**Documentation du script PowerShell pour l'inventaire système**

**Description**

Ce script PowerShell permet de récupérer diverses informations système sur un ordinateur Windows et de les exporter dans un fichier CSV situé sur le bureau de l'utilisateur. Il collecte des données sur l'adresse IP, le nom d'hôte, l'uptime, l'utilisation CPU, la mémoire vive (RAM), l'espace SWAP, l'utilisation des disques, ainsi que les informations sur le système d'exploitation.

**Emplacement du fichier de sortie**

Le fichier CSV sera créé sur le bureau de l'utilisateur sous le nom inventaire.csv.

$outputFile = "$env:USERPROFILE\Desktop\inventaire.csv"

**Informations collectées**

Le script récupère les informations suivantes :

* **Adresse IP** : Première adresse IPv4 détectée sur le système
* **Nom d'hôte** : Nom de l'ordinateur
* **Uptime (jours)** : Nombre de jours écoulés depuis le dernier redémarrage
* **Charge CPU (%)** : Utilisation moyenne du processeur
* **RAM (Go)** : Quantité de mémoire vive totale
* **SWAP (Go)** : Utilisation actuelle de la mémoire d'échange (SWAP)
* **Espace disque utilisé (Go)** : Somme de l'espace utilisé sur tous les disques durs
* **Espace disque disponible (Go)** : Somme de l'espace libre sur tous les disques durs
* **Système d'exploitation** : Nom de l'OS installé
* **Version de l'OS** : Version détaillée du système d'exploitation

**Fonctionnement du script**

**1. Création du fichier CSV**

Si le fichier n'existe pas, il est créé avec une ligne d'en-tête.

"Adresse IP,Nom d'hôte,Uptime (jours),Charge CPU (%),RAM (Go),SWAP (Go),Espace disque utilisé (Go),Espace disque dispo (Go),OS,Version OS" | Out-File -FilePath $outputFile -Encoding utf8

**2. Récupération des informations système**

**2.1 Adresse IP et Nom d'hôte**

$ipAddress = (Get-NetIPAddress -AddressFamily IPv4 | Select-Object -First 1).IPAddress

$hostname = $env:COMPUTERNAME

**2.2 Uptime (en jours)**

$uptime = (Get-CimInstance -ClassName Win32\_OperatingSystem).LastBootUpTime

$uptimeDays = ((Get-Date) - $uptime).Days

**2.3 Charge CPU moyenne**

$cpuUsage = (Get-WmiObject Win32\_Processor | Measure-Object -Property LoadPercentage -Average).Average

**2.4 Mémoire RAM et SWAP**

$ramUsage = [math]::Round((Get-CimInstance -ClassName Win32\_OperatingSystem).TotalVisibleMemorySize / 1MB, 2)

$swapUsage = [math]::Round((Get-CimInstance -ClassName Win32\_PageFileUsage).CurrentUsage / 1MB, 2)

**2.5 Utilisation des disques**

$diskInfo = Get-CimInstance -ClassName Win32\_LogicalDisk -Filter "DriveType=3"

$totalDiskUsed = [math]::Round(($diskInfo | Measure-Object -Property Size -Sum).Sum / 1GB, 2)

$totalDiskAvailable = [math]::Round(($diskInfo | Measure-Object -Property FreeSpace -Sum).Sum / 1GB, 2)

**2.6 Informations sur le système d'exploitation**

$osName = (Get-CimInstance -ClassName Win32\_OperatingSystem).Caption

$osVersion = (Get-CimInstance -ClassName Win32\_OperatingSystem).Version

**3. Enregistrement des données dans le fichier CSV**

Les informations collectées sont stockées dans un objet PowerShell et ajoutées au fichier CSV.

$info = [PSCustomObject]@{

Adresse\_IP = $ipAddress

Nom\_Hôte = $hostname

Uptime\_Jours = $uptimeDays

Charge\_CPU = $cpuUsage

RAM\_Go = $ramUsage

SWAP\_Go = $swapUsage

Espace\_Disque\_Utilisé\_Go = $totalDiskUsed

Espace\_Disque\_Disponible\_Go = $totalDiskAvailable

OS = $osName

Version\_OS = $osVersion

}

$info | ConvertTo-Csv -NoTypeInformation | Select-Object -Skip 1 | Out-File -FilePath $outputFile -Append -Encoding utf8

**4. Affichage du message de confirmation**

Une fois l'inventaire terminé, un message est affiché pour informer l'utilisateur.

Write-Output "Inventaire terminé. Le fichier CSV a été créé sur le bureau : $outputFile"

**Exécution du script**

1. Ouvrir PowerShell en tant qu'administrateur
2. Se rendre dans le répertoire contenant le script
3. Exécuter la commande suivante :
4. .\NomDuScript.ps1

**Améliorations possibles**

* Ajouter un horodatage au fichier CSV pour conserver un historique des inventaires.
* Intégrer une collecte des informations sur les interfaces réseau.
* Récupérer plus de détails sur chaque disque dur (modèle, numéro de série, etc.).
* Automatiser l'exécution du script via une tâche planifiée.

**Conclusion**

Ce script est un outil pratique pour collecter rapidement des informations essentielles sur un poste Windows. Il peut être utilisé pour un audit interne, un suivi des performances ou encore une gestion de parc informatique.