

Guide de Déploiement - Générateur de CIU WhatsApp

Health Equity & Resilience Observatory (HERO), UBC

January 2026

Table des matières

1 Guide de Déploiement Windows Server - Générateur de CIU WhatsApp	2
1.1 Note importante	2
1.2 Table des matières	3
1.3 Prérequis Windows	3
1.3.1 Matériel serveur recommandé	3
1.3.2 Système d'exploitation	3
1.3.3 Logiciels requis (nous les installerons)	3
1.4 Préparation du serveur	4
1.4.1 Étape 1 : Activer l'accès Bureau à distance (si nécessaire)	4
1.4.2 Étape 2 : Configurer le pare-feu Windows	4
1.4.3 Étape 3 : Mettre à jour Windows	4
1.5 Installation de Python	4
1.5.1 Étape 1 : Télécharger Python 3.12	4
1.5.2 Étape 2 : Installer Python	4
1.5.3 Étape 3 : Vérifier l'installation	5
1.5.4 Étape 4 : Installer Git for Windows	5
1.6 Installation de PostgreSQL	5
1.6.1 Étape 1 : Télécharger PostgreSQL	5
1.6.2 Étape 2 : Installer PostgreSQL	5
1.6.3 Étape 3 : Créer la base de données pour l'application	6
1.6.4 Étape 4 : Créer un utilisateur pour l'application	6
1.6.5 Étape 5 : Donner les permissions	6
1.7 Installation de l'application	6
1.7.1 Étape 1 : Créer un dossier pour l'application	6
1.7.2 Étape 2 : Cloner le dépôt GitHub	7
1.7.3 Étape 3 : Créer l'environnement virtuel Python	7
1.7.4 Étape 4 : Installer les dépendances	7
1.7.5 Étape 5 : Générer le sel de sécurité	7
1.7.6 Étape 6 : Configurer le fichier .env	7
1.7.7 Étape 7 : Initialiser la base de données	9
1.7.8 Étape 8 : Tester localement	9
1.8 Configuration IIS (Serveur Web)	9

1.8.1	Étape 1 : Installer IIS	9
1.8.2	Étape 2 : Installer URL Rewrite et ARR	9
1.8.3	Étape 3 : Activer le proxy dans ARR	9
1.8.4	Étape 4 : Créer un site IIS	10
1.8.5	Étape 5 : Configurer le reverse proxy	10
1.9	Configuration du service Windows	10
1.9.1	Étape 1 : Installer NSSM (Non-Sucking Service Manager)	10
1.9.2	Étape 2 : Créer le service	10
1.9.3	Étape 3 : Démarrer le service	11
1.10	Configuration SSL	11
1.10.1	Option 1 : Certificat Let's Encrypt (Gratuit, Recommandé)	11
1.10.2	Option 2 : Certificat acheté	11
1.11	Configuration Twilio	12
1.12	Test et validation	12
1.12.1	Test 1 : Vérifier le service Windows	12
1.12.2	Test 2 : Tester l'endpoint de santé	12
1.12.3	Test 3 : Vérifier IIS	12
1.12.4	Test 4 : Test WhatsApp complet	13
1.13	Maintenance Windows	13
1.13.1	Vérifier les logs	13
1.13.2	Redémarrer le service	13
1.13.3	Sauvegarde de la base de données	13
1.13.4	Automatiser les sauvegardes avec Task Scheduler	13
1.14	Dépannage Windows	14
1.14.1	Problème : Le service ne démarre pas	14
1.14.2	Problème : Erreur 500 dans IIS	14
1.14.3	Problème : Erreur de politique d'exécution PowerShell	15
1.14.4	Problème : PostgreSQL ne démarre pas	15
1.14.5	Problème : Port 8000 déjà utilisé	15
1.15	Différences clés Windows vs Linux	15
1.16	Recommandations finales	16
1.16.1	Si vous choisissez Windows :	16
1.16.2	Si vous pouvez choisir Linux :	16
1.17	Support	16

1 Guide de Déploiement Windows Server - Générateur de CIU WhatsApp

Guide complet pour le déploiement sur Windows Server

1.1 Note importante

Ce guide est conçu pour **Windows Server 2019 ou 2022**. Si vous utilisez un serveur Linux, veuillez consulter `README_FR.md` à la place, qui contient des instructions complètes pour Ubuntu/Linux.

Recommandation : Linux (Ubuntu) est généralement préféré pour ce type d'application en production car il est plus stable, plus sécurisé, et plus facile à maintenir pour les applications web. Cependant, Windows Server fonctionne également très bien si c'est ce qui est disponible dans votre infrastructure.

