

# Guide Complémentaire Windows Server \*\*Addendum au README\_FR.md pour Déploiement sur Windows Server 2019/2022\*\* Ce guide couvre uniquement les différences spécifiques à Windows Server. Pour toutes les autres sections (inscription Meta, intégration DHIS-2, codes QR, migration), consultez le [README\_FR.md](./README\_FR.md) principal. --- ## Quand Utiliser Ce Guide - Vous avez une infrastructure Windows Server existante - Votre organisation préfère Windows Server à Linux - Vous devez respecter des politiques informatiques nécessitant Windows \*\*Note\*\* : Linux (Ubuntu) reste recommandé pour la stabilité, les performances et le coût. Ce guide suppose que vous avez une raison spécifique d'utiliser Windows. --- ## Différences Clés : Windows vs Linux | Aspect | Linux (Ubuntu) | Windows Server | |-----|-----|-----| | \*\*Serveur Web\*\* | Nginx | IIS (Internet Information Services) | | \*\*Service Système\*\* | systemd | Service Windows avec NSSM | | \*\*Gestionnaire de Paquets\*\* | apt | Chocolatey (recommandé) | | \*\*Shell\*\* | bash | PowerShell / cmd | | \*\*Certificat SSL\*\* | certbot (Let's Encrypt) | win-acme ou certificat commercial | | \*\*Coût de Licence\*\* | Gratuit | Licence Windows Server requise | --- ## Prérequis Windows ### Matériel Serveur Identique au guide principal, mais ajoutez : - \*\*Système d'exploitation\*\* : Windows Server 2019 ou 2022 (Standard ou Datacenter) - \*\*Licence\*\* : Licence Windows Server valide ### Logiciels Requis ##### 1. Python 3.12+ ``powershell # Télécharger depuis python.org ou utiliser Chocolatey choco install python --version=3.12.0 # Vérifier l'installation python --version `` ##### 2. Git ``powershell choco install git # Vérifier l'installation git --version `` ##### 3. PostgreSQL (Production) ``powershell # Télécharger l'installateur depuis postgresql.org # Ou utiliser Chocolatey choco install postgresql14 # Le service démarre automatiquement `` ##### 4. IIS (Internet Information Services) ``powershell # Installer IIS avec PowerShell (en tant qu'administrateur) Install-WindowsFeature -name Web-Server -IncludeManagementTools # Vérifier l'installation Get-WindowsFeature -Name Web-Server `` ##### 5. URL Rewrite Module pour IIS ``powershell # Télécharger depuis microsoft.com/iis ou utiliser Chocolatey choco install urlrewrite # Requis pour le proxy inverse IIS `` --- ## Installation de l'Application (Windows) ### Étape 1 : Cloner le Dépôt ``powershell # Naviguer vers le répertoire souhaité cd C:\inetpub\ # Cloner le dépôt git clone https://github.com/drjforrest/whatsapp-uic-generator.git # Naviguer dans le projet cd whatsapp-uic-generator `` ### Étape 2 : Créer l'Environnement Virtuel Python ``powershell # Créer l'environnement virtuel python -m venv .venv # Activer l'environnement virtuel .\venv\Scripts\Activate.ps1 # Si vous obtenez une erreur de politique d'exécution : Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser # Puis réessayez l'activation .