|  |
| --- |
| **OC Pizza**  **Système de gestion pour pizzeria**  Dossier d'exploitation  Version 1.2 |
| **Auteur**  Mathieu Lemaire  *Développeur d’application Android* |

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 -Serveur de Base de données 6

3.1.1.1 -Caractéristiques techniques 6

3.1.2 -Serveur Web 6

3.1.2.1 -Caractéristiques techniques 6

3.1.3 -Serveur de Batches 6

3.1.4 -Serveur de Fichiers 6

3.2 -Bases de données 6

3.3 -Web-services 6

3.4 -Autres Ressources 6

4 -Procédure de déploiement 7

4.1 -Déploiement des Batches 7

4.1.1 -Artefacts 7

4.1.2 -Variables d'environnement 7

4.1.3 -Configuration 7

4.1.3.1 -Fichier xxx.yyy 8

4.1.3.2 -Fichier zzz.ttt 8

4.1.3.3 -Fichier ... 8

4.1.4 -Ressources 8

4.1.5 -Vérifications 8

4.2 -Déploiement de l'Application Web 9

4.2.1 -Artefacts 9

4.2.2 -Environnement de l’application web 9

4.2.2.1 -Variables d’environnement 9

4.2.3 -Répertoire de configuration applicatif 9

4.2.3.1 -Fichier xxx.yyy 9

4.2.4 -DataSources 9

4.2.5 -Ressources 10

4.2.6 -Vérifications 10

5 -Procédure de démarrage / arrêt 11

5.1 -Base de données 11

5.2 -Batches 11

5.3 -Application web 11

6 -Procédure de mise à jour 12

6.1 -Base de données 12

6.2 -Batches 12

6.3 -Application web 12

7 -Supervision/Monitoring 13

7.1 -Supervision de l’application web 13

8 -Procédure de sauvegarde et restauration 14

9 -Glossaire 15

# Versions

|  |  |  |  |
| --- | --- | --- | --- |
| Auteur | Date | Description | Version |
| Mathieu  Lemaire | 27/09/2025 | Création du document | 1.1 |
| Mathieu  Lemaire | 29/09/2025 | Modification du document sur les dernière parties | 1.2 |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Objet du document

Le présent document constitue le dossier d’exploitation de l'application OC Pizza pour l’équipe technique, les développeurs ainsi que les personnes qui maintiendront le système.

L’objectif du document est de présenter les informations dont l’équipe d’exploitation a besoin pour pouvoir assurer une exploitation du système et le suivis pour anticiper les problèmes éventuels sur le système.

## Références

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

1. **Projet 10 – Dossier de conception technique** : Dossier de conception technique de l'application
2. **Projet 10 – Dossier de conception fonctionnelle** : Dossier d’exploitation de l’application
3. **Projet 10 – PV de Livraison** : Procès-verbal de la livraison finale

# Pré-requis

## Système

### Serveur de Base de données

Serveur de base de données hébergeant est stocké sur un serveur dédié. Nous utiliserons une base de données PostgreSQL en version 17, pour une meilleure stabilité. Cette version sera couplée à des sauvegardes automatiques journalières.

#### Caractéristiques techniques

* vCPU de 4 à 8 coeurs
* RAM : 16Go (Minimum)
* Stockage : 200 à 500 Go SSD
* Réseau: 1Gbps

### Serveur Web

Serveur virtuel hébergeant l'application web. Nous utiliserons l’application Tomcat Embedded dans sa version 10 pour une meilleure stabilité.

#### Caractéristiques techniques

* vCPU de 4 à 8 coeurs
* RAM : 16Go (Minimum)
* Stockage : 200 à 500 Go SSD
* Réseau: 1Gbps

## Bases de données

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

* **Installation PostgreSQL 17 sur Linux:**

Utilisation de la commande : **sudo apt install postgresql postgresql-contrib -y**

Ensuite vérification de l’état de service : **sudo systemctl status postgresql**

Création de la base et les utilisateurs : **sudo -u postgres psql**

## Web-services

Les web services suivants doivent être accessibles et à jour :

* Installation du Web Service (Spring Boot et Tomcat Embedded) sur OVH

Prérequis :

* + - Serveur OVH (VPS ou dédié) sous Ubuntu 22
    - Java 21 installé
    - Maven installé
    - PostgreSQL configuré et en cours de fonctionnement