1.2 Table des matières

1. Prérequis Windows
 2. Préparation du serveur
 3. Installation de Python
 4. Installation de PostgreSQL
 5. Installation de l'application
 6. Configuration IIS (Serveur Web)
 7. Configuration du service Windows
 8. Configuration SSL
 9. Configuration Twilio
 10. Test et validation
 11. Maintenance Windows
 12. Dépannage Windows
-

1.3 Prérequis Windows

1.3.1 Matériel serveur recommandé

- **CPU** : 2 coeurs minimum (4 coeurs recommandés)
- **RAM** : 4 GB minimum (8 GB recommandés pour Windows)
- **Stockage** : 40 GB minimum (SSD préféré)
- **Réseau** : Connexion stable avec IP statique

1.3.2 Système d'exploitation

- **Windows Server 2019** ou **Windows Server 2022** (Standard ou Datacenter)
- Accès administrateur complet
- Windows Update à jour

1.3.3 Logiciels requis (nous les installerons)

- Python 3.12+
 - PostgreSQL 14+
 - Git for Windows
 - IIS (Internet Information Services)
 - URL Rewrite pour IIS
 - Application Request Routing (ARR) pour IIS
-

1.4 Préparation du serveur

1.4.1 Étape 1 : Activer l'accès Bureau à distance (si nécessaire)

1. Ouvrez **Gestionnaire de serveur** (Server Manager)
2. Cliquez sur **Serveur local** dans le menu gauche
3. Trouvez **Bureau à distance** et cliquez sur **Désactivé**
4. Sélectionnez **Autoriser les connexions à distance à cet ordinateur**
5. Cliquez sur **OK**

1.4.2 Étape 2 : Configurer le pare-feu Windows

1. Ouvrez **Windows Defender Firewall with Advanced Security**
2. Créez les règles entrantes pour :

Pour HTTP (port 80) :

- Clic droit sur "Règles de trafic entrant" → Nouvelle règle
- Type : Port
- Port TCP spécifique : 80
- Autoriser la connexion
- Profils : Tous
- Nom : "WhatsApp UIC - HTTP"

Pour HTTPS (port 443) :

- Même procédure mais pour le port 443
- Nom : "WhatsApp UIC - HTTPS"

1.4.3 Étape 3 : Mettre à jour Windows

1. Ouvrez **Paramètres** → **Mise à jour et sécurité**
 2. Cliquez sur **Rechercher des mises à jour**
 3. Installez toutes les mises à jour disponibles
 4. Redémarrez si nécessaire
-

1.5 Installation de Python

1.5.1 Étape 1 : Télécharger Python 3.12

1. Ouvrez un navigateur et allez sur : <https://www.python.org/downloads/windows/>
2. Téléchargez **Python 3.12.x Windows installer (64-bit)**
3. Enregistrez le fichier (ex : python-3.12.1-amd64.exe)

1.5.2 Étape 2 : Installer Python

1. Exécutez l'installateur en tant qu'administrateur (clic droit → **Exécuter en tant qu'administrateur**)
2. **IMPORTANT** : Cochez "**Add Python 3.12 to PATH**" en bas de la fenêtre

3. Cliquez sur “Install Now”
4. Attendez la fin de l’installation
5. Cliquez sur “Close”

1.5.3 Étape 3 : Vérifier l’installation

1. Ouvrez **PowerShell** (clic droit → **Exécuter en tant qu’administrateur**)
2. Tapez :

```
python --version
```

Vous devriez voir : Python 3.12.x

3. Vérifiez pip :

```
pip --version
```

1.5.4 Étape 4 : Installer Git for Windows

1. Téléchargez Git depuis : <https://git-scm.com/download/win>
2. Exécutez l’installateur
3. Utilisez les paramètres par défaut (cliquez sur **Next** à chaque étape)
4. Vérifiez l’installation :

```
git --version
```

1.6 Installation de PostgreSQL

1.6.1 Étape 1 : Télécharger PostgreSQL

1. Allez sur : <https://www.postgresql.org/download/windows/>
2. Cliquez sur “Download the installer”
3. Téléchargez **PostgreSQL 14.x Windows x86-64**

