

# WINDOWS SERVEUR

**VERSION CORE  
EN POWER SHELL**



Windows Server en version "Core" est une option d'installation minimaliste disponible pour Windows Server. Cette version est conçue pour réduire la surface d'attaque du système et optimiser les performances en supprimant l'interface graphique utilisateur (GUI) et en ne conservant que les composants essentiels nécessaires au fonctionnement du serveur.

## TUTORIEL : CONFIGURATION D'ACTIVE DIRECTORY SUR WINDOWS SERVER AVEC POWERSHELL

### Introduction

*Ce tutoriel a pour but de fournir une série de commandes PowerShell pour configurer un environnement Active Directory sur une machine virtuelle (VM) Windows Server. Nous couvrirons la création d'un domaine Active Directory, l'ajout d'un compte utilisateur, la mise en place d'un poste client, ainsi que la configuration des services DHCP et DNS. Des tests de validation seront également effectués pour s'assurer du bon fonctionnement de la configuration.*

### Prérequis

- *Avoir une VM Windows Server prête à l'emploi.*
- *Avoir les privilèges administrateurs sur la VM.*
- *Avoir une VM Windows 10/11 prête à l'emploi .*
- *Les configurer en réseau interne les deux VMS .*

# I. NOMMER L'ORDINATEUR

Dans cette étape, vous êtes sur le point de changer le nom de votre ordinateur serveur, ce qui est un aspect important de la configuration d'un serveur, surtout lors de son intégration dans un réseau d'entreprise.

## Importance de la Définition d'un Nom d'Ordinateur Serveur

1. Identification Unique : Chaque serveur dans un réseau doit avoir un nom unique pour éviter les conflits et garantir que les ressources réseau peuvent être localisées et identifiées correctement. Ceci est particulièrement crucial dans un domaine où les noms des ordinateurs sont utilisés pour la gestion des politiques, le déploiement de logiciels, et l'accès aux ressources partagées.
2. Facilité de Gestion : Un nom de serveur clair et descriptif rend plus facile pour les administrateurs réseau de reconnaître la fonction du serveur et de gérer le réseau efficacement. Par exemple, un serveur nommé "DC-01" pourrait indiquer qu'il s'agit du premier contrôleur de domaine dans l'infrastructure.
3. Conventions de Nomination : Les entreprises utilisent souvent des conventions de nommage pour organiser et standardiser la façon dont les appareils sont identifiés. Ces conventions peuvent inclure le rôle du serveur, son emplacement physique, ou le numéro de série, facilitant ainsi la maintenance et l'expansion du système d'information.

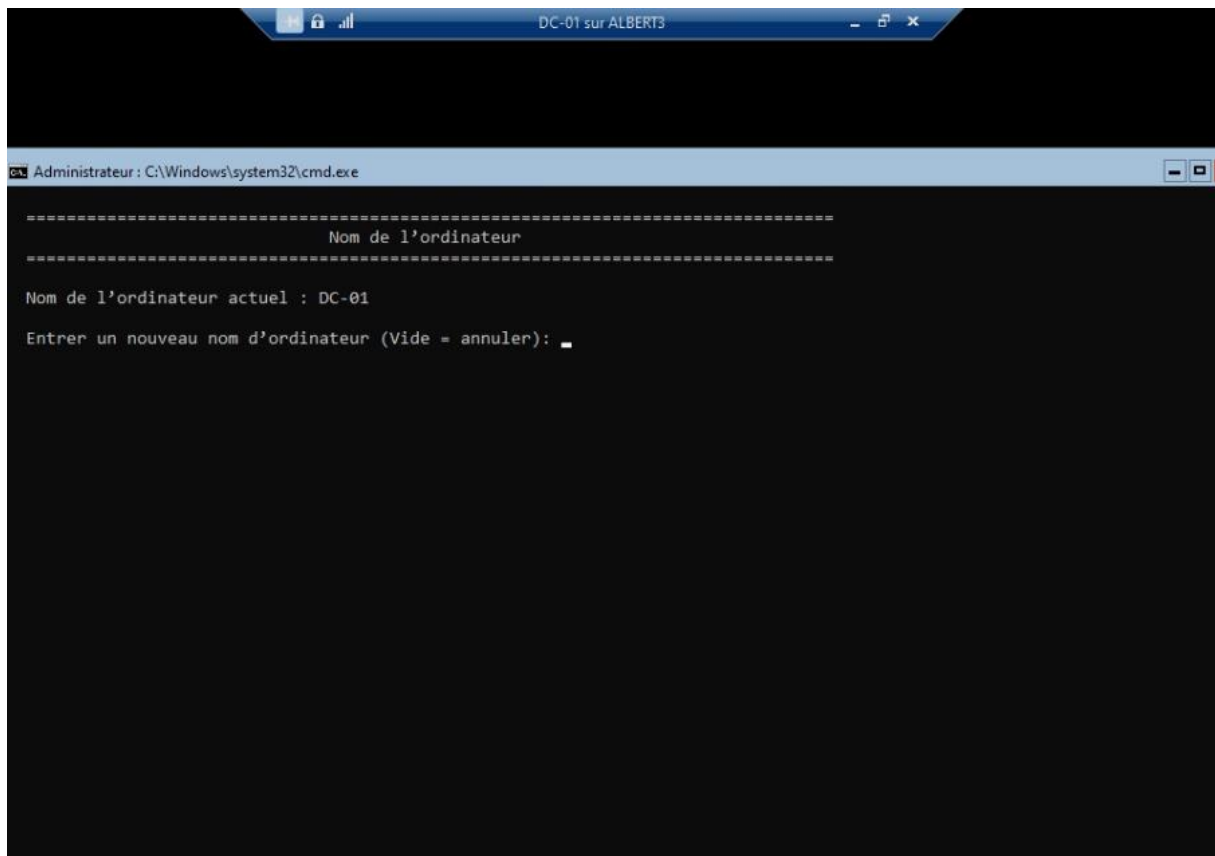
4. Sécurité : Le nom d'ordinateur peut également être utilisé dans le cadre de la sécurité réseau. Par exemple, des politiques de sécurité spécifiques peuvent être appliquées à des serveurs en fonction de leur rôle indiqué dans leur nom.

### Utilité de cette Étape

- Vous pourriez être à l'étape de la préparation initiale de votre serveur, en le renommant avant de le joindre à un domaine ou de configurer des rôles spécifiques comme Active Directory, DNS, ou DHCP.
- Si vous gérez plusieurs serveurs, le nommage est utile pour le suivi des fonctionnalités, des performances, et de l'emplacement de chaque serveur.
- En renommant le serveur à ce stade, vous vous assurez que toutes les conventions et politiques sont respectées dès le début, ce qui réduit la complexité des modifications futures.

Le nom de l'ordinateur que vous choisissez ici devra probablement respecter les politiques de votre organisation et être assez descriptif pour indiquer le but du serveur à toute personne qui le gère ou qui y accède.

