

Solution de Sauvegarde Odoo entre Serveur OVH et Serveur Local

1. Introduction

Contexte

Notre application Odoo est hébergée sur un serveur OVH, et il est essentiel de mettre en place un système de sauvegarde fiable pour protéger nos données critiques. Cependant, notre serveur local (Windows 10) ne dispose pas d'une adresse IP publique, ce qui rend impossible une connexion directe depuis le serveur OVH. De plus, les sauvegardes peuvent atteindre plusieurs gigaoctets, nécessitant une méthode optimisée pour transférer ces données de manière sécurisée et efficace.

Objectif

Ce document présente une solution technique pour sauvegarder automatiquement les données Odoo du serveur OVH vers le serveur local, en utilisant PowerShell et l'Outil de planification de tâches de Windows.

2. Contraintes et Défis

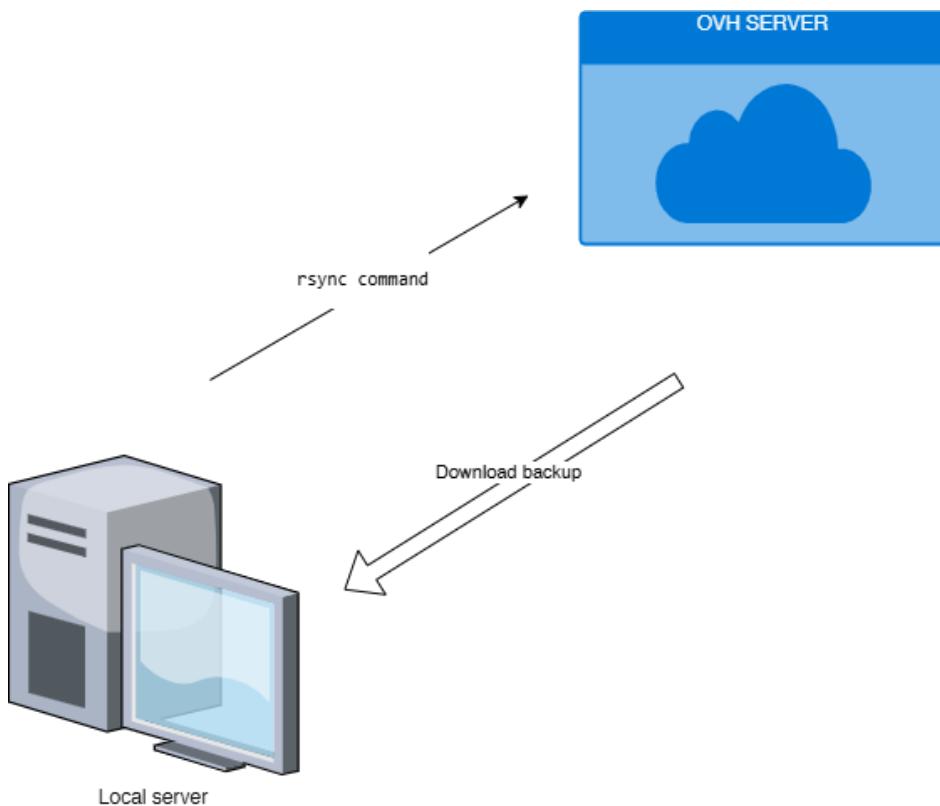
Avant de choisir une solution, nous avons identifié plusieurs contraintes techniques :

- **Pas d'IP publique sur le serveur local**, ce qui empêche le serveur OVH d'initier le transfert.
- **Taille importante des sauvegardes** (plusieurs Go), nécessitant une solution optimisée.
- **Fiabilité** : éviter les pertes de données en cas de coupure réseau.
- **Automatisation complète** : la solution doit fonctionner sans intervention manuelle.

3. Solution Proposée

Méthode : Utilisation de PowerShell avec SSH et l'Outil de planification de tâches

Architecture de la Solution



1. **Le serveur OVH génère des sauvegardes (base de données et fichiers).**
2. **Le serveur Windows 10 exécute un script PowerShell programmé via le Planificateur de Tâches.**
3. **Le script récupère les fichiers depuis OVH avec rsync.**
4. **Les fichiers sont stockés localement et nettoyés automatiquement après un certain délai.**

Pourquoi cette méthode ?

- **Sécurité** : La connexion SSH garantit que les données sont transférées de manière sécurisée.
- **Automatisation** : Le processus peut être entièrement automatisé via l'Outil de planification de tâches.
- **Compatibilité** : PowerShell est natif sur Windows 10, ce qui évite l'installation d'outils tiers.
- **Flexibilité** : Cette méthode fonctionne même si le serveur local n'a pas d'adresse IP publique.

Pourquoi rsync au lieu de scp ?