Installation Java 21 :

* Utilisation du script :

**sudo apt update**

**sudo apt install wget -y**

**wget https://github.com/adoptium/temurin21-binaries/releases/download/jdk-21.0.2+9/OpenJDK21U-jdk\_x64\_linux\_hotspot\_21.0.2\_9.tar.gz**

**sudo mkdir -p /opt/java**

**sudo tar -xzf OpenJDK21U-jdk\_x64\_linux\_hotspot\_21.0.2\_9.tar.gz -C /opt/java**

**export JAVA\_HOME=/opt/java/jdk-21.0.2+9**

**export PATH=$JAVA\_HOME/bin:$PATH**

**java -version**

Transfert backend Spring Boot avec la commande :

**mvn clean package**

Ensuite copier le .jar sur le serveur OVH :

**scp target/mon-appli.jar user@IP\_SERVEUR\_OVH:/opt/app/**

Lancement du Web Service :

**java -jar /opt/app/mon-appli.jar**

Pour tester le web service vous pouvez utiliser la commande par défaut **curl http://localhost:8080/actuator/health**

## Autres Ressources

# Procédure de déploiement

## Déploiement de Tomcat Embedded sur le serveur dédié

### Artefacts

Les batches de Tomcat Embedded sont construits sous la forme d’une archive ZIP contenant les répertoires :

* Conf : contient les fichiers de configuration
* Lib : contient les différentes API
* Logs : contient les logs générés par Tomcat
* Bin : contient les scripts .sh et .bat des différents batches
* Webapps : contient les applications déployées sur le serveur grâce au WAR
* BackupTomcat.bat : permet d’automatiser la sauvegarde des applications web déployées dans le répertoire webapps

Extraire l’archive apache-tomcat-10.1.42.zip dans le répertoire préalablement créé à la racine du serveur :

**C:\apache-tomcat-10.1.42**

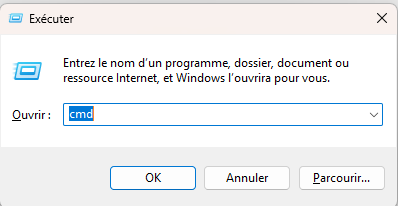
### Variables d'environnement

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

|  |  |  |
| --- | --- | --- |
| Nom | Obligatoire | Description |
| BATCH\_ENABLED | Non | Active ou désactives l’exécution des batches |
| BATCH\_SCHEDULE | Oui | Cron d’exécution (Toutes les heures) |
| BATCH\_MAX\_THREADS | Non | Nombre maximum de threads pour exécution parallèle |
| BATCH\_LOG\_PATH | Non | Chemin du fichier de logs spécifique aux batches |
| BATCH\_INPUT\_DIR | Oui | Répertoire source des fichiers traités |
| BATCH\_OUTPUT\_DIR | Oui | Répertoire de sortie des résultats |
| BATCH\_ERROR\_DIR | Non | Répertoire pour stocker les fichiers en erreur |
| BATCH\_NOTIFY\_EMAIL | Non | Adresse e-mail pour alertes en cas d’échec |

Définissez les variables d’environnement nécessaires comme ceci :

* Ouvrez un terminal de commande en tapant cmd dans la fenêtre exécuter de Windows (Touche Windows + R).



* Créer un utilisateur pour Tomcat :

**sudo useradd -m -U -d /opt/tomcat -s /bin/false tomcat**

* Télécharger Tomcat (Vérifier directement la version sur le site <https://tomcat.apache.org>) :

**cd /tmp**

**TOMCAT\_VER=10.1.30**

**wget** [**https://dlcdn.apache.org/tomcat/tomcat-10/v${TOMCAT\_VER}/bin/apache-tomcat-${TOMCAT\_VER}.tar.gz**](https://dlcdn.apache.org/tomcat/tomcat-10/v$%7bTOMCAT_VER%7d/bin/apache-tomcat-$%7bTOMCAT_VER%7d.tar.gz)

* Installer Tomcat dans /opt/tomcat :

**sudo mkdir -p /opt/tomcat**

**sudo tar xzvf apache-tomcat-${TOMCAT\_VER}.tar.gz -C /opt/tomcat --strip-components=1**

* Ensuite modifier les permissions de Tomcat :