## "Maîtrise de PowerShell : Configuration Complète sur Windows Server"



```
DC-01 sur ALBERT3
Administrateur : C:\Windows\system32\cmd.exe
=====
Nom de l'ordinateur
=====
Nom de l'ordinateur actuel : DC-01
Entrer un nouveau nom d'ordinateur (Vide = annuler):
```

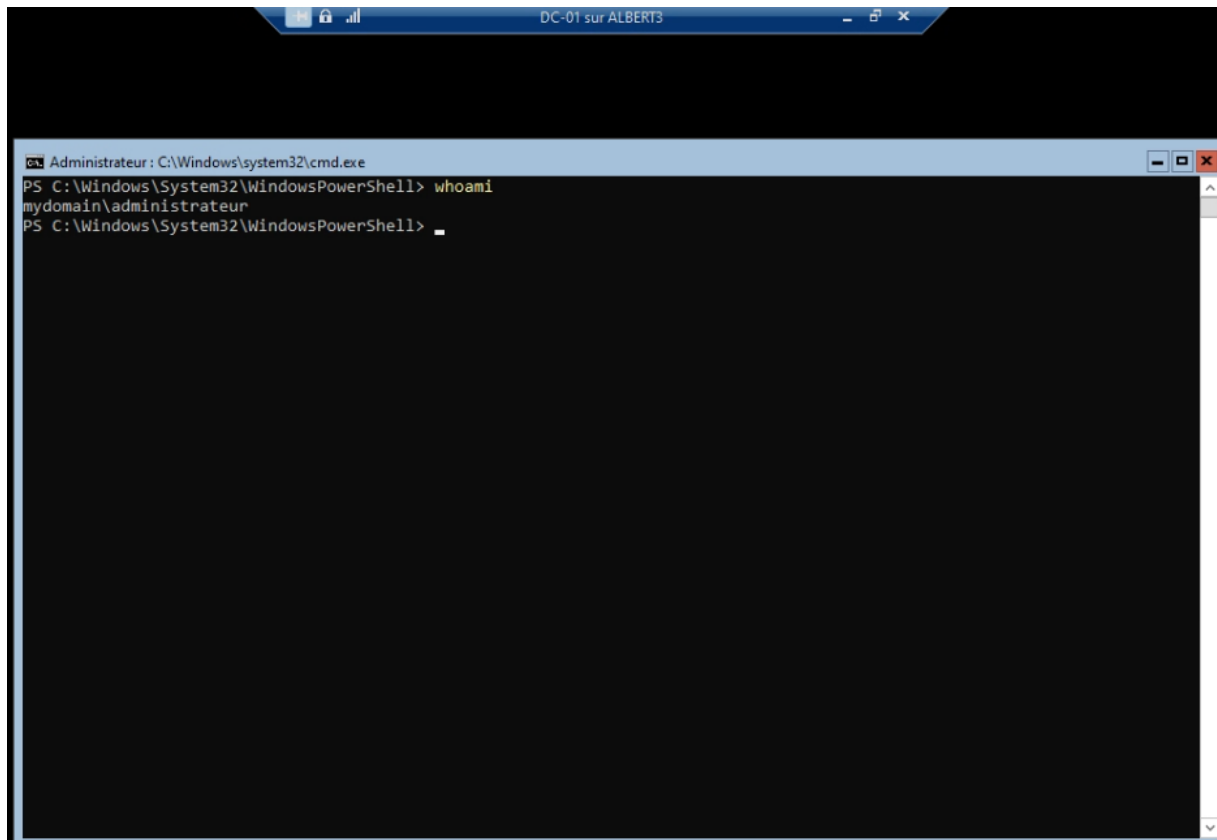
*Cette capture d' écran ci-joint la nomination du serveur maître ici appelé « DC-01 » pour Domain Controller 01*

## II. VERIFICATION DE L'IDENTITE DE LA MACHINE

Vérifier le Nom et le Domaine de la Machine

Pour confirmer que vous travaillez sur le bon serveur et vérifier son appartenance à un domaine, utilisez la commande suivante :

**whoami**

A screenshot of a Windows command prompt window. The title bar reads "Administrateur : C:\Windows\system32\cmd.exe". The window content shows the following text:

```
PS C:\Windows\System32\WindowsPowerShell> whoami  
mydomain\administrateur  
PS C:\Windows\System32\WindowsPowerShell> _
```

*Cette capture d' écran ci-joint la montre qui est actuellement connecté au poste plus précisément à la session*

Cette commande affiche le nom d'utilisateur complet et le domaine, ce qui aide à s'assurer que les modifications système sont effectuées sur la machine correcte.

## III. SE METTRE EN POWERSHELL POUR QUE LES PROMPTS SOIENT PLUS CLAIRS POUR LE SYSTEME

Pour naviguer dans l'arborescence des dossiers de votre serveur via PowerShell et atteindre le répertoire `WindowsPowerShell`, suivez ces étapes :

1. Accéder au Répertoire Windows :

```
PS C:\> cd .\Windows\
```

2. Entrer dans le Dossier System32 :

```
PS C:\Windows> cd .\System32\
```

3. Naviguer vers WindowsPowerShell :

```
PS C:\Windows\System32> cd .\WindowsPowerShell\
```

## "Maîtrise de PowerShell : Configuration Complète sur Windows Server"

```
Administrateur: C:\Windows\system32\cmd.exe
VERTISSEMENT : Pour lancer de nouveau l'outil de configuration du serveur, exécutez « SConfig »
PS C:\Windows\System32> cd..
PS C:\Windows> cd ..
PS C:\> Cd .\Windows\
PS C:\Windows> cd .\System32\
PS C:\Windows\System32> Cd .\WindowsPowerShell\
PS C:\Windows\System32\WindowsPowerShell> _
```

*Cette capture d'écran ci-joint le chemin d'accès Power Shell*

À ce stade, vous êtes dans le dossier `WindowsPowerShell`, où vous pouvez gérer les scripts et modules spécifiques à PowerShell pour votre serveur.



## IV. CONFIGURATION D'UNE ADRESSE IP STATIQUE SUR UN SERVEUR WINDOWS

Configurer une adresse IP statique sur un serveur est essentiel pour garantir la fiabilité et la stabilité de la communication au sein du réseau. Une adresse IP statique assure que l'adresse du serveur ne change pas, contrairement à une adresse IP dynamique attribuée via DHCP. Cela est particulièrement crucial pour un serveur maître qui héberge des services critiques tels que l'Active Directory, DNS, et d'autres services réseau.

### Configuration Réseau

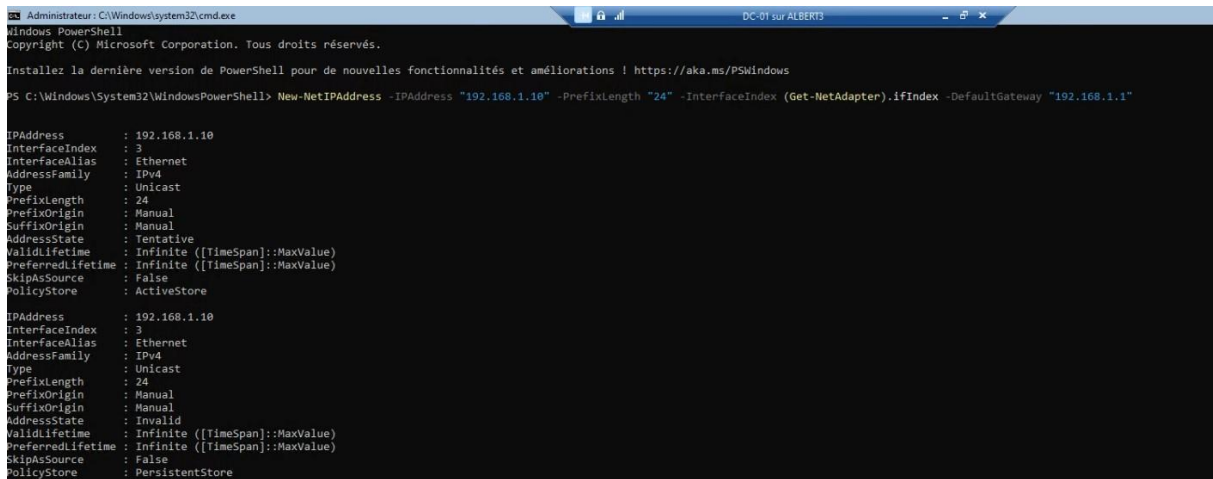
#### Attribution d'une Adresse IP Statique

Pour attribuer une adresse IP statique, nous utiliserons les commandes PowerShell suivantes. Chaque commande joue un rôle spécifique dans la configuration de l'adresse IP du serveur et de ses paramètres réseau.

