

Scripting Powershell



Scripting Powershell

Sommaire


Logique Powershell	
- Powershell / Powershell ISE :	3
- Les commandes :	3
Scripts	
- Active directory :	4
→ création UO :	4
→ création utilisateur :	5
- DHCP :	6
→ création d'une pool:	6
- DNS :	7
→ Enregistrement AAA et PTR associé :	7

Utilisation du Document

commande



Information détaillée



Scripting Powershell

Logique Powershell

Powershell / Powershell ISE

Il existe 2 manières de passer des commandes powershell, depuis le powershell classique, et depuis l'ISE qui est une application hôte pour le windows powershell.

Au niveau des commandes ce sont les mêmes, il n'y a pas de changement notable si ce n'est que Powershell ISE met à disposition un espace semblable au bloc note qui permet de travailler son script de manière organisée et plus claire car on peut utiliser plusieurs lignes pour faire une commande. La syntaxe est seulement un peu différente.

Aussi cet hôte permet de tester ses commandes et de les moduler en cas d'échec sans avoir à rappeler la commande comme on le ferait dans le shell.

Les commandes

On distingue les cmdlet des fonctions, les cmdlet sont des commandes powershell natives et non des exécutables, elles sont collectées dans le module powershell, comme par exemple le module Active directory qu'on abordera dans les scripts AD.

Depuis un fichier batch on utilise la commande powershell suivie d'une commande powershell. On peut aussi utiliser des fichiers .ps1 qui est l'extension pour les scripts powershell.

3 types de cmdlet utilisés pour les scripts

- la commande new, qui permet de créer quelque chose de nouveau, une nouvelle UO, un nouvel utilisateur
- la commande Add, qui permet d'ajouter de nouvelles données comme pour créer une étendue DHCP
- la commande Set qui permet de définir des valeurs pour des objets existants.
- la commande Get, qui permet de vérifier les données entrées et de les avoir sous forme de tableau.

Scripting Powershell

Scripts

Active directory

Création UO

Pour créer une UO sur l'active directory il faut en premier lieu importer les modules active directory qui rendent possible les commandes AD

```
Import-Module ActiveDirectory
```

New-ADOrganizationalUnit est un cmdlet qui crée une uo



-Name → nom de l'UO
-ProtectedFromAccidentalDeletion → possibilité de supprimer l'UO

Le script étant implémenté à une application batch il contient des données variables comme le nom de l'uo qui est récupéré par une saisie de l'utilisateur avec %nomUO%

Le paramètre -name permet de spécifier le nom que va recevoir l'UO

Le paramètre -ProtectedFromAccidentalDeletion permet de définir si l'UO est supprimable ou non, ici on définit le paramètre sur false ce qui signifie que l'on peut la supprimer à tout moment.

```
New-ADOrganizationalUnit  
-Name "%nomUO%"  
-ProtectedFromAccidentalDeletion $false
```

Cette commande s'écrit sur une seule ligne comme ci-dessous mais pour des raisons de lisibilité elle a été divisée. Cependant il est possible de l'écrire en plusieurs lignes aussi mais avec une autre syntaxe.

exemple de la commande pour créer l'UO Karate


```
New-ADOrganizationalUnit -Name Karate -ProtectedFromAccidentalDeletion $false
```

Scripting Powershell

création utilisateur


Pour créer un utilisateur sur l'active directory il faut en premier lieu importer les modules active directory si ce n'est pas déjà fait avec le script précédent

```
Import-Module ActiveDirectory
```




-Name	→ prénom nom
-DisplayName	→ nom prénom
-GivenName	→ prénom
-Surname	→ nom
-SamAccountName	→ prénom.nom
-UserPrincipalName	→ initiale du prénom concaténé au nom
-EmailAddress	→ prénom.nom@domaine
-Path	→ chemin vers l'UO
-AccountPassword	→ mot de passe
-ChangePasswordAtLogon	→ changement de mdp à la 1ère connexion
-CannotChangePassword	→ ne peut pas changer de mot de passe
-PasswordNeverExpires	→ le mot de passe n'expire jamais
-Enabled	→ le compte est activé
-Server	→ spécifie le nom complet du serveur (CN)

