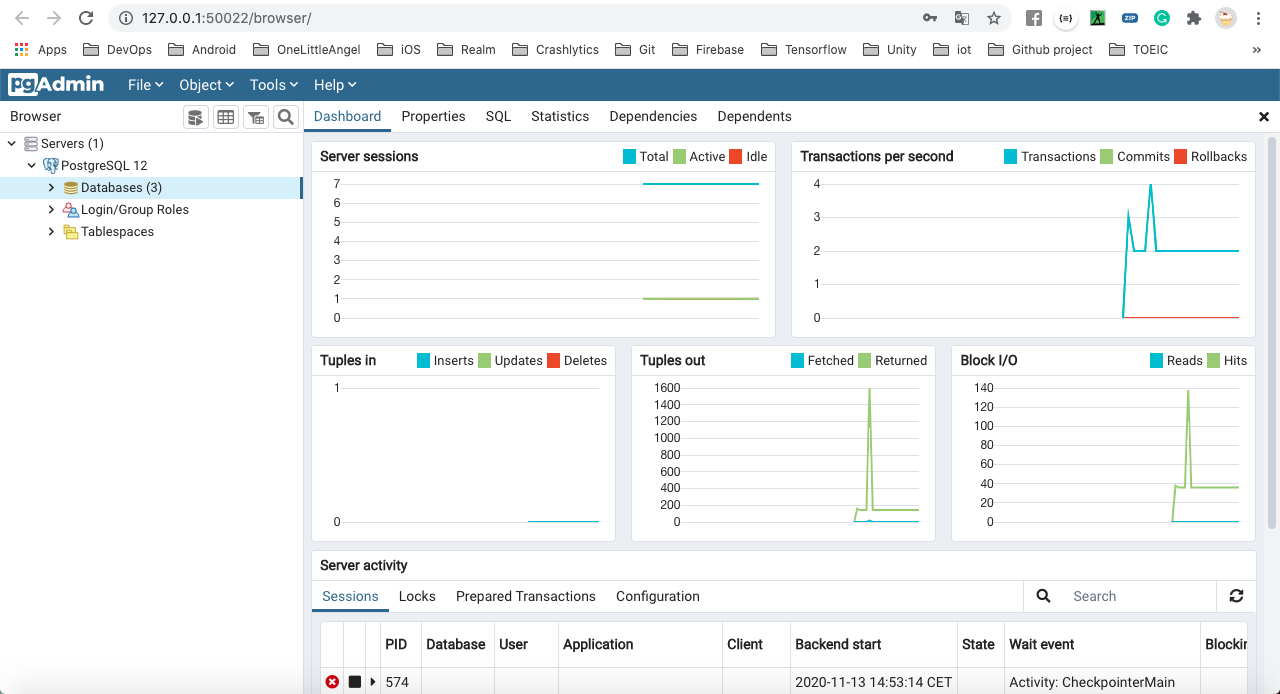
# Supervision/Monitoring

## Supervision du serveur de base de données

Pour superviser la base de données, vous pouvez utiliser le logiciel pgAdmin qui est installer en même temps que la base de données. Vous pouvez y accéder à l’adresse suivante :

<http://127.0.0.1:50022/browser/>

Voici un aperçu :

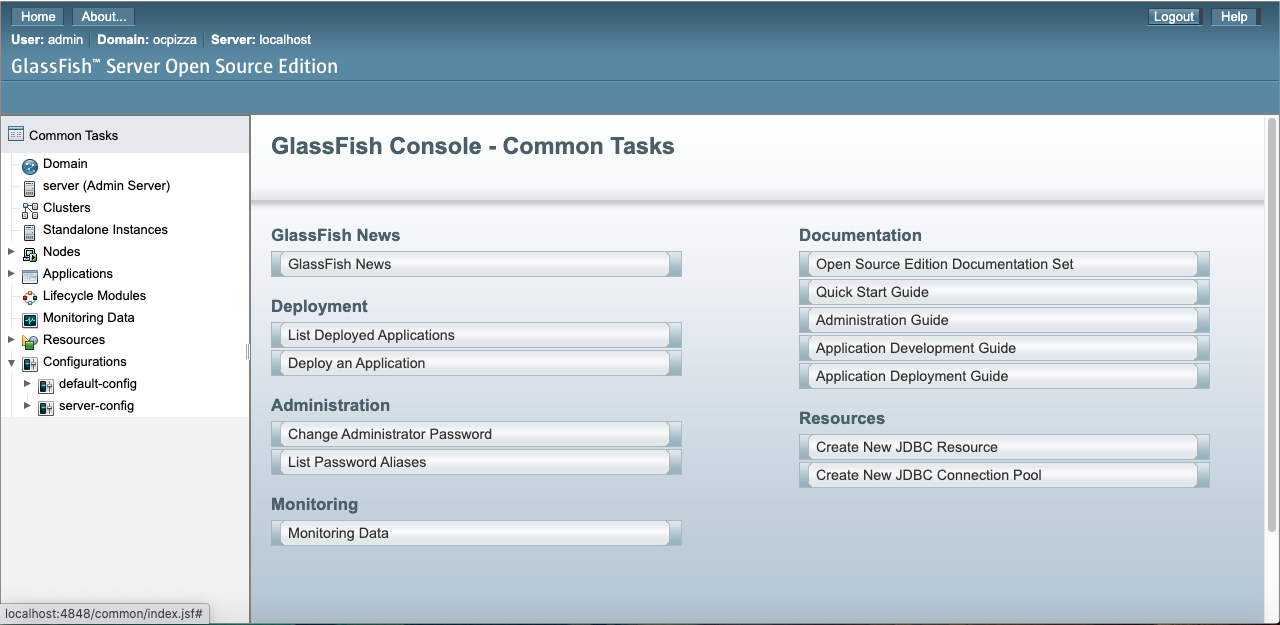


## Supervision du serveur d’application

Pour superviser la base de données, vous pouvez utiliser le logiciel pgAdmin qui est installer en même temps que la base de données. Vous pouvez y accéder à l’adresse suivante :

<http://localhost:4848/common/index.jsf>

Voici un aperçu :



# Procédure de sauvegarde et restauration

Pour sauvegarder les données de la base, dans la console, exécuter la commande suivante :

pg\_dump --user=root --password=root password --all-databases >

ocbdd\_save\_ddMMYYYY.backup

Une tâche CRON quotidienne sera mise en place à *03h00 AM* pour sauvegarder la base. Les sauvegardes quotidiennes seront conservées un an.

Afin de répondre aux exigences de la CNIL sur la protection des données personnelles, tous les premiers jours du mois à *03h30 AM*, une tache planifiée simplifiera et anonymisera les données dont l’ancienneté́ dépasse un an.

Pour restaurer la base à un état précédant, exécuter la commande suivante :

pg\_dump --user=root --password=root password < ocbdd\_save\_ddMMYYYY.backup