1.6.2 Étape 2 : Installer PostgreSQL

1. Exécutez l’installateur en tant qu’administrateur
2. Suivez les étapes :
 - **Installation Directory** : Laissez par défaut (C:\Program Files\PostgreSQL\14)
 - **Select Components** : Cochez tout
 - **Data Directory** : Laissez par défaut
 - **Password** : **IMPORTANT** - Créez un mot de passe fort pour l’utilisateur **postgres**
 - Notez ce mot de passe, vous en aurez besoin !
 - **Port** : Laissez 5432
 - **Locale** : Laissez par défaut (ou choisissez French si disponible)
3. Cliquez sur **Next** puis **Finish**

** PLACEHOLDER - À REMPLIR PAR L’ÉQUIPE CANADIENNE :**

POSTGRES_ADMIN_PASSWORD=" [À COMPLÉTER PAR L'ÉQUIPE CANADIENNE]"

1.6.3 Étape 3 : Créer la base de données pour l'application

1. Ouvrez **pgAdmin 4** depuis le menu Démarrer
2. Connectez-vous avec le mot de passe créé ci-dessus
3. Cliquez droit sur “**Databases**” → **Create** → **Database**
4. Remplissez :
 - **Database** : **whatsapp_uic_db**
 - **Owner** : **postgres** (pour l'instant)
5. Cliquez sur **Save**

1.6.4 Étape 4 : Créer un utilisateur pour l'application

1. Dans pgAdmin, cliquez droit sur “**Login/Group Roles**” → **Create** → **Login/Group Role**
2. Onglet **General** :
 - **Name** : **whatsapp_bot**
3. Onglet **Definition** :
 - **Password** : Créez un mot de passe fort
4. Onglet **Privileges** :
 - Cochez **Can login?**
5. Cliquez sur **Save**

** PLACEHOLDER - À REMPLIR PAR L'ÉQUIPE CANADIENNE :**

DB_PASSWORD=" [À COMPLÉTER PAR L'ÉQUIPE CANADIENNE - Mot de passe pour whatsapp_bot]"

1.6.5 Étape 5 : Donner les permissions

1. Cliquez droit sur la base **whatsapp_uic_db** → **Properties**
 2. Onglet **Security**
 3. Cliquez sur + pour ajouter
 4. Sélectionnez **whatsapp_bot**
 5. Cochez tous les priviléges
 6. Cliquez sur **Save**
-

1.7 Installation de l'application

1.7.1 Étape 1 : Créer un dossier pour l'application

1. Ouvrez **PowerShell en tant qu'administrateur**
2. Créez le dossier :

```
# Créer le dossier principal  
New-Item -Path "C:\Applications" -ItemType Directory -Force
```

```
# Se déplacer dans ce dossier  
cd C:\Applications
```

1.7.2 Étape 2 : Cloner le dépôt GitHub

```
# Cloner le dépôt  
git clone https://github.com/drjforrest/whatsapp-uic-generator.git
```

```
# Entrer dans le dossier  
cd whatsapp-uic-generator
```

1.7.3 Étape 3 : Créer l'environnement virtuel Python

```
# Créer l'environnement virtuel  
python -m venv .venv
```

```
# Activer l'environnement virtuel  
.venv\Scripts\Activate.ps1
```

Note : Si vous voyez une erreur de politique d'exécution, exécutez :

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

Puis réessayez d'activer l'environnement virtuel.

1.7.4 Étape 4 : Installer les dépendances

```
# Mettre à jour pip  
python -m pip install --upgrade pip
```

```
# Installer l'application  
pip install -e .
```

```
# Vérifier  
python -c "import fastapi; print('FastAPI installé avec succès')"
```

1.7.5 Étape 5 : Générer le sel de sécurité

```
# Générer le sel  
python scripts\generate_salt.py
```

Copiez le sel généré – vous en aurez besoin pour le fichier .env.

** PLACEHOLDER - À REMPLIR PAR L'ÉQUIPE CANADIENNE :**

UIC_SALT=" [À COMPLÉTER PAR L'ÉQUIPE CANADIENNE - Utilisez generate_salt.py]"