Définir l'Adresse IP et le Masque de Sous-Réseau :

```
New-NetIPAddress -IPAddress "192.168.1.10" -PrefixLength 24 -  
InterfaceIndex (Get-NetAdapter).ifIndex -DefaultGateway  
"192.168.1.1"
```

## "Maîtrise de PowerShell : Configuration Complète sur Windows Server"



```
Administrateur: C:\Windows\system32\cmd.exe
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

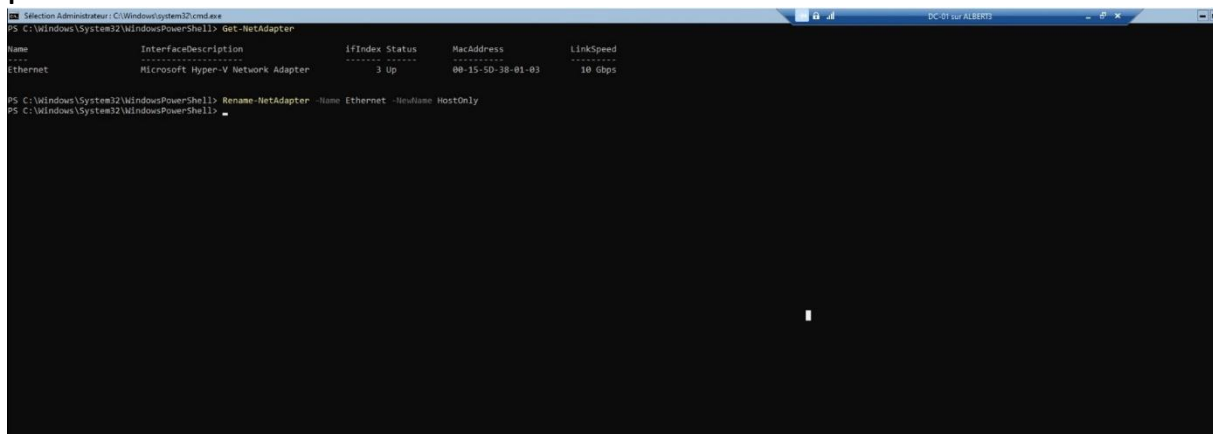
PS C:\Windows\System32\WindowsPowerShell> New-NetIPAddress -IPAddress "192.168.1.10" -PrefixLength "24" -InterfaceIndex (Get-NetAdapter).ifIndex -DefaultGateway "192.168.1.1"

IPAddress          : 192.168.1.10
InterfaceIndex     : 3
InterfaceAlias     : Ethernet
AddressFamily      : IPv4
Type               : Unicast
PrefixLength       : 24
PrefixOrigin       : Manual
SuffixOrigin       : Manual
AddressState       : Tentative
ValidLifetime      : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime  : Infinite ([TimeSpan]::MaxValue)
SkipAsSource       : False
PolicyStore        : ActiveStore

IPAddress          : 192.168.1.10
InterfaceIndex     : 3
InterfaceAlias     : Ethernet
AddressFamily      : IPv4
Type               : Unicast
PrefixLength       : 24
PrefixOrigin       : Manual
SuffixOrigin       : Manual
AddressState       : Invalid
ValidLifetime      : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime  : Infinite ([TimeSpan]::MaxValue)
SkipAsSource       : False
PolicyStore        : PersistentStore
```

*Cette capture d' écran ci-joint la montre qui est actuellement connecté au poste plus précisément à la session*

- `New-NetIPAddress` : Crée une nouvelle entrée d'adresse IP pour un adaptateur réseau. Cette commande est utilisée pour attribuer manuellement une adresse IP statique à l'adaptateur.
- `-IPAddress` : Spécifie l'adresse IP à attribuer. Ici, `"192.168.1.10"` est l'adresse choisie, typique pour un réseau local.
- `-PrefixLength` : Définit la longueur du préfixe du masque de sous-réseau en notation CIDR. `"24"` équivaut à un masque de sous-réseau de `255.255.255.0`, indiquant que les adresses IP de `192.168.1.1` à `192.168.1.254` sont dans le même sous-réseau.
- `-InterfaceIndex` : Identifie l'adaptateur réseau cible par son index. `(Get-NetAdapter).ifIndex` récupère dynamiquement l'index de l'adaptateur réseau actif.
- `-DefaultGateway` : La passerelle par défaut, ici `"192.168.1.1"`, est l'adresse IP du routeur ou du dispositif de sortie du sous-réseau, permettant la communication avec d'autres réseaux.



```
Sélection Administrateur: C:\Windows\system32\cmd.exe
PS C:\Windows\System32\WindowsPowerShell> Get-NetAdapter

Name          InterfaceDescription      ifIndex Status      MacAddress      LinkSpeed
-----
Ethernet      Microsoft Hyper-V Network Adapter  3 Up         00-15-5D-38-01-03 10 Gbps

PS C:\Windows\System32\WindowsPowerShell> Rename-NetAdapter -Name Ethernet -NewName HostOnly
PS C:\Windows\System32\WindowsPowerShell>
```

Configurer le Serveur DNS :

```
Set-DnsClientServerAddress -InterfaceIndex (Get-NetAdapter).ifIndex -ServerAddresses "127.0.0.1"
```

- `Set-DnsClientServerAddress`` : Configure l'adresse du serveur DNS pour l'adaptateur réseau spécifié.

- `-ServerAddresses`` : L'adresse IP du serveur DNS. Utiliser `"127.0.0.1"` (l'adresse de bouclage ou localhost) indique que ce serveur utilise son propre service DNS. C'est une pratique courante pour les serveurs qui gèrent le rôle DNS.

3. Renommer l'Adaptateur Réseau :

```
Rename-NetAdapter -Name "Ethernet0" -NewName "HostOnly"
```

- `Rename-NetAdapter`` : Change le nom d'un adaptateur réseau, ce qui peut aider à identifier clairement l'usage ou le rôle de l'adaptateur dans des configurations réseau complexes.

- `-Name`` et `-NewName`` : Spécifient respectivement le nom actuel de l'adaptateur et le nouveau nom à lui attribuer. Ici, l'adaptateur nommé `"Ethernet0"` est renommé en `"HostOnly"`, suggérant que cet adaptateur est utilisé pour des communications isolées ou spécifiques au serveur.

Configurer une adresse IP statique sur un serveur maître est une étape fondamentale pour assurer une infrastructure réseau stable et fiable. Les adresses IP statiques facilitent la gestion des services réseau, la configuration de la sécurité, et la résolution des noms de domaine, contribuant à l'efficacité globale de l'administration système. Les commandes PowerShell fournies ici offrent une méthode puissante et flexible pour accomplir cette tâche essentielle.



```
Try{  
    Add-WindowsFeature -Name $Feature -  
    IncludeManagementTools -IncludeAllSubFeature  
    Write-Output" $Feature : Installation  
    approuvée !"  
}Catch{  
    Write-Output" $Feature :Non et c'est  
    dommage!"  
    }  
} # if(((Get-WindowsFeature -Name  
$Feature).InstallState) -eq "Available")  
} # Foreach($Feature in $FeatureList
```

**Initialisation de la Liste des Fonctionnalités :** `$FeatureList = @("AD-Domain-Services", "DNS")` crée un tableau avec les éléments "AD-Domain-Services" et "DNS". Ce sont les fonctionnalités que le script cherchera à installer.

**Parcours des Fonctionnalités :** La boucle **foreach** passe en revue chaque fonctionnalité listée dans `$FeatureList`.

**Vérification de l'État de la Fonctionnalité :** La commande **(Get-WindowsFeature -Name \$Feature).InstallState** récupère l'état d'installation actuel de la fonctionnalité désignée par `$Feature`.

**Condition d'Installation :** L'expression **-eq "Available"** compare l'état de la fonctionnalité à "Available", ce qui indique que la fonctionnalité peut être installée.