L'outil **scp** pourrait être utilisé pour transférer les sauvegardes, mais il présente plusieurs inconvénients :

- Pas de reprise en cas d'interruption.
- Pas de transfert différentiel (il retransfère tout à chaque fois, gaspillant la bande passante).
- Pas de compression, ce qui ralentit le transfert.

rsync, en revanche, offre plusieurs avantages cruciaux :

- Compression des données pour un transfert plus rapide.
- Reprise automatique en cas d'interruption.
- Transfert différentiel : seuls les fichiers modifiés sont transférés.
- Suppression des fichiers après un transfert réussi, pour éviter l'encombrement du serveur OVH.

Avantages de la Solution :

Sécurité Renforcée

- Les données sont transférées via une connexion SSH chiffrée, protégeant ainsi les informations sensibles contre les interceptions.

Automatisation Complète

- Les sauvegardes peuvent être planifiées et exécutées sans intervention manuelle, garantissant une régularité et une fiabilité accrues.

Utilisation d'Outils Natifs

- Aucune installation supplémentaire n'est requise (sauf pour SSH, qui est facile à configurer).

Faible Coût

- Aucun service tiers coûteux n'est nécessaire, contrairement à l'utilisation de solutions cloud.

4. Description Technique de la Solution

Configuration et Script de Sauvegarde sur OVH

Sur le serveur OVH, nous allons automatiser la création des sauvegardes avec un script exécuté via cron.

Création du script de sauvegarde

Script complet pour le serveur OVH (backup_odoo.sh)

```
bash                                         Copy

BACKUP_DIR="/backups/odoo"                  # Répertoire de stockage des sauvegardes
DB_NAME="nom_de_la_base"                     # Nom de la base de données Odoo
DB_USER="utilisateur_postgres"               # Utilisateur PostgreSQL
ODOO_DATA_DIR="/var/lib/odoo"                 # Répertoire des données Odoo
DATE=$(date +%Y%m%d_%H%M%S)                  # Date et heure actuelles
BACKUP_NAME="odoo_backup_$DATE"               # Nom de la sauvegarde
RETENTION_DAYS=30                            # Nombre de jours de rétention des sauvegardes

# Créer le répertoire de sauvegarde s'il n'existe pas
mkdir -p "$BACKUP_DIR"

# Fonction pour logger les messages
log_message() {
    echo "[$(date +"%Y-%m-%d %H:%M:%S")] $1"
}

# Sauvegarde de la base de données PostgreSQL
log_message "Début de la sauvegarde de la base de données PostgreSQL..."
pg_dump -U "$DB_USER" -h localhost -F c -b -f "$BACKUP_DIR/${BACKUP_NAME}.dump" "$DB_NAME"
if [ $? -eq 0 ]; then
    log_message "Sauvegarde de la base de données terminée : $BACKUP_DIR/${BACKUP_NAME}.dump"
else
    log_message "Erreur lors de la sauvegarde de la base de données."
    exit 1
fi

# Sauvegarde des fichiers Odoo
log_message "Début de la sauvegarde des fichiers Odoo..."
tar -czvf "$BACKUP_DIR/${BACKUP_NAME}.tar.gz" "$ODOO_DATA_DIR"
if [ $? -eq 0 ]; then
    log_message "Sauvegarde des fichiers Odoo terminée : $BACKUP_DIR/${BACKUP_NAME}.tar.gz"
else
    log_message "Erreur lors de la sauvegarde des fichiers Odoo."
    exit 1
fi

# Nettoyage des anciennes sauvegardes
log_message "Nettoyage des anciennes sauvegardes (plus de $RETENTION_DAYS jours)..."
find "$BACKUP_DIR" -type f -name "*.dump" -mtime +$RETENTION_DAYS -exec rm -f {} \;
find "$BACKUP_DIR" -type f -name "*.tar.gz" -mtime +$RETENTION_DAYS -exec rm -f {} \;
log_message "Nettoyage terminé."

log_message "Sauvegarde hebdomadaire terminée avec succès."
```

Utilisation du Script

Créer le répertoire de sauvegarde :

```
bash                                         Copy
sudo mkdir -p /backups/odoo
sudo chown -R $USER:$USER /backups/odoo
```

Rendre le script exécutable :

```
bash                                         Copy
chmod +x /chemin/vers/backup_odoo.sh
```

Tester le script :

Exécuter le script manuellement pour vérifier qu'il fonctionne correctement :

```
bash                                         Copy
/chemin/vers/backup_odoo.sh
```

Automatiser avec cron :

```
bash                                         Copy
crontab -e
```

Ajoutez la ligne suivante pour exécuter le script tous les dimanches à 3h du matin :

```
bash                                         Copy
0 3 * * 0 /chemin/vers/backup_odoo.sh >> /var/log/odoo_backup.log 2>&1
```