New-ADUser est aussi une cmdlet qui elle permet la création d'utilisateur sur l'active directory



```
New-ADUser -Name "%nomcomplet%" -DisplayName "%displayname%"  
-GivenName "%prenom%" -Surname "%nom%" -SamAccountName "%sam%"  
-UserPrincipalName "%ouverture%" -EmailAddress "" -Path  
"OU=%ou%,DC=M2L,DC=LAN" -AccountPassword  
(ConvertTo-SecureString "ServESP201808" -AsPlainText -Force)  
-ChangePasswordAtLogon $false -CannotChangePassword $true  
-PasswordNeverExpires $true -Enabled $true -Server  
SRVligue.m2l.lan
```

Exemple pour l'utilisateur florent buteux dans l'uo karate avec le mot de passe Sistr-2023



```
New-ADUser -Name "florent buteux" -DisplayName "buteux florent"  
-GivenName "florent" -Surname "buteux" -SamAccountName  
"%florent.buteux%" -UserPrincipalName "%fbuteux%" -EmailAddress  
"" -Path "OU=karate,DC=M2L,DC=LAN" -AccountPassword  
(ConvertTo-SecureString "Sistr-2023" -AsPlainText -Force)
```

Scripting Powershell

```
-ChangePasswordAtLogon $false -CannotChangePassword $true  
-PasswordNeverExpires $true -Enabled $true -Server  
SRVligue.m21.lan
```

DHCP

création d'un pool

Ce script permet la création d'une étendue DHCP avec toutes les options.

```
Add-DhcpServerv4Scope -Name "%nom%" -StartRange %ipdebut%  
-EndRange %ipfin% -SubnetMask %mask% -State Active  
Set-DhcpServerv4OptionValue -ScopeID %reseau% -OptionID 3  
-Value %passerelle%  
Set-DhcpServerv4OptionValue -DnsServer %DNS%  
Get-DhcpServerv4Scope -ScopeID %reseau%  
Get-DhcpServerv4OptionValue -ScopeId %reseau%
```

on va dans un premier temps spécifier le nom de l'étendue, l'étendue avec une ip de début et une ip de fin, un masque et son état, c'est-à-dire si on l'active tout de suite ou non. Ici on l'active tout de suite.

```
Add-DhcpServerv4Scope -Name "karate" -StartRange 172.16.10.30  
-EndRange 172.16.10.60 -SubnetMask 255.255.255.0 -State Active
```

ensuite on définit l'adresse réseau ainsi que les options ID (3) à savoir le routeur, donc la passerelle.

```
Set-DhcpServerv4OptionValue -ScopeID 172.16.10.0 -OptionID 3  
-Value 172.16.10.252
```

On ajoute aussi l'option DNS qui va permettre aux clients DHCP de connaître leurs DNS

```
Set-DhcpServerv4OptionValue -DnsServer 172.16.3.251
```

Enfin on a les commandes de vérification "GET" qui permettent d'afficher des informations du serveur, ici on demande les étendue du réseau 172.16.10.0 et les options de ce réseau donc le routeur et DNS pour vérifier la bonne configuration et la prise en compte des commandes ci-dessus.

Scripting Powershell

```
Get-DhcpServerv4Scope -ScopeID 172.16.10.0  
Get-DhcpServerv4OptionValue -ScopeId 172.16.10.0
```

DNS

Enregistrement AAA et PTR associé

Ce script permet la création d'enregistrement AAA et du PTR associé, il faut néanmoins que la zone de recherche inversée soit configurée au préalable afin d'assurer la création du PTR.

```
Add-DnsServerResourceRecord -A -Name %nom% -IPv4Address %ip%  
-ZoneName m21.lan -CreatePtr  
Get-DnsServerResourceRecord -ComputerName SRVligue -ZoneName  
m21.lan
```

Exemple pour la création d'un enregistrement AAA pour un pc du vlan karate avec l'adresse 172.16.10.30

```
Add-DnsServerResourceRecord -A -Name karate1 -IPv4Address  
172.16.10.30 -ZoneName m21.lan -CreatePtr
```

Enfin la commande de vérification "GET" qui permet d'afficher des informations du serveur, ici on demande les informations DNS du serveur SRVligue de la zone m21.lan

```
Get-DnsServerResourceRecord -ComputerName SRVligue -ZoneName  
m21.lan
```

Scripting Powershell