**Affichage de la Disponibilité :** **Write-Output "Feature \$Feature A Présent disponible !"** si une fonctionnalité est disponible, un message est affiché pour informer l'utilisateur.

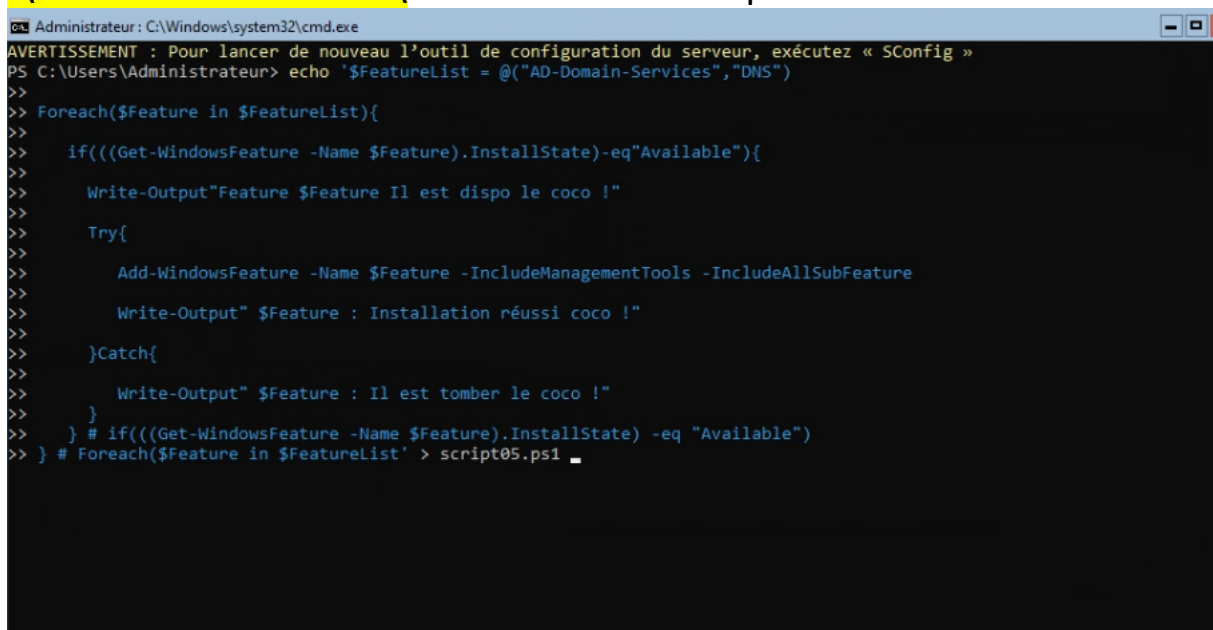
**Tentative d'Installation** : Le bloc **try** contient la commande **Add-WindowsFeature -Name \$Feature - IncludeManagementTools -IncludeAllSubFeature** qui lance l'installation de la fonctionnalité avec tous les outils de gestion nécessaires.

**Confirmation d'Installation** : En cas de succès, **Write-Output " \$Feature : Installation approuvée !"** affiche un message indiquant que l'installation a été réussie.

**Gestion des Échecs d'Installation** : Si une exception se produit pendant l'installation, le bloc **catch** capture l'erreur et **Write-Output " \$Feature : Non et c'est dommage!"** affiche un message d'échec.

Cette séquence scriptée vérifie la disponibilité des rôles avant de tenter leur installation, optimisant le processus de configuration.

Ce script est à configurer dans **PS C:\Windows\System32> cd .\WindowsPowerShell\** il à été nommé script05 comme ceci :



```
Administrateur: C:\Windows\system32\cmd.exe
AVERTISSEMENT : Pour lancer de nouveau l'outil de configuration du serveur, exécutez « SConfig »
PS C:\Users\Administrateur> echo '$FeatureList = @("AD-Domain-Services","DNS")'
>>
>> Foreach($Feature in $FeatureList){
>>
>>   if(((Get-WindowsFeature -Name $Feature).InstallState)-eq"Available"){
>>
>>     Write-Output"Feature $Feature Il est dispo le coco !"
>>
>>     Try{
>>
>>       Add-WindowsFeature -Name $Feature -IncludeManagementTools -IncludeAllSubFeature
>>
>>       Write-Output" $Feature : Installation réussi coco !"
>>
>>     }Catch{
>>
>>       Write-Output" $Feature : Il est tomber le coco !"
>>     }
>>   } # if(((Get-WindowsFeature -Name $Feature).InstallState) -eq "Available")
>> } # Foreach($Feature in $FeatureList' > script05.ps1
```

Ensuite notre :

```
AVERTISSEMENT : Pour lancer de nouveau l'outil de configuration du serveur, exécutez « S
PS C:\Users\Administrateur> echo '$FeatureList = @("AD-Domain-Services","DNS")
>>
>> Foreach($Feature in $FeatureList){
>>
>>   if(((Get-WindowsFeature -Name $Feature).InstallState)-eq"Available"){
>>
>>     Write-Output"Feature $Feature Il est dispo le coco !"
>>
>>     Try{
>>
>>       Add-WindowsFeature -Name $Feature -IncludeManagementTools -IncludeAllSubFeatu
>>
>>       Write-Output" $Feature : Installation réussi coco !"
>>
>>     }Catch{
>>
>>       Write-Output" $Feature : Il est tomber le coco !"
>>     }
>>   } # if(((Get-WindowsFeature -Name $Feature).InstallState) -eq "Available")
>> } # Foreach($Feature in $FeatureList' > script05.ps1
PS C:\Users\Administrateur> .\script05.ps1
```

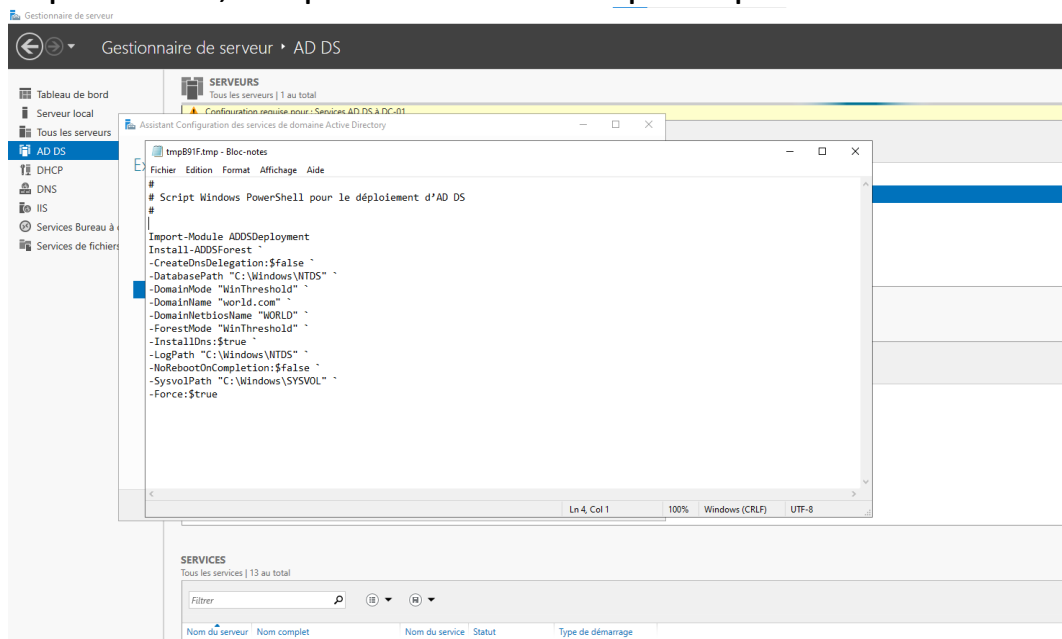
`.\script05.ps1` pour exécuter le script  
Et ensuite le tour est joué.