1.7.6 Étape 6 : Configurer le fichier .env

```
# Copier le fichier d'exemple  
Copy-Item .env.example .env
```

```

# Éditer avec Notepad
notepad .env

Remplissez le fichier .env avec ces valeurs :

# =====
# CONFIGURATION DE L'APPLICATION
# =====

APP_NAME="Générateur de CIU WhatsApp"
APP_VERSION="0.1.0"
ENVIRONMENT=production
DEBUG=False

# =====
# SÉCURITÉ
# =====

UIC_SALT="[VOTRE SEL GÉNÉRÉ]"

# =====
# BASE DE DONNÉES - WINDOWS
# =====

# Format Windows pour PostgreSQL
DATABASE_URL="postgresql://whatsapp_bot:[DB_PASSWORD]@localhost:5432/whatsapp_uic_db"

# Remplacez [DB_PASSWORD] par le mot de passe créé pour whatsapp_bot

# =====
# CONFIGURATION TWILIO
# =====

TWILIO_ACCOUNT_SID="[À COMPLÉTER PAR L'ÉQUIPE CANADIENNE]"
TWILIO_AUTH_TOKEN="[À COMPLÉTER PAR L'ÉQUIPE CANADIENNE]"
TWILIO_WHATSAPP_NUMBER="[À COMPLÉTER PAR L'ÉQUIPE CANADIENNE]"

# =====
# AUTRES PARAMÈTRES
# =====

WEBHOOK_PATH="/whatsapp/webhook"
SESSION_TIMEOUT_MINUTES=15
LOG_LEVEL=INFO
LOG_JSON=True

```

Sauvegardez et fermez Notepad.

1.7.7 Étape 7 : Initialiser la base de données

```
# Initialiser la base de données
python scripts\init_db.py
```

Vous devriez voir : Tables de base de données créées avec succès !

1.7.8 Étape 8 : Tester localement

```
# Démarrer le serveur pour tester
python -m uvicorn app.main:app --host 0.0.0.0 --port 8000
```

Dans un autre PowerShell :

```
# Tester l'endpoint de santé
Invoke-WebRequest -Uri http://localhost:8000/health
```

Si ça fonctionne, arrêtez le serveur avec CTRL + C.

1.8 Configuration IIS (Serveur Web)

1.8.1 Étape 1 : Installer IIS

1. Ouvrez **Gestionnaire de serveur**
2. Cliquez sur **Gérer → Ajouter des rôles et fonctionnalités**
3. Cliquez sur **Suivant** jusqu'à **Rôles de serveurs**
4. Cochez **Serveur Web (IIS)**
5. Cliquez sur **Ajouter des fonctionnalités** quand demandé
6. Cliquez sur **Suivant** puis **Installer**
7. Attendez la fin de l'installation

1.8.2 Étape 2 : Installer URL Rewrite et ARR

1. Téléchargez **URL Rewrite** depuis : <https://www.iis.net/downloads/microsoft/url-rewrite>
2. Installez-le
3. Téléchargez **Application Request Routing (ARR)** depuis : <https://www.iis.net/downloads/microsoft/application-request-routing>
4. Installez-le

1.8.3 Étape 3 : Activer le proxy dans ARR

1. Ouvrez **Internet Information Services (IIS) Manager**
2. Cliquez sur le nom de votre serveur dans l'arborescence gauche
3. Double-cliquez sur **Application Request Routing Cache**
4. Dans le panneau de droite, cliquez sur **Server Proxy Settings**
5. Cochez **Enable proxy**
6. Cliquez sur **Apply**

1.8.4 Étape 4 : Créer un site IIS

1. Dans IIS Manager, cliquez droit sur **Sites** → **Add Website**
2. Remplissez :
 - **Site name** : WhatsAppUIC
 - **Physical path** : C:\inetpub\wwwroot\whatsappuic (créez ce dossier si nécessaire)
 - **Binding** :
 - Type : http
 - Port : 80
 - Host name : Votre nom de domaine (ex : whatsapp.votre-org.cd)
3. Cliquez sur **OK**

** PLACEHOLDER - À remplir par l'équipe canadienne :**

NOM_DE_DOMAINE=" [À compléter par l'équipe canadienne - ex: whatsapp.votre-org.cd]"

1.8.5 Étape 5 : Configurer le reverse proxy

1. Dans IIS Manager, sélectionnez votre site **WhatsAppUIC**
 2. Double-cliquez sur **URL Rewrite**
 3. Dans le panneau de droite, cliquez sur **Add Rule(s)**
 4. Sélectionnez **Reverse Proxy**
 5. Si demandé d'activer le proxy, cliquez sur **OK**
 6. Dans **Inbound Rules** :
 - Server name or IP : localhost:8000
 7. Cliquez sur **OK**
-

1.9 Configuration du service Windows

Pour que l'application démarre automatiquement, nous allons créer un service Windows.

