

Script PowerShell

1- Crée un script copiant tous les fichiers de « Mes documents » vers le disque Z:\Backup

```
Copy-Item -Path "C:\Users\Mes documents\*" -Destination "Z:\Backup\" -Recurse
```

2- Commentez les actions faites par ce script à chaque étapes.

```
1 New-LocalUser -Name "Nom_utilisateur" -Description "Description utilisateur" -Password #Cette ligne crée un nouvelle utilisateur avec le nom "Nom_utilisateur", une Description "Description utilisateur".
2 (ConvertTo-SecureString "Mot_de_passe" -AsPlainText -Force) # et un mot de passe spécifique le Mdp et ensuite converti en une chaîne de caractère sécurisée. AsPlainText dit que le mot de passe est en texte
3 #brut et -Force est utilisé pour forcer
4
5 Set-LocalUser -Name "Nom utilisateur" -Description "Nouvelle description" #Cette ligne modifie les propriétés d'un utilisateur local. Ici change le nom et la description de l'utilisateur local
6
7
8
9 Remove-LocalUser -Name "Nom_utilisateur" #Ici on viens supprime l'utilisateur local spécifié par le nom "Nom_utilisateur" de la machine.
```

```
New-LocalUser -Name "Nom_utilisateur" -Description "Description_utilisateur" -Password
(ConvertTo-SecureString "Mot_de_passe" -AsPlainText -Force)

Set-LocalUser -Name "Nom_utilisateur" -Description "Nouvelle_description"

Remove-LocalUser -Name "Nom_utilisateur"
```

3- par déduction quelle serai la commande à exécuter afin d'avoir des informations sur un compte « Nom_utilisateur »

Get-localUser -name

4- commentez ligne par ligne des actions faites par ce script :

Get-Process | Format-Table Id, ProcessName, CPU, Memory -AutoSize #Ici on prend les processus en cours sur la machine et on le met dans un tableau qui répertorie tous ces procès avec un ID, le nom du process la quantité de CPU utiliser et la R.A.M.

\$processId = Read-Host "Entrez l'ID du processus que vous souhaitez terminer" # Ici on vient demander d'entrer l'ID du processus que l'on veut arrêter en affichant le message "Entrez l'ID du processus que vous souhaitez terminer". La valeur saisie par l'utilisateur est stockée dans la variable \$processId.

Stop-Process -Id \$processId -Force # Ici on vient arrête le processus spécifié par l'ID stocké dans la variable \$processId. -Force viens forcer l'arrêt du processus sans demander de confirmation à l'utilisateur.

5- commentez ligne par ligne des actions faite par ce script :

```
$threshold = 80 # Ici on crée une variable threshold que l'on définit a 80
$d disks = Get-PSDrive -PSProvider FileSystem | Where-Object {$_.Free -ne $null} # Ici
on récupère la liste des disques sur le système qui sont des systèmes de fichiers et ont de
l'espace libre non nul
foreach ($disk in $disks) { # Ici on initialise une boucle pour chaque disk dans disks
    $usage = ($disk.Used / $disk.Free) * 100
    #Calcule le pourcentage d'utilisation du
    disque
    if ($usage -gt $threshold) { # Ici on dit
    que si l'utilisation du disk dépasse les
    80% on envoie une alerte disant que le
    seuil d'utilisation du disque depasse le
    seuil autorisé
        # Définie le sujet et le corps de l'email.
        $subject = "Alerte d'utilisation du disque"
        $body = "L'utilisation du disque $($disk.Name) a dépassé le seuil de
        $threshold%."
        # Envoie l'e-mail d'alerte à admin@example.com , car l'utilisation dépasse 80%
        Send-MailMessage -To "admin@example.com" -From "alert@example.com" -Subject
        $subject -Body $body -SmtpServer "smtp.example.com"
    }
    # Récupère les informations sur le disque et les affiche.
    $name = $disk.Name
    $size = [math]::Round($disk.Used / 1GB, 2)
    $free = [math]::Round($disk.Free / 1GB, 2)
    $used = [math]::Round($disk.Used / 1GB, 2)
    $usage = [math]::Round($usage, 2)
    #Affiche les informations sur le disque, y compris le nom, la taille, l'espace libre, l'espace utilisé et le
    pourcentage d'utilisation.
    Write-Host "$name - Taille: $size Go, Libre: $free Go, Utilisé: $used Go,
    Utilisation: $usage%"
}
```

6- créez un script faisant les actions suivantes :

- Crée un dossier dans C: nommé « Apps »
- Dans ce dossier créez un fichier Texte « info.txt »
- Écrivez dans ce fichier text :
- En ligne 1 la date du jour
- En ligne 2 le nom de l'ordinateur qui exécute ce script
- En ligne 3 l'espace du disque « c: » de cet ordinateur
- En ligne 4 la ram total de cet ordinateur
- En ligne 5 le cpu de cet ordinateur
- En ligne 6 le compte utilisateur actuel de cet ordinateur

```
New-Item -Path "C:\Apps" -ItemType Directory
```

```
$chemin = "C:\Apps\info.txt"
```

```
Get-Date | Set-Content -Path $chemin
```

```
$env:COMPUTERNAME | Add-Content -Path $chemin
```

```
(Get-WmiObject Win32_LogicalDisk -Filter "DeviceID='C:'" | Select-Object -ExpandProperty FreeSpace) / 1GB | Add-Content -Path $chemin
```

```
(Get-WmiObject Win32_ComputerSystem | Select-Object -ExpandProperty TotalPhysicalMemory) / 1GB | Add-Content -Path $chemin
```

```
(Get-WmiObject Win32_Processor | Select-Object -ExpandProperty Name) | Add-Content - Path $chemin $env:USERNAME | Add-Content -Path $chemin
```

7- comment appeler de l'aide sur une commande en powershell ?

Get-Help

8- quelle commande pouvez-vous lancer pour obtenir la liste de tous les sous-dossiers de « C: \Windows »

Expliquez

Get-ChildItem -Directory -Path "C:\Windows"

9- crée une modification du script de la question 2 avec la création d'un compte jean et de test si le compte existe avec un message d'erreur comme quoi le compte existe déjà.

```
if (-not (Get-LocalUser -Name "Nom_utilisateur" -ErrorAction SilentlyContinue)) {  
New-LocalUser -Name "Nom_utilisateur" -Description "Description utilisateur" -Password (ConvertTo-SecureString  
"Mot_de_passe" -AsPlainText -Force)  
Write-Host "Utilisateur créé"  
} else { # s'il existe déjà écrire "utilisateur déjà créé" Write-Host "Utilisateur déjà créé" }
```

```
Set-LocalUser -Name "Nom_utilisateur" -Description "Nouvelle description"  
Remove-LocalUser -Name "Nom_utilisateur"
```

