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)

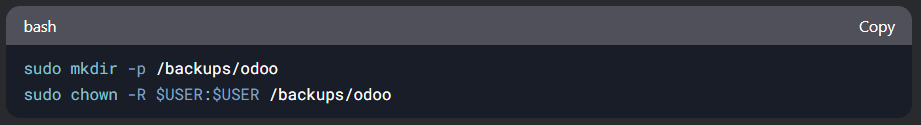


### Explication du Script

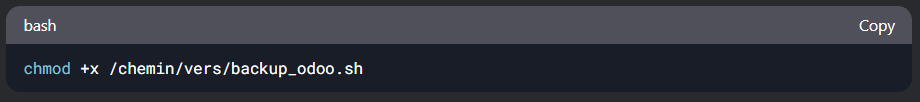
1. **Configuration** :
   * BACKUP\_DIR : Répertoire où les sauvegardes seront stockées.
   * DB\_NAME : Nom de la base de données Odoo.
   * DB\_USER : Utilisateur PostgreSQL pour accéder à la base de données.
   * ODOO\_DATA\_DIR : Répertoire contenant les fichiers Odoo.
   * RETENTION\_DAYS : Nombre de jours pendant lesquels les sauvegardes sont conservées.
2. **Fonction log\_message** :
   * Permet de logger les étapes du script avec un timestamp pour faciliter le débogage et le suivi.
3. **Sauvegarde de la base de données** :
   * Utilise pg\_dump pour créer une sauvegarde compressée de la base de données.
   * Le fichier de sauvegarde est nommé avec la date et l'heure pour éviter les conflits.
4. **Sauvegarde des fichiers Odoo** :
   * Utilise tar pour compresser le répertoire Odoo en un fichier .tar.gz.
5. **Nettoyage des anciennes sauvegardes** :
   * Supprime les sauvegardes de plus de 30 jours pour éviter l'accumulation de fichiers inutiles.
6. **Journalisation** :
   * Chaque étape est enregistrée avec un timestamp pour un suivi clair.

### Utilisation du Script

Créer le répertoire de sauvegarde :

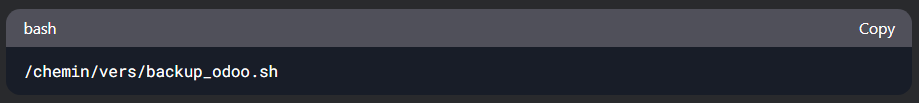


Rendre le script exécutable :

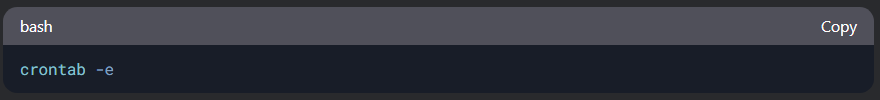


Tester le script :

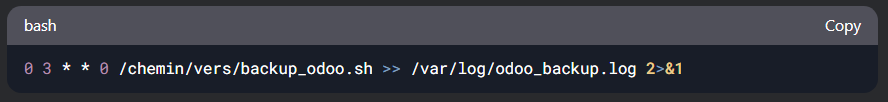
Exécutez le script manuellement pour vérifier qu'il fonctionne correctement :



Automatiser avec cron :



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



## Côté Serveur Local (Windows 10)

Étape 1 : Récupérer les sauvegardes via PowerShell

Voici un script PowerShell pour récupérer les sauvegardes depuis le serveur OVH.

Script PowerShell (fetch\_backups.ps1)