# VI. CONFIGURATION DE LA FORET AD

Création de la Forêt et du Domaine :

Configurez votre forêt et domaine Active Directory en utilisant le script suivant, adapté à vos besoins spécifiques :



Screen prit sur

## Import-Module ADDSDeployment

```
$ForestConfiguration = @{  
    'DatabasePath'      = 'C:\Windows\NTDS'  
    'DomainMode'       = 'Default'  
    'DomainName'       = 'Mydomain.lab'  
    'DomainNetbiosName' = 'Mydomain'  
    'ForestMode'      = 'Default'  
    'InstallDns'      = $true  
    'LogPath'         = 'C:\Windows\NTDS'  
    'NoRebootOnCompletion' = $false  
    'SysvolPath'      = 'C:\Windows\SYSVOL'
```



```
'Force' = $true  
'CreateDnsDelegation' = $false  
}
```

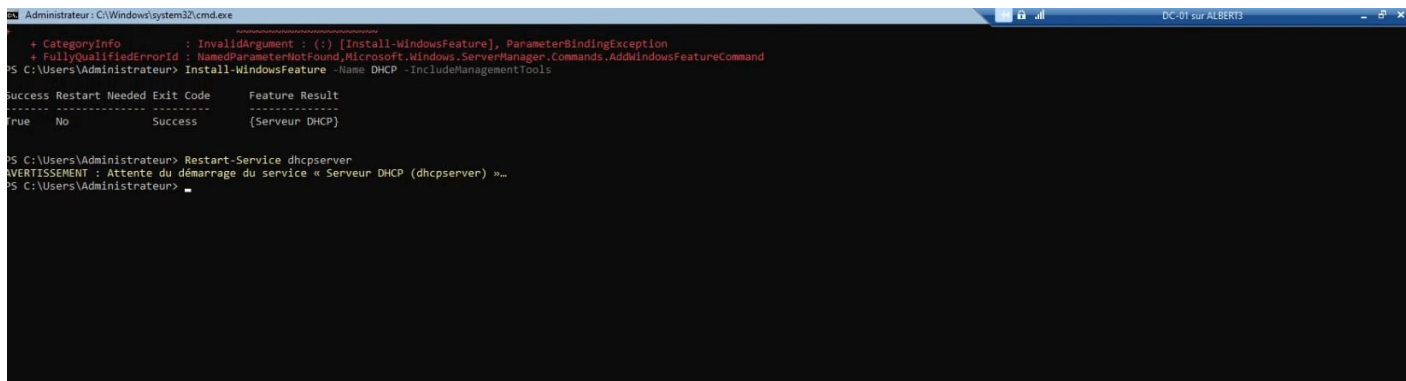
## Install-ADDSTree @ForestConfiguration

Ce script configure les chemins pour la base de données AD, le SYSVOL, installe le service DNS, et précise les noms DNS et NetBIOS du domaine.

# VII. CONFIGURATION DU SERVEUR DHCP

Installation et Configuration du Serveur DHCP :

Installez le rôle DHCP et configurez le service DHCP pour la distribution des adresses IP.

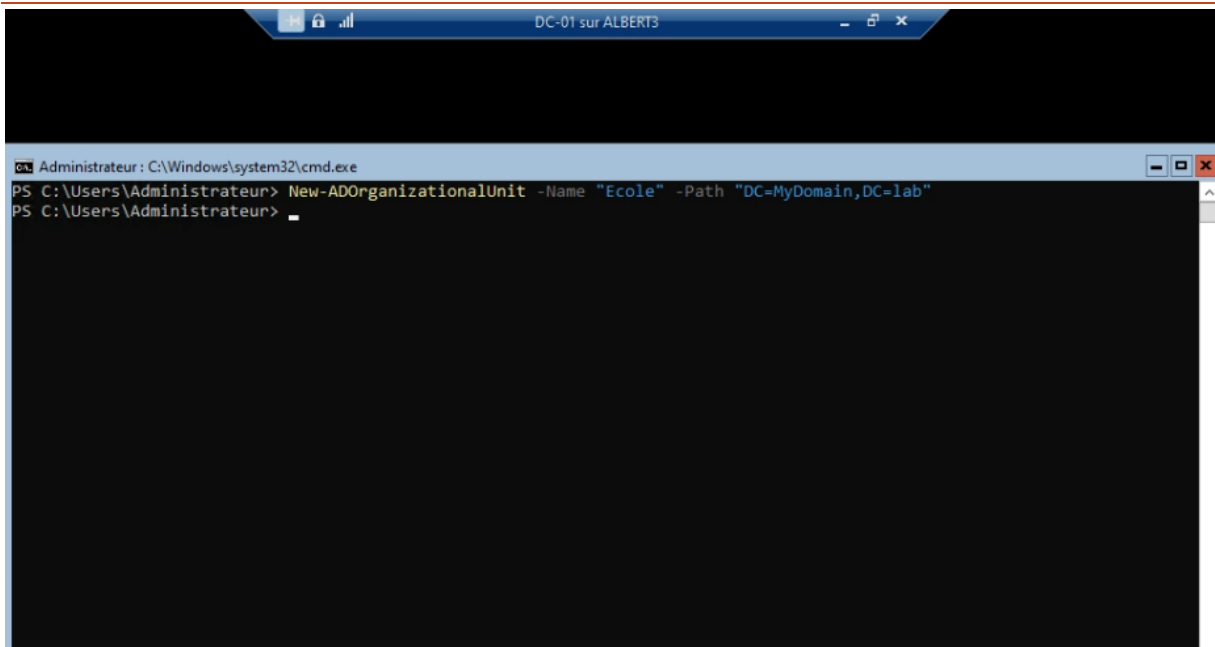


```
Administrateur : C:\Windows\system32\cmd.exe  
+ CategoryInfo          : InvalidArgument : (?) [Install-WindowsFeature], ParameterBindingException  
+ FullyQualifiedErrorId : NamedParameterNotFound,Microsoft.Windows.ServerManager.Commands.AddWindowsFeatureCommand  
PS C:\Users\Administrateur> Install-WindowsFeature -Name DHCP -IncludeManagementTools  
Success Restart Needed Exit Code      Feature Result  
-----  
True     No             Success      (Serveur DHCP)  
  
PS C:\Users\Administrateur> Restart-Service dhcpserver  
AVERTISSEMENT : Attente du démarrage du service « Serveur DHCP (dhcpserver) »...  
PS C:\Users\Administrateur>
```

```
Install-WindowsFeature -Name DHCP -IncludeManagementTools  
Restart-Service dhcpserver
```

```
Add-DhcpServerv4Scope -Name "DHCP" -StartRange 192.168.1.50 -  
EndRange 192.168.1.100 -SubnetMask 255.255.255.0  
Add-DHCPServerV4ExclusionRange -ScopelId 192.168.1.0 -StartRange  
192.168.1.70 -EndRange 192.168.1.75  
Set-DhcpServerv4OptionDefinition -OptionId 3 -DefaultValue  
192.168.1.254  
Set-DhcpServerv4OptionDefinition -OptionId 6 -DefaultValue  
192.168.1.253  
Set-DhcpServerv4OptionDefinition -OptionId 15 -DefaultValue  
Mydomain.lab  
Get-DhcpServerv4OptionDefinition  
Set-DhcpServerv4Scope -ScopelId 172.16.0 -Name "DHCP" -State  
Active  
netsh dhcp  
Restart-Service dhcpserver  
Add-DhcpServerInDC -DnsName DHCP01 -IPAddress 192.168.1.200  
Get-DhcpServerv4Scope
```