**sudo chown -R tomcat:tomcat /opt/tomcat**

**sudo chmod +x /opt/tomcat/bin/\*.sh**

* Configurer le systemd pour démarrer Tomcat automatiquement :

Créer **/etc/systemd/system/tomcat.service** avec ce contenu :

[Unit]

Description=Apache Tomcat Web Application Container

After=network.target

[Service]

Type=forking

User=tomcat

Group=tomcat

Environment="JAVA\_HOME=/usr/lib/jvm/java-17-openjdk-amd64"

Environment="CATALINA\_PID=/opt/tomcat/temp/tomcat.pid"

Environment="CATALINA\_HOME=/opt/tomcat"

Environment="CATALINA\_BASE=/opt/tomcat"

Environment="CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

Environment="JAVA\_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom"

ExecStart=/opt/tomcat/bin/startup.sh

ExecStop=/opt/tomcat/bin/shutdown.sh

Restart=on-failure

RestartSec=10

[Install]

WantedBy=multi-user.target

Puis verifier que Tomcat est bien actif :

**sudo systemctl daemon-reload**

**sudo systemctl enable tomcat**

**sudo systemctl start tomcat**

**sudo systemctl status tomcat**

### Configuration

Voici les différents fichiers de configuration :

* **Serveur.xml :** Fichier principal de configuration de Tomcat
* **Web.xml :** Fichier de configuration global pour toutes les applications
* **Context.xml :** Fichier qui définit par défaut la configuration des applications web
* **Tomcat-users.xml :** Fichier de configuration des utilisateurs et des rôles
* **Logging.properties :** Fichier de gestion du niveau des logs
* **Catalina.properties :** Fichier qui définit les propriétés système de Catalina (Cœur de Tomcat)

#### Fichier Serveur.xml

Le rôle de ce fichier est de définir le serveur, les services, les connecteurs réseau. C’est le fichier le plus important.

#### Fichier Web.xml

Le rôle de ce fichier est de définir les paramètres de déploiement pour toutes les applications.

#### Fichier Context.xml

Le rôle de ce fichier est de définir la configuration par défaut de toutes les applications web.

#### Tomcat-users.xml

Le rôle de ce fichier est de définir les utilisateurs et les rôles pour accéder à l’interface web de gestion.

#### Logging.properties

Le rôle de ce fichier est de définir les différents fichiers de logs ainsi que sélectionner le niveau de criticité (INFO, WARNING, SEVERE).

#### Catalina.properties

Le rôle de ce fichier est de définir les propriétés système utilisées par Catalina (le cœur de Tomcat). Ce fichier le classpath, la liste des packages autorisés/interdits et les ressources communes.

### Ressources

Pour faire fonctionner les applications web, Catalina a besoin que le Jdk ou le Jre Java soit installé sur le serveur hébergeant Tomcat. Vous pouvez télécharger cette ressource avec le lien suivant :

**https://www.oracle.com/technetwork/java/javase/downloads/index.html**

### Vérifications

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

## Déploiement de PostgreSQL sur le serveur

### Artefacts

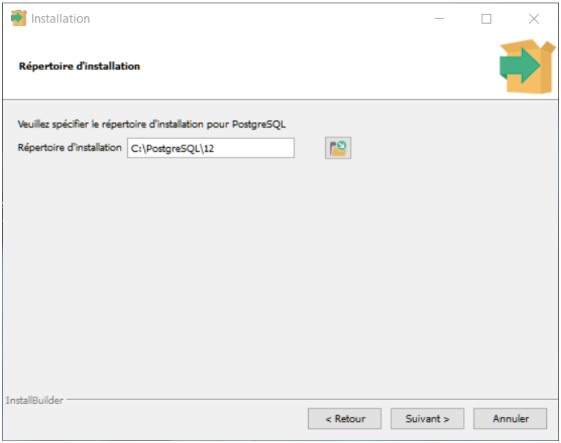
Les batches de PostgreSQL se présentent sous la forme d’une archive zip contenant certains fichiers :

* Postgresql-12.0.1-windows-x64.exe : permet de procéder à une installation simple et rapide
* OCPizzaBackup.bat : permet d’automatiser quotidiennement les sauvegardes de la base de données
* Script\_bdd\_ocpizza.sql : permet de créer l’ensemble des tables de la base de données
* Script\_depart\_bdd.sql : permet l’insertion des premières données nécessaires au démarrage de l’activité