\venv\Scripts\Activate.ps1 `` ### Étape 3 : Installer les Dépendances ``powershell # Mettre à jour pip python -m pip install --upgrade pip # Installer les dépendances du projet pip install -r requirements.txt `` ### Étape 4 : Générer le Sel de Sécurité ``powershell python scripts\generate\_salt.py # Copiez la sortie pour l'étape suivante `` --- ## Configuration des Variables d'Environnement (Windows) ### Méthode 1 : Fichier .env (Recommandé) ``powershell # Copier l'exemple Copy-Item .env.example .env # Éditer avec notepad ou votre éditeur préféré notepad .env `` Configurez les mêmes variables que dans le guide principal. ### Méthode 2 : Variables d'Environnement Système ``powershell # Définir les variables d'environnement (PowerShell en tant qu'administrateur) [System.Environment]::SetEnvironmentVariable('UIC\_SALT', 'votre\_sel\_ici', 'Machine') [System.Environment]::SetEnvironmentVariable('TWILIO\_ACCOUNT\_SID', 'AC...', 'Machine') # Vérifier [System.Environment]::GetEnvironmentVariable('UIC\_SALT', 'Machine') `` --- ## Configuration de la Base de Données (Windows) ### PostgreSQL sur Windows ``powershell # Se connecter à PostgreSQL (utilisez le mot

de passe défini lors de l'installation) psql -U postgres # Dans l'invite PostgreSQL : CREATE DATABASE uic\_db; CREATE USER uic\_user WITH PASSWORD 'mot\_de\_passe\_securise'; GRANT ALL PRIVILEGES ON DATABASE uic\_db TO uic\_user; \q "" Mettez à jour votre `.env` : "" DATABASE\_URL=postgresql+asyncpg://uic\_user:mot\_de\_passe\_securise@localhost/uic\_db "" --- ## Déploiement en Production (Windows) ### Étape 1 : Créer un Service Windows avec NSSM NSSM (Non-Sucking Service Manager) transforme votre application en service Windows. ""powershell # Installer NSSM choco install nssm # Créer le service (PowerShell en tant qu'administrateur) nssm install WhatsAppUIC "C:\inetpub\whatsapp-uic-generator\venv\Scripts\python.exe" # Configurer les arguments nssm set WhatsAppUIC AppParameters "-m uvicorn app.main:app --host 0.0.0.0 --port 8000 --workers 4" # Définir le répertoire de travail nssm set WhatsAppUIC AppDirectory "C:\inetpub\whatsapp-uic-generator" # Configurer les journaux nssm set WhatsAppUIC AppStdout "C:\inetpub\whatsapp-uic-generator\logs\stdout.log" nssm set WhatsAppUIC AppStderr "C:\inetpub\whatsapp-uic-generator\logs\stderr.log" # Démarrer le service nssm start WhatsAppUIC # Vérifier le statut nssm status WhatsAppUIC "" ### Étape 2 : Configurer IIS comme Proxy Inverse ##### 2.1 Créer le Site IIS ""powershell # Créer un nouveau site dans IIS New-IISSite -Name "WhatsAppUIC" -BindingInformation "\*:80:" -PhysicalPath "C:\inetpub\whatsapp-uic-generator" # Ajouter la liaison HTTPS (après avoir configuré SSL) New-IISSiteBinding -Name "WhatsAppUIC" -BindingInformation "\*:443:" -Protocol https "" ##### 2.2 Configurer URL Rewrite pour le Proxy Créez un fichier `web.config` dans `C:\inetpub\whatsapp-uic-generator\` : ""xml "" ### Étape 3 : Configuration SSL ##### Option 1 : Let's Encrypt avec win-acme (Recommandé) ""powershell # Télécharger win-acme depuis github.com/win-acme/win-acme # Ou installer via Chocolatey choco install win-acme # Exécuter win-acme wacs.exe # Suivre l'assistant interactif : # 1. Choisissez "Create new certificate" # 2. Sélectionnez votre site IIS # 3. Entrez votre email # 4. Acceptez les conditions de service # 5. Le certificat sera installé automatiquement "" ##### Option 2 : Certificat Commercial 1. Achetez un certificat SSL auprès d'une autorité de certification 2. Importez le certificat dans le magasin de certificats Windows 3. Liez le certificat à votre site IIS : ""powershell # Lister les certificats disponibles Get-ChildItem -Path Cert:\LocalMachine\My # Lier le certificat au site New-IISSiteBinding -Name "WhatsAppUIC" -BindingInformation "\*:443:" -CertificateThumbprint "VOTRE\_EMPREINTE\_ICI" -Protocol https -SslFlag 0 "" ### Étape 4 : Ouvrir les Ports du Pare-feu ""powershell # Ouvrir le port 80 (HTTP) New-NetFirewallRule -DisplayName "WhatsApp UIC HTTP" -Direction Inbound -LocalPort 80 -Protocol TCP -Action Allow # Ouvrir le port 443 (HTTPS) New-NetFirewallRule -DisplayName "WhatsApp UIC HTTPS" -Direction Inbound -LocalPort 443 -Protocol TCP -Action Allow "" --- ## Gestion du Service ### Commandes NSSM Utiles ""powershell # Démarrer le service nssm start WhatsAppUIC # Arrêter le service nssm stop WhatsAppUIC # Redémarrer le service nssm restart WhatsAppUIC # Vérifier le statut nssm status WhatsAppUIC # Voir la configuration nssm dump WhatsAppUIC # Supprimer le service (si nécessaire) nssm remove WhatsAppUIC confirm "" ### Gestion via Services Windows ""powershell # Démarrer via PowerShell Start-Service WhatsAppUIC # Arrêter via PowerShell Stop-Service WhatsAppUIC # Vérifier le statut Get-Service WhatsAppUIC # Ou utiliser l'interface graphique : services.msc "" --- ## Surveillance et Journaux (Windows) ### Visualiser les Journaux ""powershell # Journaux de l'application Get-Content C:\inetpub\whatsapp-uic-generator\logs\app.log -Tail 50 -Wait # Journaux du service (stdout) Get-Content C:\inetpub\whatsapp-uic-generator\logs\stdout.log -Tail 50 # Journaux du service (stderr) Get-Content C:\inetpub\whatsapp-uic-generator\logs\stderr.log -Tail 50 #

Journaux des événements Windows Get-EventLog -LogName Application -Source WhatsAppUIC -Newest 20 --- ### Surveillance des Performances "powershell # Utilisation CPU et mémoire du service Get-Process python | Where-Object {\$\_.Path -like "\*whatsapp-uic-generator\*"} | Select-Object CPU, WorkingSet, Id # Surveillance continue (Gestionnaire des Tâches) taskmgr # Ou utiliser Performance Monitor perfmon --- ## Maintenance (Windows) ### Mises à Jour de l'Application "powershell # Arrêter le service nssm stop WhatsAppUIC # Naviguer vers le projet cd C:\inetpub\whatsapp-uic-generator # Activer l'environnement virtuel .\venv\Scripts\Activate.ps1 # Récupérer les mises à jour git pull origin main # Mettre à jour les dépendances pip install -r requirements.txt --upgrade # Redémarrer le service nssm start WhatsAppUIC --- ### Sauvegarde de la Base de Données "powershell # Sauvegarder PostgreSQL \$date = Get-Date -Format "yyyyMMdd" pg\_dump -U uic\_user uic\_db > "C:\backups\uic\_db\_\$date.sql" --- ## Rotation des Journaux Créez une tâche planifiée pour la rotation des journaux : "powershell # Script PowerShell pour la rotation (save as rotate-logs.ps1) \$logPath = "C:\inetpub\whatsapp-uic-generator\logs" \$archivePath = "\$logPath\archive" \$date = Get-Date -Format "yyyyMMdd" # Créer le répertoire d'archive si nécessaire if (!(Test-Path \$archivePath)) { New-Item -ItemType Directory -Path \$archivePath } # Archiver les journaux de plus de 7 jours Get-ChildItem \$logPath -Filter \*.log | Where-Object {\$\_.LastWriteTime -lt (Get-Date).AddDays(-7)} | ForEach-Object { Compress-Archive -Path \$\_.FullName -DestinationPath "\$archivePath\\$(\$\_.BaseName)\_\$date.zip" Remove-Item \$\_.FullName } --- Créer une tâche planifiée : "powershell # Créer une tâche planifiée (exécuter quotidiennement à 2h du matin) \$action = New-ScheduledTaskAction -Execute "PowerShell.exe" -Argument "-File C:\inetpub\whatsapp-uic-generator\scripts\rotate-logs.ps1" \$trigger = New-ScheduledTaskTrigger -Daily -At 2am \$principal = New-ScheduledTaskPrincipal -UserId "SYSTEM" -LogonType ServiceAccount -RunLevel Highest Register-ScheduledTask -TaskName "WhatsAppUIC-LogRotation" -Action \$action -Trigger \$trigger -Principal \$principal --- ## Dépannage Spécifique à Windows ### Problème 1 : Erreur de Politique d'Exécution PowerShell "powershell # Si vous ne pouvez pas activer l'environnement virtuel Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser # Ou pour le système entier (nécessite l'administrateur) Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope LocalMachine --- ### Problème 2 : Le Service Ne Démarre Pas "powershell # Vérifier les journaux d'erreur Get-Content C:\inetpub\whatsapp-uic-generator\logs\stderr.log # Vérifier la configuration NSSM nssm dump WhatsAppUIC # Tester manuellement cd C:\inetpub\whatsapp-uic-generator .\venv\Scripts\Activate.ps1 python -m uvicorn app.main:app --host 0.0.0.0 --port 8000 --- ### Problème 3 : IIS Ne Peut Pas Se Connecter à FastAPI "powershell # Vérifier que FastAPI écoute sur le bon port netstat -ano | findstr :8000 # Tester le proxy manuellement Invoke-WebRequest -Uri http://localhost:8000/whatsapp/health # Vérifier les permissions IIS icacls C:\inetpub\whatsapp-uic-generator --- ### Problème 4 : Erreurs de Certificat SSL "powershell # Vérifier les certificats installés Get-ChildItem -Path Cert:\LocalMachine\My # Vérifier la liaison SSL IIS Get-IISSiteBinding -Name "WhatsAppUIC" # Réimporter le certificat si nécessaire Import-PfxCertificate -FilePath "C:\path\to\cert.pfx" -CertStoreLocation Cert:\LocalMachine\My --- ## Commandes Windows Équivalentes Pour référence rapide lors de l'utilisation de la documentation Linux principale : | Commande Linux | Équivalent Windows PowerShell | -----|-----|-----| `sudo systemctl start service` | `Start-Service ServiceName` | `sudo systemctl stop service` | `Stop-Service ServiceName` | `sudo systemctl restart service` | `Restart-Service ServiceName` | `sudo systemctl status service` | `Get-Service`

```
ServiceName` || `tail -f logfile` | `Get-Content logfile -Tail 50 -Wait` || `grep "pattern" file` | `Select-String -Pattern "pattern" -Path file` || `ps aux \| grep process` | `Get-Process \| Where-Object {$_.Name -like "*process*"}` || `netstat -tulpn` | `Get-NetTCPConnection` || `chmod +x file` | `icacls file /grant Users:RX` | ---  
## Notes Importantes 1. **Mises à Jour Windows** : Planifiez des fenêtres de maintenance pour les mises à jour Windows qui peuvent nécessiter des redémarrages 2. **Antivirus** : Ajoutez des exceptions pour le répertoire de l'application et les ports 8000, 80, 443 3. **Sauvegardes** : Utilisez Windows Server Backup ou une solution tierce pour les sauvegardes système complètes 4. **Surveillance** : Envisagez d'utiliser SCOM (System Center Operations Manager) pour une surveillance d'entreprise --- ## Support Supplémentaire Pour toutes les autres sections non couvertes ici (inscription Meta/Twilio, intégration DHIS-2, codes QR, migration, tests), consultez le [README_FR.md](./README_FR.md) principal. Les concepts et configurations sont identiques—seuls les commandes système et outils diffèrent. --- **Dernière Mise à Jour :** Janvier 2026 **Version :** 1.0.0
```