1.9.1 Étape 1 : Installer NSSM (Non-Sucking Service Manager)

1. Téléchargez NSSM depuis : <https://nssm.cc/download>
2. Extrayez le ZIP
3. Copiez nssm.exe (version 64-bit) dans C:\Windows\System32

1.9.2 Étape 2 : Crée le service

```
# Ouvrir PowerShell en tant qu'administrateur
# Naviguer vers le dossier de l'application
cd C:\Applications\whatsapp-uic-generator

# Crée le service avec NSSM
nssm install WhatsAppUICService
```

Une fenêtre s'ouvrira. Remplissez :

Onglet Application : - Path : C:\Applications\whatsapp-uic-generator\.venv\Scripts\python.exe
- Startup directory : C:\Applications\whatsapp-uic-generator - Arguments : -m uvicorn
app.main:app --host 0.0.0.0 --port 8000 --workers 2

Onglet Details : - Display name : WhatsApp UIC Generator Service - Description : Service pour le générateur de CIU WhatsApp - Startup type : Automatic

Onglet Log on : - Laissez Local System account

Cliquez sur **Install service**.

1.9.3 Étape 3 : Démarrer le service

```
# Démarrer le service
nssm start WhatsAppUICService
```

```
# Vérifier le statut
nssm status WhatsAppUICService
```

Ou via l'interface graphique : 1. Ouvrez **Services** (services.msc) 2. Trouvez **WhatsApp UIC Generator Service** 3. Clic droit → **Démarrer**

1.10 Configuration SSL

1.10.1 Option 1 : Certificat Let's Encrypt (Gratuit, Recommandé)

Utiliser **Win-ACME** (anciennement LetsEncrypt-Win-Simple) :

1. Téléchargez Win-ACME depuis : <https://www.win-acme.com/>
2. Extrayez dans C:\Tools\win-acme
3. Ouvrez PowerShell en administrateur :

```
cd C:\Tools\win-acme
.\wacs.exe
```

4. Suivez le menu interactif :
 - Choisissez **N** pour créer un nouveau certificat
 - Choisissez **1** pour Single binding of an IIS site
 - Sélectionnez votre site **WhatsAppUIC**
 - Entrez votre adresse email
 - Acceptez les conditions

Win-ACME installera automatiquement le certificat dans IIS.

1.10.2 Option 2 : Certificat acheté

Si vous avez acheté un certificat SSL :

1. Dans IIS Manager, cliquez sur le nom de votre serveur

2. Double-cliquez sur **Server Certificates**
3. Dans le panneau de droite, cliquez sur **Import**
4. Parcourez et sélectionnez votre fichier **.pfx**
5. Entrez le mot de passe du certificat
6. Cliquez sur **OK**

Puis configurez le binding HTTPS :

1. Cliquez droit sur votre site **WhatsAppUIC** → **Edit Bindings**
 2. Cliquez sur **Add**
 3. Type : **https**
 4. Port : **443**
 5. Host name : Votre domaine
 6. SSL certificate : Sélectionnez votre certificat
 7. Cliquez sur **OK**
-

1.11 Configuration Twilio

Consultez la section correspondante dans `README_FR.md` - les étapes sont identiques pour Windows.

Une fois votre webhook configuré, l'URL sera :

`https://votre-domaine.cd/whatsapp/webhook`

1.12 Test et validation

1.12.1 Test 1 : Vérifier le service Windows

```
# Vérifier le statut
Get-Service | Where-Object {$_.Name -like "*WhatsApp*"}  
  
# Ou
nssm status WhatsAppUICService
```

1.12.2 Test 2 : Tester l'endpoint de santé

```
# Via localhost
Invoke-WebRequest -Uri http://localhost:8000/health  
  
# Via le domaine (après configuration DNS)
Invoke-WebRequest -Uri https://votre-domaine.cd/health
```

1.12.3 Test 3 : Vérifier IIS

1. Ouvrez un navigateur
2. Allez sur `https://votre-domaine.cd/health`

3. Vous devriez voir le JSON de santé

1.12.4 Test 4 : Test WhatsApp complet

Suivez les mêmes étapes que dans `README_FR.md` section “Test et validation”.