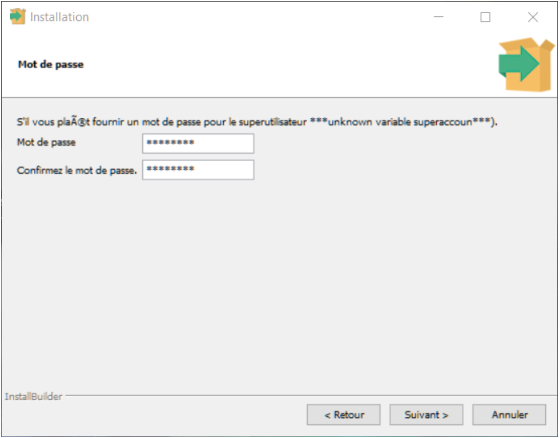
Lancement de l’installateur de postgreSQL



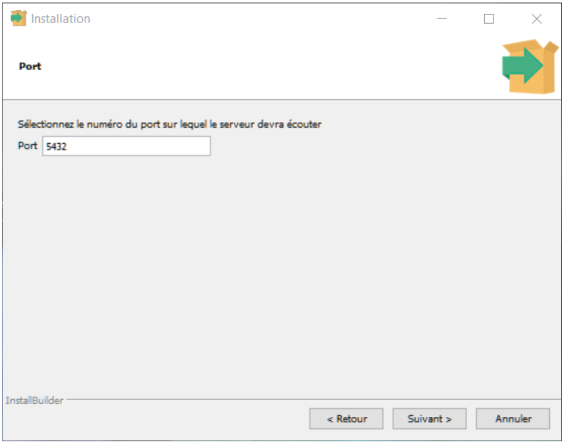
* Ensuite renseignez le chemin de destination de PostgreSQL (la version change en fonction de ce que vous avez sélectionné) :



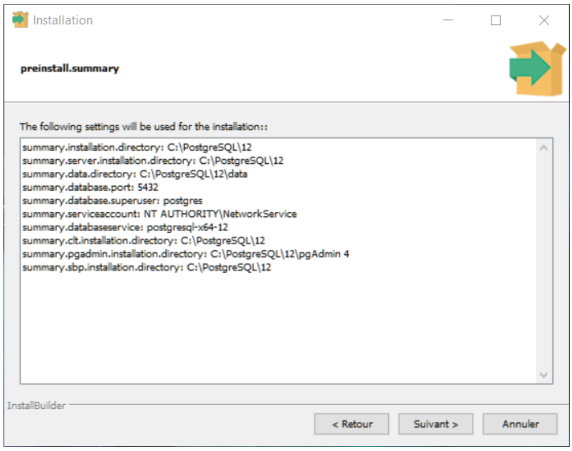
* Suivez l’installation sans modifier ce qui est proposé et ensuite renseignez le mot de passe pour le Super-Utilisateur et l’administration de PostgreSQL :



* Renseignez le port d’écoute pour le serveur PostgreSQL :



* Suivre l’installation et pour finir, vérifiez lors du résumé que l’ensemble des informations saisies sont correctes :



### Configuration

Vous trouverez les fichiers permettant de configurer la base de données dans le répertoire suivant :

**C:\PostgreSQL\12\data**

#### pg\_hba.conf

Ce fichier permet de contrôler quels hôtes peuvent se connecter, comment les clients sont authentifiés et de vérifier à quelles bases de données les utilisateurs peuvent avoir accès.

#### pg\_ident.conf

Ce fichier permet le contrôle du mappage des noms d’utilisateur PostgreSQL.

#### postgresql.conf

Ce fichier permet de configurer de nombreux paramètres comme les connexions et les authentifications auprès du serveur, l’usage des ressources du serveur ainsi que le paramétrage des logs générés.

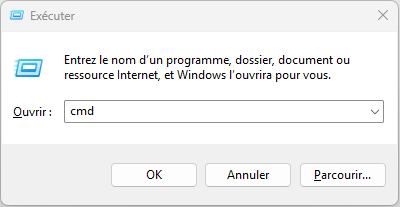
### Création de la base de données

Vous pouvez utiliser les deux scripts SQL évoqué précédemment dans le dossier :

**C:\PostgreSQL\12\data**

Vous pouvez configurer et initialiser la base de données en réalisant les étapes suivantes :

* Ouvrez un terminal en tapant cmd dans la fenêtre exécuter Windows



* Utiliser la commande pour accéder au répertoire bin du dossier d’installation de PostgreSQL :

**cd PostgreSQL\12\bin**

* Pour démarrer le serveur, entrez la commande suivante :

**pg\_ctl -D « C:\PostgreSQL\12\data » start**

* Vous pouvez maintenant créer la base de données avec la commande suivante et ensuite validez avec le mot de passe administrateur :

**createdb -U postgres ocpizza**

La base donnée nommée « ocpizza » est maintenant créée sur le serveur PostgreSQL.

