Automatisation de la gestion des utilisateurs et des dossiers

1) Contexte

Ordinateur Windows 10/11

Commandes :

```
PS > Get-Command -Module Microsoft.PowerShell.LocalAccounts

CommandType     Name
-----------     ----
Cmdlet          Add-LocalGroupMember          => Ajouter un utilisateur au groupe local
Cmdlet          Disable-LocalUser             => Désactiver un compte d'utilisateur local
Cmdlet          Enable-LocalUser              => Activer un compte d'utilisateur local
Cmdlet          Get-LocalGroup                => Afficher les préférences du groupe local
Cmdlet          Get-LocalGroupMember          => Afficher la liste des membres du groupe local
Cmdlet          Get-LocalUser                 => Afficher les informations d'un compte d'utilisateur
local
Cmdlet          New-LocalGroup                => Créer un nouveau groupe local
Cmdlet          New-LocalUser                 => Créer un nouveau compte d'utilisateur local
Cmdlet          Remove-LocalGroup             => Supprimer un groupe local
Cmdlet          Remove-LocalGroupMember       => Supprimer un membre d'un groupe local
Cmdlet          Remove-LocalUser              => Supprimer un compte d'utilisateur local
Cmdlet          Rename-LocalGroup             => Renommer un groupe local
Cmdlet          Rename-LocalUser              => Renommer un compte d'utilisateur local
Cmdlet          Set-LocalGroup                => Modifier les paramètres d'un groupe local
Cmdlet          Set-LocalUser                 => Modifier les paramètres de compte d'un utilisateur
local
```

2) Objectifs

Sur un ordinateur local Windows 10/11, vous devez **créer 4 nouveaux comptes**

- Ces quatre comptes doivent être membre du groupe local Windows existant appelé **Utilisateurs**.
- Un de ces quatre comptes appartient en plus à un groupe local **Responsable** qui est à créer.

Dans l'arborescence des dossiers de votre ordinateur, 2 dossiers doivent être également créés :

- Un dossier **Travail** avec un accès en **mise à jour** (Modifier) ;
- Un dossier **Projet** avec un **accès en lecture seulement** pour ces quatre comptes mais en **mise à jour** pour le compte membre du groupe local **Responsable**.

Vous devez automatiser toutes ces actions d'administration dans un seul script Powershell.

3) création de 4 nouveaux comptes

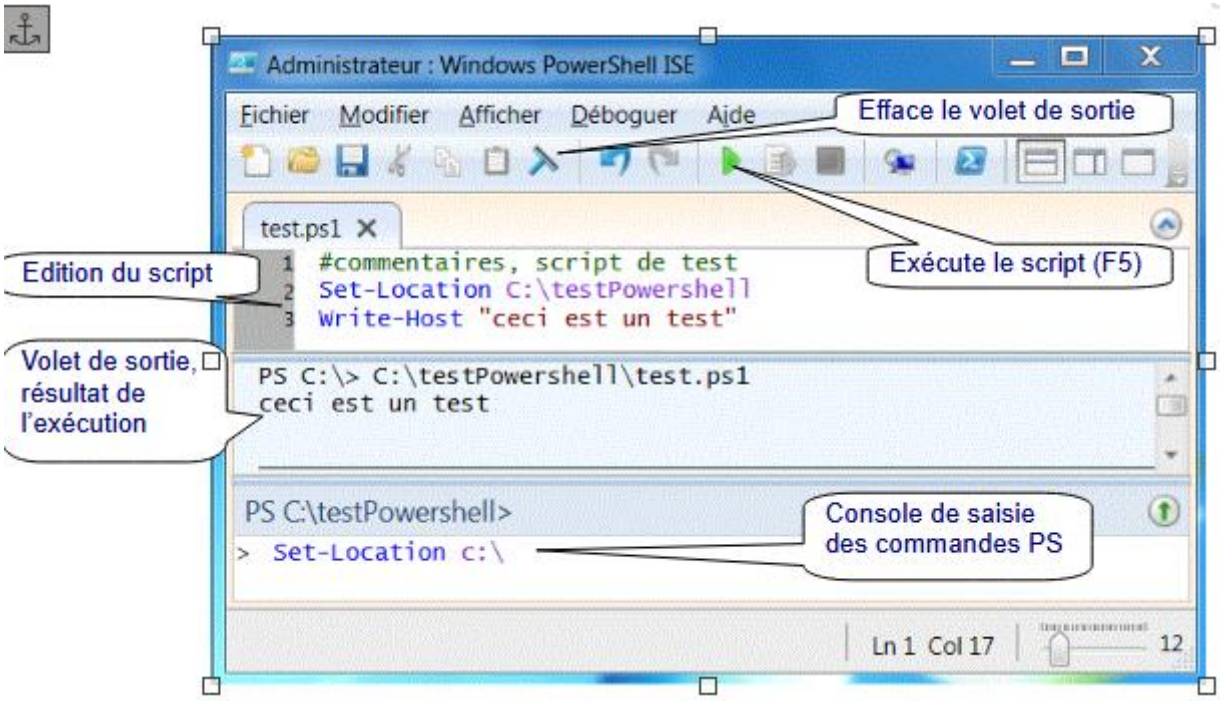
Utilisation de Windows PowerShell ISE

Lancer Windows PowerShell ISE sous Windows 10:

Démarrer\Tous les programmes\Accessoires\Windows PowerShell\Windows PowerShell ISE

Présentation des différentes fenêtres :

L'éditeur présente trois volets, un premier volet pour l'éditeur de scripts, un deuxième volet de sortie pour afficher le résultat d'exécution et un volet qui correspond à la console de saisie interactive des commandes PowerShell (PS). Suite à l'exécution d'un script, les variables de celui-ci sont accessibles dans la console de saisies des commandes.




```
$fichier= "C:\testPowerShell\Projets\listeCompte.txt"
if (Test-Path $fichier){
   $colLIgnes=Get-Content $fichier
   foreach($ligne in $colLignes){
      $tabCompte=$ligne.Split("/")
      $nom= $tabCompte[0]
      $FullName= $tabCompte[1]
      $description= $tabCompte[2]
      $newGroupe=$tabCompte[4]

try
{
  $compte=New-LocalUser -Name $nom  -FullName $FullName -Description $description
  Write-Host "$nom ajouté"
  # ajout des comptes au groupe local Utilisateurs

  Add-LocalGroupMember -Group "Utilisateurs" -Member $nom
   # ajout du compte util4 au groupe local Responsable
```

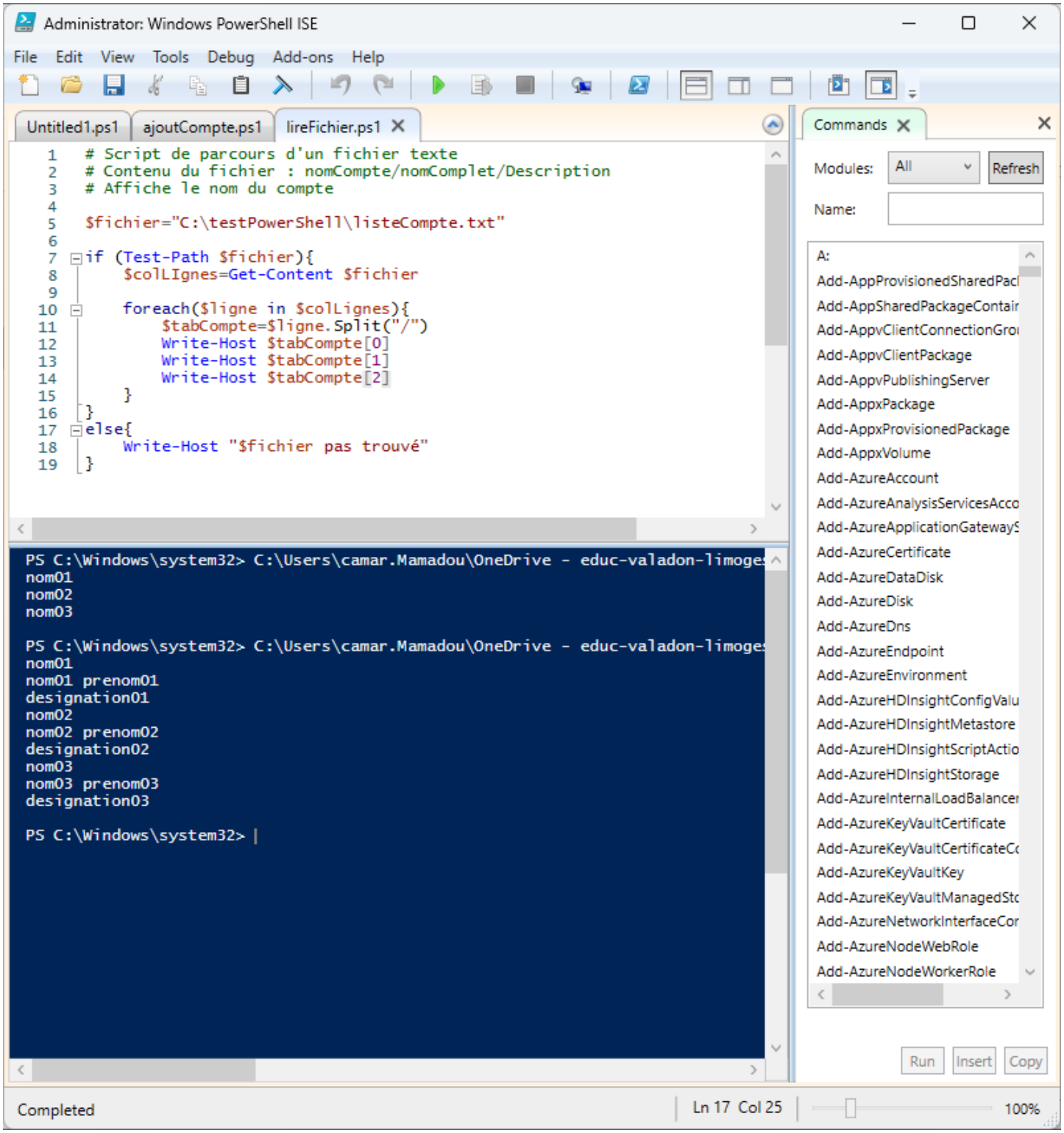
```
   if($description -eq "Responsable"){
        $newGroupe= New-LocalGroup -Name "Responsable"
        Add-LocalGroupMember -Group $newGroupe -Member $nom
}
}
   catch{
      Write-Host "$nom existe déjà"
}
}
}
# Création du dossier Travail et Projet

   $empl= "C:\testPowerShell\Projets\"
   if(Test-Path $empl)
{
   try
 {
        $Travail = New-Item Travail -ItemType Directory
        $Projet = New-Item Projet -ItemType Directory
        Write-Host "Le dossier $Travail a été crée"
         Write-Host  "Le dossier $Projet a été crée"
}
catch
{
   Write-Host "$Travail existe déja"
   Write-Host "$Projet existe déja"
}
 # Droits d'accès aux dossiers
  $doss= "C:\testPowerShell\Projets\Travail"
 if(Test-Path $doss )
 {
 try{
   $droitUtilTrail= Add-NTFSAccess –Path "C:\Travail" -Account "Utilisateurs" –AccessRights Modify
```

```
    $droitUtilProj= Add-NTFSAccess –Path "C:\Projet" -Account "Utilisateurs" –AccessRights Read
    $droitRespProj= Add-NTFSAccess –Path "C:\Projet" -Account "Responsable" –AccessRights Modify
    Write-Host "Droit de modification du dossier  $Travail accordé  aux utilisateurs"
     Write-Host "Droit de lecture du dossier $Projet accordé  aux utilisateurs"
      Write-Host "Droit de modification du dossier $Projet accordé  au Responsable"
}
catch
{
 Write-Host "Dossier $Travail n'existe pas"
  Write-Host "Dossier $Projet n'existe pas"
  }
  }
  }
```