1.13 Maintenance Windows

1.13.1 Vérifier les logs

```
# Logs du service (si configurés)
Get-Content C:\Applications\whatsapp-uic-generator\logs\app.log -Tail 50

# Logs IIS
Get-Content C:\inetpub\logs\LogFiles\W3SVC1\*.log -Tail 100
```

1.13.2 Redémarrer le service

```
# Via NSSM
nssm restart WhatsAppUICService

# Via PowerShell
Restart-Service -Name "WhatsAppUICService"

# Via Services.msc
# Services → Clic droit sur le service → Redémarrer
```

1.13.3 Sauvegarde de la base de données

```
# Créer un dossier de sauvegarde
New-Item -Path "C:\Backups" -ItemType Directory -Force

# Sauvegarder avec pg_dump (ajuster le chemin)
$date = Get-Date -Format "yyyyMMdd"
& "C:\Program Files\PostgreSQL\14\bin\pg_dump.exe" -U whatsapp_bot -h localhost whatsapp_uic_db
```

1.13.4 Automatiser les sauvegardes avec Task Scheduler

1. Ouvrez **Task Scheduler** (`taskschd.msc`)
2. Cliquez sur **Create Basic Task**
3. Name : **WhatsApp UIC Backup**
4. Trigger : **Daily** à 2h du matin
5. Action : **Start a program**
6. Program : `powershell.exe`
7. Arguments :

```
-File C:\Applications\whatsapp-uic-generator\scripts\backup.ps1
```

Créez le script backup.ps1 :

```
# C:\Applications\whatsapp-uic-generator\scripts\backup.ps1
$date = Get-Date -Format "yyyyMMdd"
$backupPath = "C:\Backups\backup-$date.sql"

# Sauvegarder
& "C:\Program Files\PostgreSQL\14\bin\pg_dump.exe" -U whatsapp_bot -h localhost whatsapp_uic_db

# Compresser
Compress-Archive -Path $backupPath -DestinationPath "$backupPath.zip"

# Supprimer le .sql non compressé
Remove-Item $backupPath

# Nettoyer les anciennes sauvegardes (garder 30 jours)
Get-ChildItem "C:\Backups\backup-*.*" | Where-Object {$_['].LastWriteTime -lt (Get-Date).AddDays(-30)}
```

1.14 Dépannage Windows

1.14.1 Problème : Le service ne démarre pas

Solutions :

1. Vérifier les logs NSSM :

```
nssm status WhatsAppUICService
```

2. Vérifier manuellement :

```
cd C:\Applications\whatsapp-uic-generator
.\.venv\Scripts\Activate.ps1
python -m uvicorn app.main:app --host 0.0.0.0 --port 8000
```

3. Vérifier les permissions du dossier

4. Vérifier que PostgreSQL est démarré :

```
Get-Service postgresql*
```

1.14.2 Problème : Erreur 500 dans IIS

Solutions :

1. Vérifier que le service Python tourne
2. Vérifier les logs IIS dans C:\inetpub\logs\LogFiles
3. Tester le proxy :

```
Invoke-WebRequest -Uri http://localhost:8000/health
```

1.14.3 Problème : Erreur de politique d'exécution PowerShell

Solution :

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

1.14.4 Problème : PostgreSQL ne démarre pas

Solutions :

1. Vérifier les services :

```
Get-Service postgresql*
Start-Service postgresql-x64-14
```

2. Vérifier les logs PostgreSQL dans :

```
C:\Program Files\PostgreSQL\14\data\log
```

1.14.5 Problème : Port 8000 déjà utilisé

Solution :

1. Trouver le processus :

```
Get-NetTCPConnection -LocalPort 8000 | Select-Object OwningProcess
```

2. Arrêter le processus si nécessaire
3. Ou changer le port dans le service NSSM

1.15 Différences clés Windows vs Linux

Aspect	Windows	Linux
Gestionnaire de services	NSSM / Services.msc	systemd
Serveur web	IIS	Nginx
Chemins	Backslash \	Forward slash /
Ligne de commande	PowerShell / CMD	Bash
Permissions	ACL Windows	chmod/chown
Base de données	Identique (PostgreSQL)	Identique
Python venv activation	.venv\Scripts\Activate.ps1	source .venv/bin/activate

1.16 Recommandations finales

1.16.1 Si vous choisissez Windows :

Avantages : - Interface graphique familière - Intégration avec l'infrastructure Windows existante - Support Microsoft si vous avez un contrat

Points d'attention : - Plus de ressources consommées (RAM) - Mises à jour Windows peuvent nécessiter des redémarrages - Coût de licence pour Windows Server

1.16.2 Si vous pouvez choisir Linux :

Avantages pour ce projet : - Plus léger et performant - Gratuit et open source - Standard de l'industrie pour applications web - Meilleure documentation en ligne - Plus stable pour les services longue durée

1.17 Support

Pour questions spécifiques à Windows, contactez l'équipe canadienne avec les détails fournis dans `README_FR.md`.

Bonne chance avec votre déploiement Windows !