## VIII. CREATION OU



```
DC-01 sur ALBERTS  
Administrateur : C:\Windows\system32\cmd.exe  
PS C:\Users\Administrateur> New-ADOrganizationalUnit -Name "Ecole" -Path "DC=MyDomain,DC=lab"  
PS C:\Users\Administrateur>
```

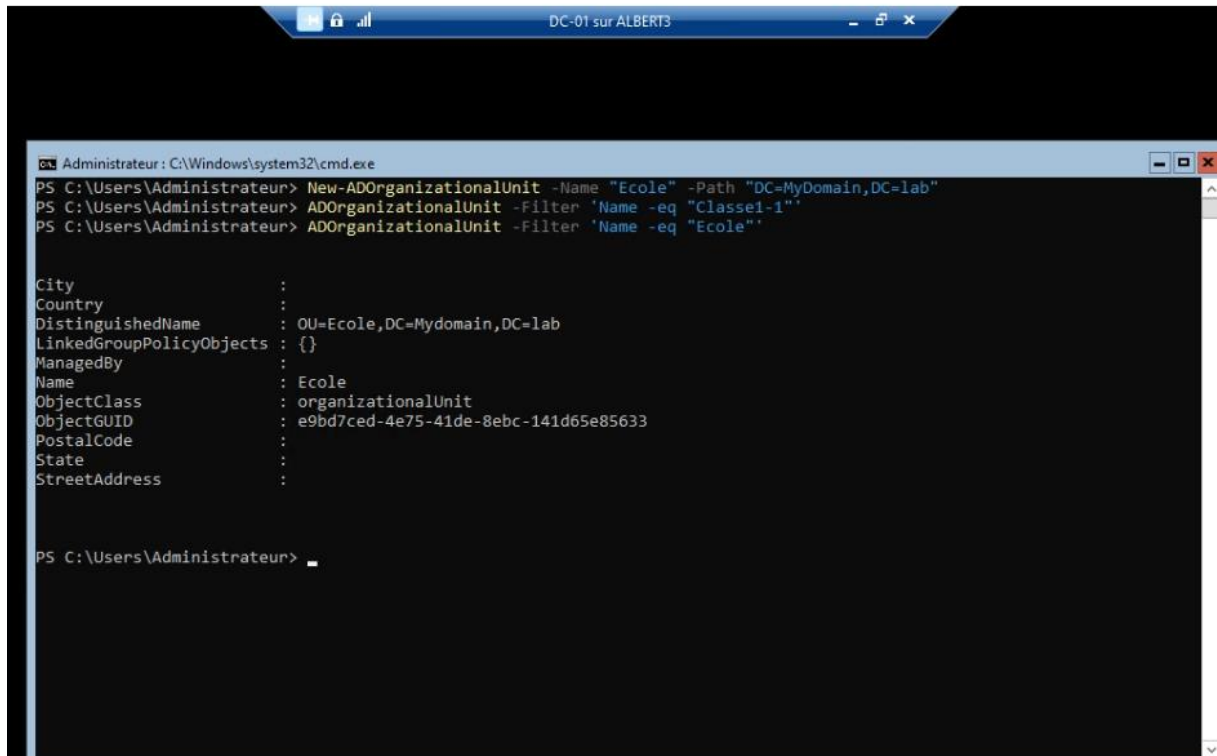
```
New-ADOrganizationalUnit -Name Ecole -Path  
"DC=Mydomain,DC=lab"
```

Comment Vérifier la Création d'une Unité Organisationnelle dans Active Directory avec PowerShell

Après avoir créé une unité organisationnelle (OU) dans Active Directory à l'aide de PowerShell, il est important de vérifier si l'opération a été réalisée avec succès. Ce tutoriel vous guidera à travers les étapes pour confirmer la création de votre OU. Nous avons utilisé l'exemple de création d'une OU nommée "Ecole". Voici comment vous pouvez procéder pour la vérification :

```
Get-ADOrganizationalUnit -Filter 'Name -eq "Ecole"'
```

Get-ADGroup -Filter \* | Select-Object Name,  
DistinguishedName



```
Administrateur : C:\Windows\system32\cmd.exe
PS C:\Users\Administrateur> New-ADOrganizationalUnit -Name "Ecole" -Path "DC=MyDomain,DC=lab"
PS C:\Users\Administrateur> ADOrganizationalUnit -Filter 'Name -eq "Classe1-1"'
PS C:\Users\Administrateur> ADOrganizationalUnit -Filter 'Name -eq "Ecole"'

City           :
Country        :
DistinguishedName : OU=Ecole,DC=Mydomain,DC=lab
LinkedGroupPolicyObjects : {}
ManagedBy     :
Name           : Ecole
ObjectClass    : organizationalUnit
ObjectGUID     : e9bd7ced-4e75-41de-8ebc-141d65e85633
PostalCode     :
State          :
StreetAddress   :

PS C:\Users\Administrateur> _
```

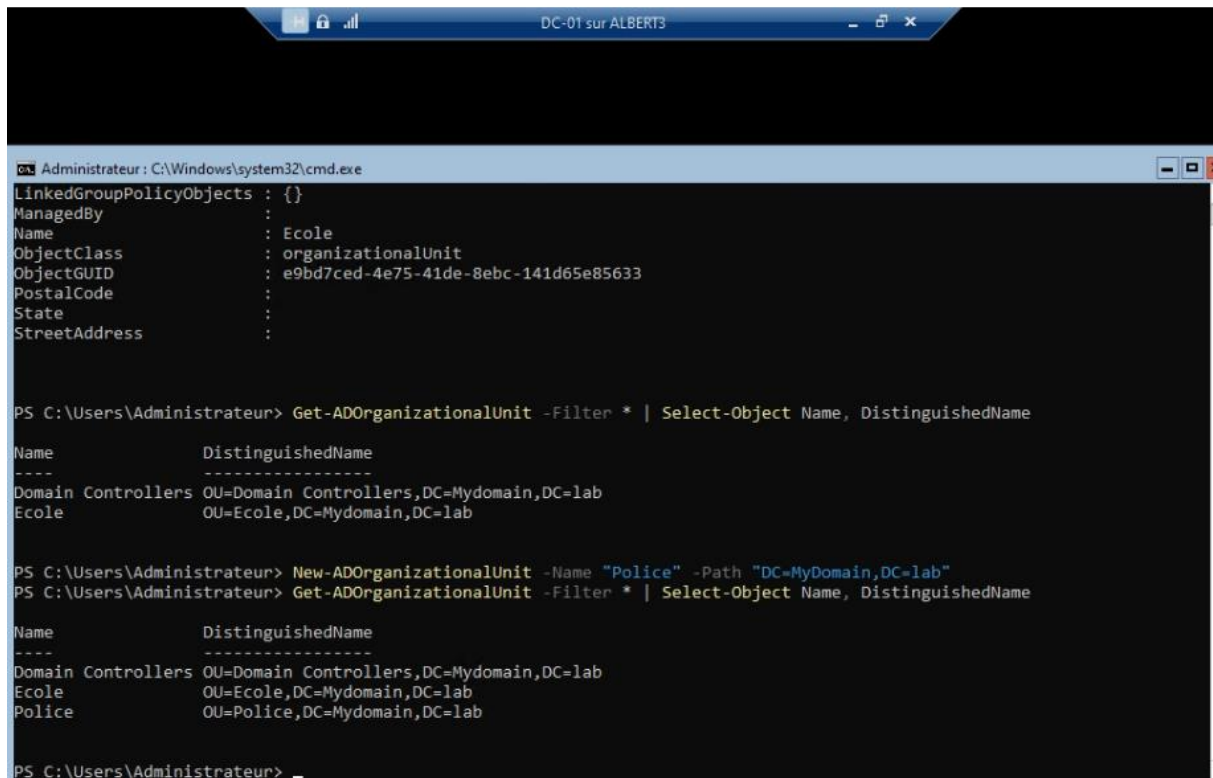
Si des informations sur l'OU apparaissent, cela signifie que la création a été réussie.