Contenu du fichier **listeCompte.txt** :

```
util1/util1 test1/Utilisateurs/password1
util2/util2 test2/Utilisateurs/password2
util3/util3 test3/Utilisateurs/password3
util4/util4 test4/Utilisateurs et Responsable/password4
```

4) Test


Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled1.ps1
ajoutCompte.ps1
lireFichier.ps1
Script de parcours d'un fichier texte
Contenu du fichier : nomCompte/nomComplet/Description
Affiche le nom du compte
$fichier="C:\testPowerShell\listeCompte.txt"
if (Test-Path $fichier){
$colLignes=Get-Content $fichier
foreach($ligne in $colLignes){
$tabCompte=$ligne.Split("/")
Write-Host $tabCompte[0]
Write-Host $tabCompte[1]
Write-Host $tabCompte[2]
}
}
else{
Write-Host "$fichier pas trouvé"
}
PS C:\Windows\system32> C:\Users\camar.Mamadou\OneDrive - educ-valadon-limoges
nom01
nom02
nom03
PS C:\Windows\system32> C:\Users\camar.Mamadou\OneDrive - educ-valadon-limoges
nom01
nom01 prenom01
designation01
nom02
nom02 prenom02
designation02
nom03
nom03 prenom03
designation03
PS C:\Windows\system32>
Commands
Modules: All
Refresh
Name:
A:
Add-AppProvisionedSharedPacl
Add-AppSharedPackageContair
Add-AppvClientConnectionGrou
Add-AppvClientPackage
Add-AppvPublishingServer
Add-AppxPackage
Add-AppxProvisionedPackage
Add-AppxVolume
Add-AzureAccount
Add-AzureAnalysisServicesAcco
Add-AzureApplicationGatewayS
Add-AzureCertificate
Add-AzureDataDisk
Add-AzureDisk
Add-AzureDns
Add-AzureEndpoint
Add-AzureEnvironment
Add-AzureHDInsightConfigValu
Add-AzureHDInsightMetastore
Add-AzureHDInsightScriptActio
Add-AzureHDInsightStorage
Add-AzureInternalLoadBalancer
Add-AzureKeyVaultCertificate
Add-AzureKeyVaultCertificateC
Add-AzureKeyVaultKey
Add-AzureKeyVaultManagedStc
Add-AzureNetworkInterfaceCor
Add-AzureNodeWebRole
Add-AzureNodeWorkerRole
Run
Insert
Copy
Completed
Ln 17 Col 25
100%