### Insertion de données

* Entrez la commande suivante afin de créer les tables dans la base OC Pizza :

**psql -U postgres -d ocpizza -f C:\PostgreSQL\12\data\script\_creation\_bdd\_ocpizza.sql**

Réalisez la même commande avec le second script et ensuite validez de nouveau avec le mot de passe administrateur.

### Vérifications

Afin de vérifier que la base est correctement déployée, vous pouvez saisir les commandes suivantes :

* Démarrez l’utilitaire psql avec l’utilisateur postgres :

**psql -U postgres**

* Saisissez la commande afin de vous connecter à la base de données :

**\c ocpizza**

* Vous pouvez saisir la commande suivante pour voir le détail des tables :

**\dt**

### Mise en place des sauvegardes automatisées de la base de données

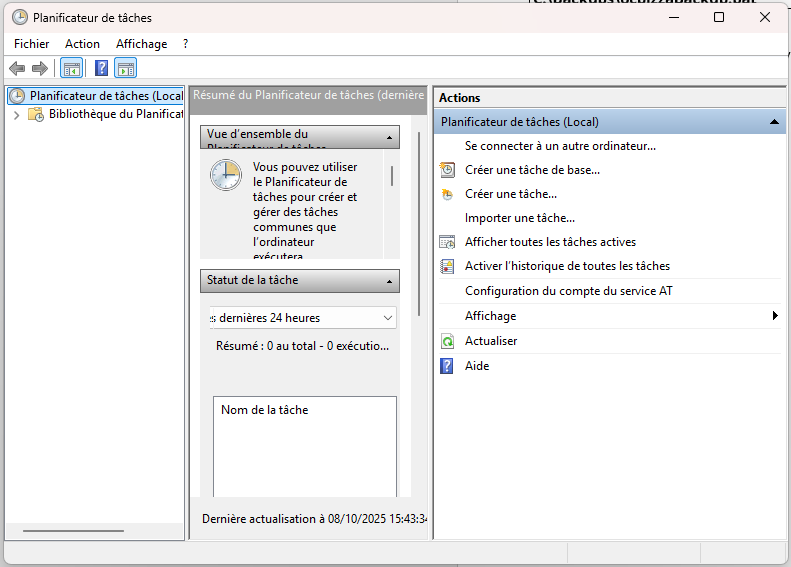
* Pour commencer créez un nouveau dossier nommé Backups à la racine du serveur.
* Dans ce dossier, créez un nouveau dossier nommé PostgreSQL.
* Ensuite déposez le fichier ocpizzaBackup.bat dans ce dossier :

**C:\Backups\ocpizzaBackup.bat**

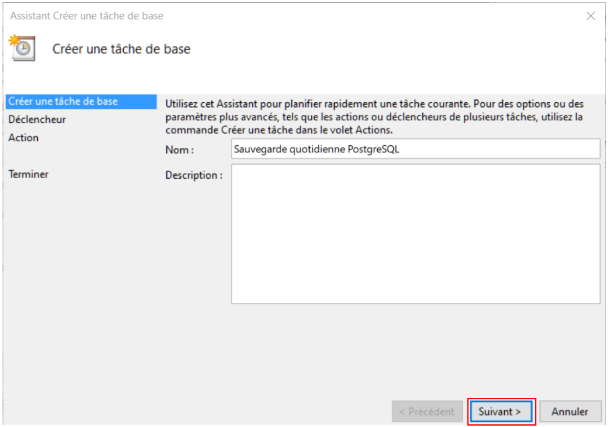
* Pour automatiser la tâche, ouvrez une fenêtre cmd et tapez :