Si aucun résultat n'est retourné, cela indique que l'OU n'a pas été trouvée, ce qui peut signifier qu'elle n'a pas été créée ou que le nom/chemin spécifié lors de la vérification est incorrect.

Pour voir toutes les Unités d'Organisation (OU) créées dans votre domaine Active Directory avec PowerShell, vous pouvez utiliser la cmdlet **Get-ADOrganizationalUnit**. Voici comment :

Get-ADOrganizationalUnit -Filter \* | Select-Object Name,  
DistinguishedName

## "Maîtrise de PowerShell : Configuration Complète sur Windows Server"



```
Administrateur : C:\Windows\system32\cmd.exe
LinkedGroupPolicyObjects : {}
ManagedBy                :
Name                      : Ecole
ObjectClass               : organizationalUnit
ObjectGUID                : e9bd7ced-4e75-41de-8ebc-141d65e85633
PostalCode                :
State                    :
StreetAddress              :

PS C:\Users\Administrateur> Get-ADOrganizationalUnit -Filter * | Select-Object Name, DistinguishedName

Name                      DistinguishedName
----                      -
Domain Controllers      OU=Domain Controllers,DC=Mydomain,DC=lab
Ecole                   OU=Ecole,DC=Mydomain,DC=lab

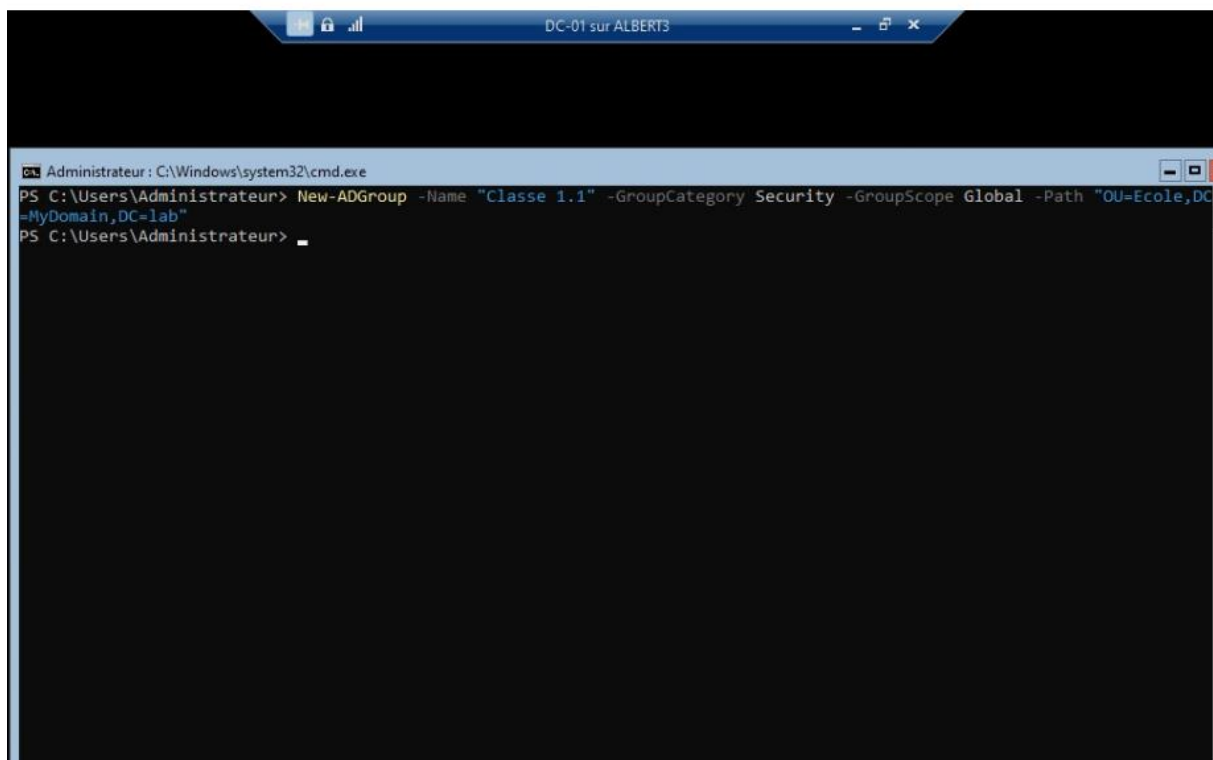
PS C:\Users\Administrateur> New-ADOrganizationalUnit -Name "Police" -Path "DC=MyDomain,DC=lab"
PS C:\Users\Administrateur> Get-ADOrganizationalUnit -Filter * | Select-Object Name, DistinguishedName

Name                      DistinguishedName
----                      -
Domain Controllers      OU=Domain Controllers,DC=Mydomain,DC=lab
Ecole                   OU=Ecole,DC=Mydomain,DC=lab
Police                  OU=Police,DC=Mydomain,DC=lab

PS C:\Users\Administrateur>
```

## IX. CREER UN GROUPE

Pour créer un groupe dans une unité organisationnelle (OU) spécifique en utilisant PowerShell, vous pouvez suivre ces étapes simples. Cela implique l'utilisation de la commande **New-ADGroup** du module Active Directory. Avant de commencer, assurez-vous d'avoir les droits nécessaires pour effectuer des modifications dans Active Directory et que le module Active Directory est installé et disponible dans votre session PowerShell.



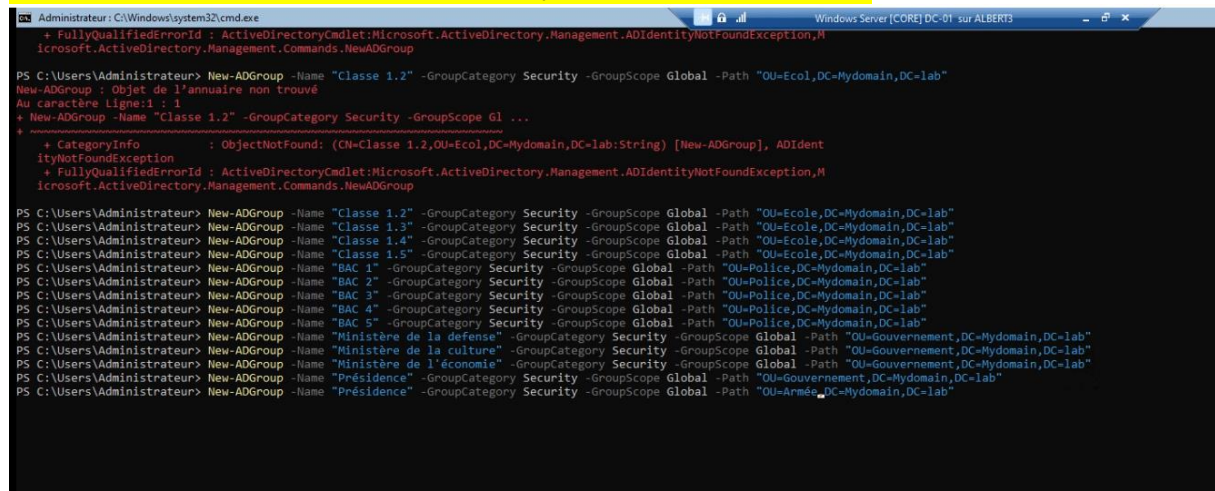
```
DC-01 sur ALBERT3
Administrateur: C:\Windows\system32\cmd.exe
PS C:\Users\Administrateur> New-ADGroup -Name "Classe 1.1" -GroupCategory Security -GroupScope Global -Path "OU=Ecole,DC=MyDomain,DC=lab"
PS C:\Users\Administrateur>
```

Ici présent la création de la classe 1.1 dans l'ou école

```
New-ADGroup -Name "Classe 1.1" -GroupCategory Security -
GroupScope Global -Path "OU=Ecole
Ingétis,DC=Mydomain,DC=lab"
```