Côté Serveur Local (Windows 10)

Script PowerShell pour Automatiser la Récupération (fetch_backups.ps1):

```
powershell
# Variables
$BackupDir = "C:\OdooBackups"          # Répertoire local pour stocker les sauvegardes
$RemoteServer = "user@monserveur.ovh"    # Utilisateur et adresse du serveur OVH
$RemotePath = "/var/backups/odoo/"       # Répertoire des sauvegardes sur le serveur OVH
$LogFile = "$BackupDir\backup_log.txt"   # Fichier de log pour enregistrer les détails
$RsyncPath = "C:\Program Files\Git\usr\bin\rsync.exe" # Chemin de rsync (Git pour Windows)
$SshPath = "C:\Program Files\Git\usr\bin\ssh.exe"      # Chemin de ssh (Git pour Windows)

# Créer le dossier de sauvegarde s'il n'existe pas
if (!(Test-Path -Path $BackupDir)) {
    New-Item -ItemType Directory -Path $BackupDir
}

# Initialiser le fichier de log
$Timestamp = Get-Date -Format "yyyy-MM-dd HH:mm:ss"
Add-Content -Path $LogFile -Value "[${$Timestamp}] Début du processus de sauvegarde."

# Commande rsync pour récupérer les sauvegardes
try {
    & $RsyncPath -avz --progress --partial --remove-source-files ` 
        $RemoteServer:$RemotePath $BackupDir >> $LogFile 2>&1
    if ($LASTEXITCODE -eq 0) {
        Add-Content -Path $LogFile -Value "[${$Timestamp}] Transfert des sauvegardes réussi."
    } else {
        throw "Erreur lors du transfert des sauvegardes."
    }
} catch {
    Add-Content -Path $LogFile -Value "[${$Timestamp}] Erreur : $_"
    Write-Error "Erreur lors du transfert des sauvegardes. Voir le fichier de log : $LogFile"
    exit 1
}

# Nettoyage des fichiers vides après transfert
try {
    & $SshPath $RemoteServer "find $RemotePath -type f -empty -delete"
    if ($LASTEXITCODE -eq 0) {
        Add-Content -Path $LogFile -Value "[${$Timestamp}] Nettoyage des fichiers vides réussi."
    } else {
        throw "Erreur lors du nettoyage des fichiers vides."
    }
} catch {
    Add-Content -Path $LogFile -Value "[${$Timestamp}] Erreur : $_"
    Write-Error "Erreur lors du nettoyage des fichiers vides. Voir le fichier de log : $LogFile"
    exit 1
}

# Message de succès
Write-Output "Sauvegarde récupérée avec succès."
Add-Content -Path $LogFile -Value "[${$Timestamp}] Processus de sauvegarde terminé avec succès."
```

Utilisation du Script

1. Installer Git pour Windows :

Téléchargez et installez [Git pour Windows](#).

Assurez-vous que rsync et ssh sont disponibles dans le chemin C:\Program Files\Git\usr\bin.

2. Configurer l'authentification SSH :

Générez une paire de clés SSH sur Windows :

```
powershell  
ssh-keygen -t rsa -b 4096
```

Copy

Copiez la clé publique sur le serveur OVH :

```
powershell  
type $env:USERPROFILE\.ssh\id_rsa.pub | ssh user@monserveur.ovh "mkdir -p ~/.ssh && cat  
>> ~/.ssh/authorized_keys"
```

Copy

3. Exécuter le script :

Sauvegardez le script dans un fichier, par exemple fetch_backups.ps1.

Exécutez-le manuellement pour tester :

```
powershell  
.\\fetch_backups.ps1
```

Copy

4. Automatiser avec l'Outil de planification de tâches :

- Ouvrez l'**Outil de planification de tâches**.
- Créez une nouvelle tâche :
 - Déclencheur : Hebdomadaire, le jour et l'heure souhaités.
 - Action :

Programme : powershell.exe

Arguments : -ExecutionPolicy Bypass -File C:\\Scripts\\retrieve_backup.ps1

Exemple de fichier de log :

```
plaintext  
[2023-10-15 14:30:22] Début du processus de sauvegarde.  
[2023-10-15 14:30:25] Transfert des sauvegardes réussi.  
[2023-10-15 14:30:27] Nettoyage des fichiers vides réussi.  
[2023-10-15 14:30:27] Processus de sauvegarde terminé avec succès.
```

Copy

Conclusion

Ce script complet et automatisé garantit une sauvegarde fiable et sécurisée de votre application Odoo, tout en étant facile à maintenir et à surveiller.

Cette solution garantit la sécurité, la rapidité et l'automatisation des sauvegardes Odoo vers un serveur Windows local, tout en optimisant la bande passante et la fiabilité des transferts.