

Automatisation de la sauvegarde de fichiers  
utilisateurs avec un script PowerShell et  
planification des sauvegardes



*Par Mohamed Belgharbi*

## 1. Objectif

Assurer la protection régulière des données utilisateurs en mettant en place une sauvegarde automatique à l'aide d'un script PowerShell planifié, afin de prévenir la perte de données en cas d'incident matériel ou logiciel.

## 2. Contexte professionnel

Dans toute organisation, la perte de données peut avoir des conséquences graves : perte de productivité, coûts financiers, atteinte à l'image de l'entreprise.

Dans ce projet, l'objectif était d'assurer une **sauvegarde régulière** des documents critiques sans intervention humaine grâce à une **solution légère** et **économique**.

## 3. Matériel et Logiciels utilisés

- Système d'exploitation : Windows 10 Professionnel
- Outil d'automatisation : PowerShell 5.1
- Outil de planification : Planificateur de tâches Windows
- Support de sauvegarde : second disque dur local ou serveur de fichiers réseau

## 4. Détail de la réalisation

### 4.1 Analyse des besoins

- Déterminer quels dossiers doivent être sauvegardés (ex : C:\Utilisateurs\Documents).
- Déterminer la fréquence souhaitée : **quotidienne** à 20h.
- Définir le support de sauvegarde : **disque local** ou **serveur distant**.

## 4.2 Rédaction du script PowerShell

Script utilisé :

### Variables

```
$source = "C:\Utilisateurs\Documents" $destination = "D:\Sauvegarde"
```

### Commande de copie

```
Copy-Item -Path $source* -Destination $destination -Recurse -Force
```

- -Recurse : copie tous les sous-dossiers.
- -Force : écrase les fichiers existants sans demander confirmation.

## 4.3 Mise en place de la planification

- Création d'une tâche dans **Planificateur de tâches Windows** :
  - Déclencheur : tous les jours à 20h.
  - Action : exécuter powershell.exe avec le script .ps1 en paramètre.
  - Conditions : exécution autorisée même si l'utilisateur n'est pas connecté.

## 4.4 Phase de tests

- Modification de fichiers dans le dossier source.
- Lancement manuel du script pour vérifier son bon fonctionnement.
- Vérification que la tâche planifiée se lance automatiquement et sans erreurs.

## 5. Résultats obtenus

- Sauvegarde quotidienne des documents sans intervention humaine.
- Facilité de mise en œuvre et faible coût (outils intégrés à Windows).
- Réduction du risque de perte de données.

## 6. Difficultés rencontrées

- Droits d'exécution restreints sous PowerShell (politique de sécurité par défaut).
- Configuration correcte des paramètres du Planificateur pour éviter les erreurs liées aux chemins d'accès.

## 7. Solutions apportées

- Modification de la politique d'exécution PowerShell :  
Set-ExecutionPolicy RemoteSigned (autorise l'exécution des scripts locaux).
- Utilisation de chemins absolus dans le Planificateur pour éviter les erreurs.

## 8. Perspectives d'amélioration

- Ajouter une **notification par email** en cas d'échec de la sauvegarde.
- Mettre en place un **historique** ou un **log** des sauvegardes réalisées.
- Envisager une **sauvegarde vers un stockage distant sécurisé** (NAS, cloud).

## 9. Compétences mises en œuvre

- Gérer le patrimoine informatique
- Mettre à disposition un service informatique
- Automatiser les tâches et assurer la continuité de service
- Organiser son développement professionnel

## Conclusion

Ce projet m'a permis de mettre en pratique mes compétences en automatisation et en gestion des services informatiques à travers l'utilisation de PowerShell et du Planificateur de tâches Windows.

Il m'a sensibilisé à l'importance de la sauvegarde régulière des données pour garantir la continuité d'activité en cas d'incident.

Grâce à cette expérience, j'ai renforcé ma capacité à analyser un besoin, à proposer

une solution technique adaptée, et à assurer sa mise en œuvre de manière fiable et sécurisée.

Ce projet ouvre également des perspectives d'amélioration continue, notamment en termes de sécurisation des sauvegardes et de mise en place de notifications en cas d'erreur.