|  |
| --- |
| **OC Pizza**  **Projet08**  Dossier d'exploitation  Version 1.0 |
| **Auteur**  Laurent Cordier  *Développeur d'application* |

Table des matières

1 - Versions 4

2 - Introduction 5

2.1 - Objet du document 5

2.2 - Références 5

3 - Pré-requis 6

3.1 - Système 6

3.1.1 - Système de gestion de base de données 6

3.1.2 - Serveurs de Base de données, de Batches, de Fichiers 6

3.1.2.1 - Caractéristiques techniques 6

3.2 - Base de données 6

3.3 - Serveurs Web 6

3.4 - Containers (services web) 6

3.4.1.1 - Caractéristiques techniques 7

3.5 - Autres Ressources - FireWall 7

4 - Procédure de déploiement 8

4.1 - Déploiement des Batches 8

4.1.1 - Artefacts 8

4.1.2 - Variables d'environnement 8

4.1.3 - Configuration 8

4.1.3.1 - Fichier xxx.yyy 9

4.1.3.2 - Fichier zzz.ttt 9

4.1.3.3 - Fichier ... 9

4.1.4 - Ressources 9

4.1.5 - Vérifications 9

4.2 - Déploiement de l'Application Web 10

4.2.1 - Artefacts 10

4.2.2 - Environnement de l’application web 10

4.2.2.1 - Variables d’environnement 10

4.2.3 - Répertoire de configuration applicatif 10

4.2.3.1 - Fichier xxx.yyy 10

4.2.4 - DataSources 10

4.2.5 - Ressources 11

4.2.6 - Vérifications 11

5 - Procédure de démarrage / arrêt 12

5.1 - Base de données 12

5.2 - Batches 12

5.3 - Application web 12

6 - Procédure de mise à jour 13

6.1 - Base de données 13

6.2 - Batches 13

6.3 - Application web 13

7 - Supervision/Monitoring 14

7.1 - Supervision de l’application web 14

8 - Procédure de sauvegarde et restauration du SGBD 15

8.1 - commande de sauvegarde 15

8.2 - Procédure de restauration (https://doc.postgresql.fr/9.5/continuous-archiving.html) 16

9 - Glossaire 17

# Versions

|  |  |  |  |
| --- | --- | --- | --- |
| Auteur | Date | Description | Version |
| Laurent Cordier | 24/12/2019 | Création du document | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Objet du document

Le présent document constitue le dossier d’exploitation de l'application OC Pizza portant sur la mise en place d’un nouveau système informatique pour l’ensemble des pizzerias du groupe.

## Références

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

* **DCF – Projet08** : Dossier de conception fonctionnelle de l'application.
* **DCT** **– Projet08** : Dossier d’exploitation.

# Pré-requis

## Système

### Système de gestion de base de données

Le SGBD PostgreSQL 12.1doit être installé et actif en tant que service Windows avec démarrage automatique. L’outil pgAdmin V4 doit être installé et fonctionnel.

### Serveurs de Base de données, de Batches, de Fichiers

Sans objet – Le DBMS est hébergé par le serveur unique d’OC Pizza.

#### Caractéristiques techniques

OC Pizza exploite un serveur sous Windows server 2019.

## Base de données

Le répertoire c:\bdData doit exister et avoir les droits d’accès en rwx pour le compte « Système locale ».

La base de données db\_projet08 doit être accessible et à jour.

Le mot de passe requis pour la connexion au SGBD est celui choisi lors de son installation.

Le mot de passe requis pour la connexion à la base de connées db\_projet08 est celui transmis pas courrier séparé par « IT Consulting » au DBA de « OC Pizza ».

## Serveurs Web

## Containers (services web)

Les containers suivants doivent être accessibles et à jour :

* **Service-FrontalWeb :** version 1.0
* **Service-**Crud : version 1.0
* **Service-**intfPaiement : version 1.0

#### Caractéristiques techniques

Java en version 11.0.4 doit être installé.

Apache Maven en version 3.6.2 doit être installé.

## Autres Ressources - FireWall

Le firewall Windows doit accepter les échanges :

* sur l’adresse localhost pour les ports 9202 et 9203 de même adresse.
* En accès externe pour les ports 9201, 4201, 4202, 4203.

# Procédure de déploiement

## Déploiement des Batches

### Artefacts

Les batches de l’application Xxx sont construits sous la forme d'une archive ZIP contenant les répertoires :

* **bin** : les scripts SH de lancement des différents batches
* **conf** : les fichiers de configuration
* **...** : ...

Extraire l'archive **XXX.zip** dans le répertoire :

**/xxx/yyy**

Positionner les droits d'exécution sur les scripts SH de lancement des batches.

….

### Variables d'environnement

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

|  |  |  |
| --- | --- | --- |
| Nom | Obligatoire | Description |
| XXX\_HOME | non | Répertoire racine de l’installation de l’application |
|  |  |  |

Définissez les variables d’environnement nécessaires comme ceci...

...

### Configuration

Voici les différents fichiers de configuration :

* **xxx.yyy** : fichier de configuration des logs
* **zzz.ttt** : fichier de configuration de l'application...
* **aaa** : fichier de configuration de la ressources XXX
* ...

#### Fichier xxx.yyy

…

#### Fichier zzz.ttt

...

#### Fichier ...

### Ressources

...

### Vérifications

Afin de vérifier le bon déploiement des batches, faire ceci…

## Déploiement de l'Application Web

### Artefacts

...

### Environnement de l’application web

#### 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**

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.