**Taskschd.msc**

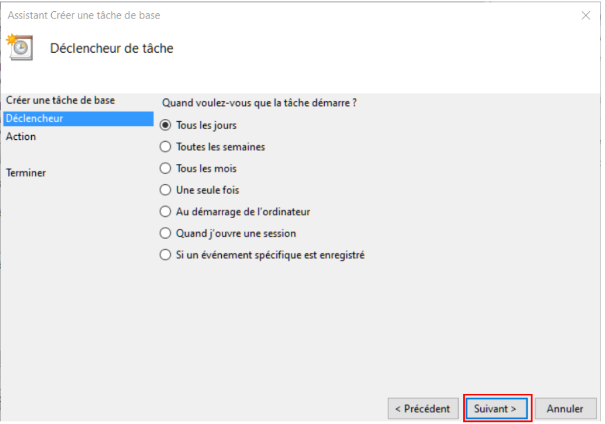
* Sur cette fenêtre vous pouvez sélectionner l’onglet « Créer une tâche » :

****

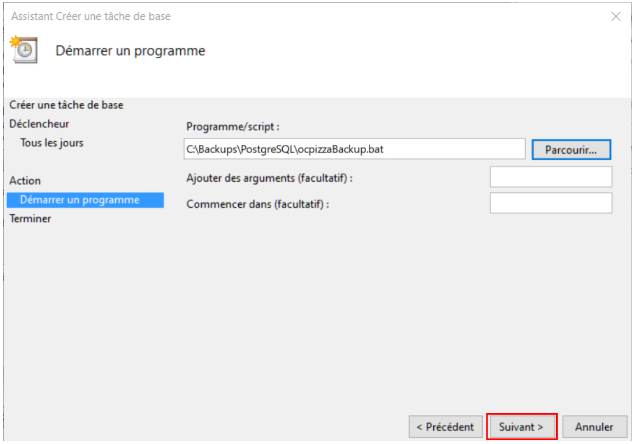
* Créer une tâche de sauvegarde :



* Précisez la fréquence de sauvegarde souhaité :

****

* Sélectionnez le fichier ocpizzaBackup.bat :

****

* Cliquez sur terminer pour enregistrer la tâche. Maintenant les sauvegardes quotidiennes sont actives.

# Procédure de démarrage / arrêt

## Serveur d’application Tomcat

* Démarrage de Tomcat :

Pour démarrer le serveur, vous devez utiliser la commande « **startup** » dans l’emplacement du répertoire de l’installation de Tomcat, plus précisément dans le dossier bin.

* Arrêt de Tomcat :

Pour arrêter le serveur, exécutez la commande « **shutdown** », toujours dans l’emplacement de l’installation de Tomcat et dans le dossier bin.

## Base de données PostgreSQL

* Démarrage du serveur Postgre :

Pour démarrer le serveur PostgreSQL, il faut exécutez la commande suivante dans le répertoire d’installation PostgreSQL

**pg\_ctl -D ”C:\PostgreSQL\12\data” start**

* Arrêt du serveur PostgreSQL:

Pour arrêter le serveur, il faut exécuter la commande suivante dans le répertoire d’installation de PostgreSQL

**pg\_ctl -D ”C:\PostgreSQL\12\data” stop**

# Procédure de mise à jour

## Base de données

* Installation de la nouvelle version de PostgreSQL

Pour vous aider dans l’installation de la nouvelle version, nous vous recommandons d’installer PostgreSQL via l’installateur Windows et suivre la section de l’installation de Postgre.

Une fois la sauvegarde effectuée, il vous faudra installer la nouvelle version de PostgreSQL dans le répertoire suivant :

**C:\** **PostgreSQL\<NUM\_VERSION>**

Indiquer un port différent de celui mit dans l’installation de départ :

**C:\** **PostgreSQL\<NUM\_VERSION>\data**

Indiquer le même mot de passe administrateur, l’utilisateur étant déjà créé.

## Application web

Pour la mise à jour des applications Web, Tomcat permet grâce à sa fonctionnalité de déploiement parallèle de facilement déployer la mise à jour d’une application web en adoptant une convention de nommage :

**NOM\_APP##<NUM\_VERSION>.war**

Tomcat identifiera automatiquement la version de l’application déployée et procédera au remplacement de celle-ci en cas d’incrémentation du numéro de version de l’application.

# Supervision/Monitoring

## Base de données

Afin de tester que l’application web est toujours fonctionnelles, faire ceci…

# Procédure de sauvegarde et restauration

# Glossaire

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