### 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/applicationX**

… fichiers de configuration… :

* ...

#### Fichier xxx.yyy

...

### DataSources

Les accès aux bases de données doivent se configurer à l'aide des fichiers…

Le fichier de drivers **postgresql (postgresql-9.2.x.)** doit être déposé dans le répertoire :

**$home\_server/lib/ext**

...

### Ressources

...

### Vérifications

Afin de vérifier le bon déploiement de l’application, faire ceci…

# Procédure de démarrage / arrêt

## Base de données

## Batches

## Application web

# Procédure de mise à jour

## Base de données

## Batches

## Application web

# Supervision/Monitoring

## Supervision de l’application web

Dans un navigateur, appelez les url :

[http://localhost:9201/actuator/health/](http://localhost:9091/actuator/health/) pour vérifier le bon fonctionnement du service-FrontWeb

<http://localhost:9202/actuator/health/> pour vérifier le bon fonctionnement du service-IntfPaiement

<http://localhost:9203/actuator/health/> pour vérifier le bon fonctionnement du service-Crud

Le navigateur affiche :

{"status":"UP"} lorsque la santé du service est normal

{"status":"DOWN"} lorsque le service n’est pas en état de fonctionnement normal.

En cas d’erreur 404, le service est arrété.

# Procédure de sauvegarde et restauration du SGBD

L’exploitation d’une base de données et en particulier du SBGD PostgreSQL est hors périmètre de ce document. Les informations qui concernent les sauvegardes et restaurations ainsi que le détail de leur mise en œuvre sont présentées ici : <https://doc.postgresql.fr/9.5/continuous-archiving.html>

Note :

* Le fichier recovery.conf n’existe plus en version 12.
* Le fichier postgresql.conf contient les informations précédemment contenues dans l’ancien fichier recovery.conf.

Les paragraphes suivants présentent une commande de sauvegarde et une méthode de restauration.

## commande de sauvegarde

La commande présentée correspond à une sauvegarde physique au fil de l’eau à l’aide de l’outil, interactif ou non , pg\_basebackup fourni avec PostgreSQL.

Le répertoire « c:/dataSauveNAS/repSauve » est créé par l’administrateur système qui se charge de la sauvegarde journalière sur support externe. La nature du support externe (Cloud, NAS, robot de DON, …) est hors périmètre de ce document.

La commande dans sa version interactive est :

pg\_basebackup

--pgdata= c:/dataSauveNAS/repSauve/ « jjmmyyyy »

--format=tar

--write-recovery-conf

--xlog-method=stream

--progress

--host=localhost

--port= 5432

--username=rl\_projet08

--password

où jjmmyyyy est à valoriser avec la date de la sauvegarde journalière. Le password a été transmis par document séparé par IT Consulting lors de la livraison.

Le répertoire c:/dataSauveNAS/repSauve /« jjmmyyyy » contient les fichiers base.tar, pg\_wal pour les journaux de transaction et un fichier xxxx.tar qui correspond au tablespace des tables de la base de données db\_projet08.

Le fichier Postgresql.conf se trouve dans le fichier base.tar de la sauvegarde.

## Procédure de restauration (<https://doc.postgresql.fr/9.5/continuous-archiving.html>)

1. Arrêter le serveur s'il est en cours d'exécution.
2. Si la place nécessaire est disponible, copier le répertoire complet de données du cluster et tous les tablespaces dans un emplacement temporaire en prévision d'un éventuel besoin ultérieur. Cette précaution nécessite qu'un espace suffisant sur le système soit disponible pour contenir deux copies de la base de données existante. S'il n'y a pas assez de place disponible, il faut au minimum copier le contenu du sous-répertoire pg\_xlog du répertoire des données du cluster car il peut contenir des journaux qui n'ont pas été archivés avant l'arrêt du serveur.
3. Effacer tous les fichiers et sous-répertoires existant sous le répertoire des données du cluster et sous les répertoires racines des tablespaces.
4. Restaurer les fichiers de la base de données à partir de la sauvegarde des fichiers. Il faut veiller à ce qu'ils soient restaurés avec le bon propriétaire (l'utilisateur système de la base de données, et non pas root !) et avec les bons droits. Si des tablespaces sont utilisés, il faut s'assurer que les liens symboliques dans pg\_tblspc/ ont été correctement restaurés.
5. Supprimer tout fichier présent dans pg\_xlog/ ; ils proviennent de la sauvegarde et sont du coup probablement obsolètes. Si pg\_xlog/ n'a pas été archivé, il suffit de recréer ce répertoire en faisant attention à le créer en tant que lien symbolique, si c'était le cas auparavant.
6. Si des fichiers de segment WAL non archivés ont été sauvegardés dans l'étape 2, les copier dans pg\_xlog/. Il est préférable de les copier plutôt que de les déplacer afin qu'une version non modifiée de ces fichiers soit toujours disponible si un problème survient et qu'il faille recommencer.
7. Modifier le fichier de commandes postgresql.conf avec restore\_command = 'copy " c:\\dataSauveNAS/\\repSauve \\« jjmmyyyy »\\pg\_wal.tar" "%p"'
8. Démarrer le serveur. Le serveur se trouve alors en mode récupération et commence la lecture des fichiers WAL archivés dont il a besoin. A la fin du processus de récupération, le serveur modifie postgresql.conf en commentant la ligne restore\_command  (pour éviter de retourner accidentellement en mode de récupération), puis passe en mode de fonctionnement normal.

# Glossaire

|  |  |
| --- | --- |
|  |  |
|  |  |