## "Maîtrise de PowerShell : Configuration Complète sur Windows Server"

```
New-ADGroup -Name "Ministère de l'économie" -  
GroupCategory Security -GroupScope Global -Path  
"OU=Gouvernement,DC=Mydomain,DC=lab"
```

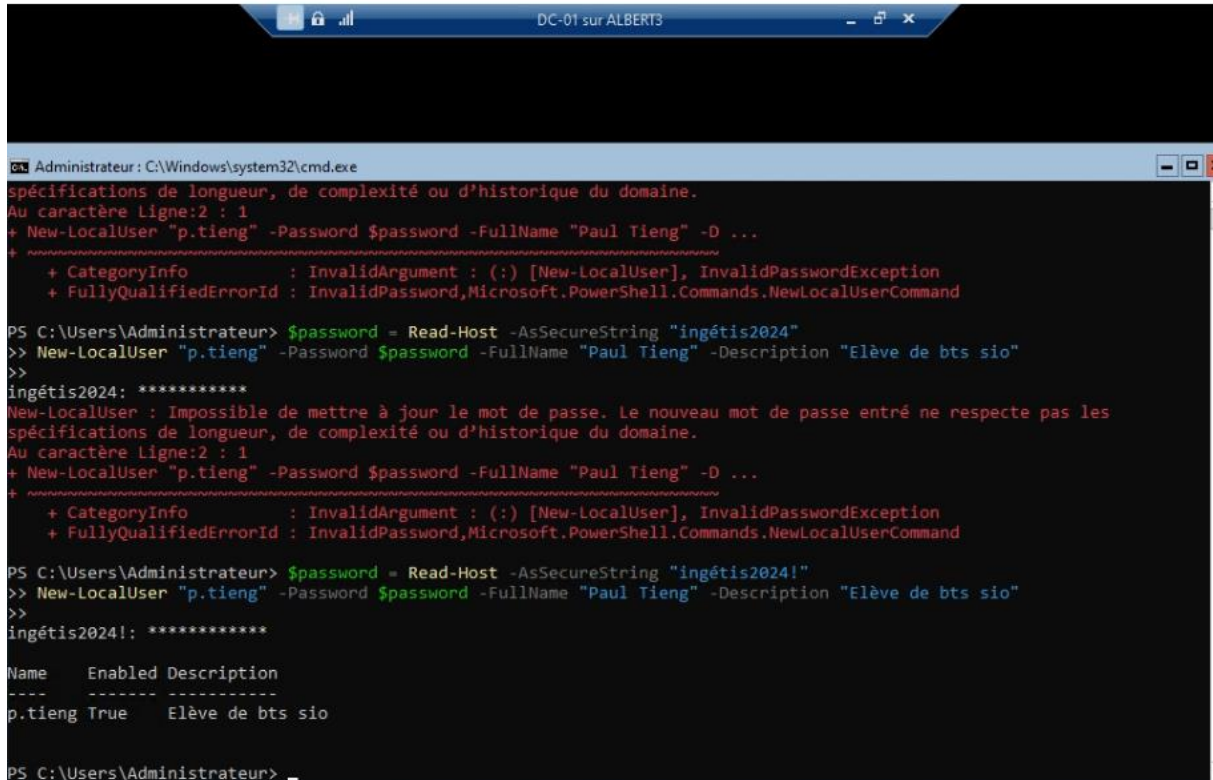


```
Administrateur : C:\Windows\system32\cmd.exe
+ FullyQualifiedErrorId : ActiveDirectoryCmdlet:Microsoft.ActiveDirectory.Management.ADIdentityNotFoundException,Microsoft.ActiveDirectory.Management.Commands.NewADGroup

PS C:\Users\Administrateur> New-ADGroup -Name "Classe 1.2" -GroupCategory Security -GroupScope Global -Path "OU=Ecole,DC=Mydomain,DC=lab"
New-ADGroup : Objet de l'annuaire non trouvé
Au caractère ligne:1 : 1
+ New-ADGroup -Name "Classe 1.2" -GroupCategory Security -GroupScope Gl ...
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (CN=Classe 1.2,OU=Ecole,DC=Mydomain,DC=lab:String) [New-ADGroup], ADIdent
ityNotFoundException
+ FullyQualifiedErrorId : ActiveDirectoryCmdlet:Microsoft.ActiveDirectory.Management.ADIdentityNotFoundException,Microsoft.ActiveDirectory.Management.Commands.NewADGroup

PS C:\Users\Administrateur> New-ADGroup -Name "Classe 1.2" -GroupCategory Security -GroupScope Global -Path "OU=Ecole,DC=Mydomain,DC=lab"
PS C:\Users\Administrateur> New-ADGroup -Name "Classe 1.3" -GroupCategory Security -GroupScope Global -Path "OU=Ecole,DC=Mydomain,DC=lab"
PS C:\Users\Administrateur> New-ADGroup -Name "Classe 1.4" -GroupCategory Security -GroupScope Global -Path "OU=Ecole,DC=Mydomain,DC=lab"
PS C:\Users\Administrateur> New-ADGroup -Name "Classe 1.5" -GroupCategory Security -GroupScope Global -Path "OU=Ecole,DC=Mydomain,DC=lab"
PS C:\Users\Administrateur> New-ADGroup -Name "BAC 1" -GroupCategory Security -GroupScope Global -Path "OU=Police,DC=Mydomain,DC=lab"
PS C:\Users\Administrateur> New-ADGroup -Name "BAC 2" -GroupCategory Security -GroupScope Global -Path "OU=Police,DC=Mydomain,DC=lab"
PS C:\Users\Administrateur> New-ADGroup -Name "BAC 3" -GroupCategory Security -GroupScope Global -Path "OU=Police,DC=Mydomain,DC=lab"
PS C:\Users\Administrateur> New-ADGroup -Name "BAC 4" -GroupCategory Security -GroupScope Global -Path "OU=Police,DC=Mydomain,DC=lab"
PS C:\Users\Administrateur> New-ADGroup -Name "BAC 5" -GroupCategory Security -GroupScope Global -Path "OU=Police,DC=Mydomain,DC=lab"
PS C:\Users\Administrateur> New-ADGroup -Name "Ministère de la defense" -GroupCategory Security -GroupScope Global -Path "OU=Gouvernement,DC=Mydomain,DC=lab"
PS C:\Users\Administrateur> New-ADGroup -Name "Ministère de la culture" -GroupCategory Security -GroupScope Global -Path "OU=Gouvernement,DC=Mydomain,DC=lab"
PS C:\Users\Administrateur> New-ADGroup -Name "Ministère de l'économale" -GroupCategory Security -GroupScope Global -Path "OU=Gouvernement,DC=Mydomain,DC=lab"
PS C:\Users\Administrateur> New-ADGroup -Name "Présidence" -GroupCategory Security -GroupScope Global -Path "OU=Gouvernement,DC=Mydomain,DC=lab"
```

## X. CREER UN UTILISATEUR :



```
Administrateur: C:\Windows\system32\cmd.exe
spécifications de longueur, de complexité ou d'historique du domaine.
Au caractère Ligne:2 : 1
+ New-LocalUser "p.tieng" -Password $password -FullName "Paul Tieng" -D ...
+ ~~~~~
+ CategoryInfo          : InvalidArgument : (:) [New-LocalUser], InvalidPasswordException
+ FullyQualifiedErrorId : InvalidPassword,Microsoft.PowerShell.Commands.NewLocalUserCommand

PS C:\Users\Administrateur> $password = Read-Host -AsSecureString "ingétis2024"
>> New-LocalUser "p.tieng" -Password $password -FullName "Paul Tieng" -Description "Elève de bts sio"
>>
ingétis2024: *****
New-LocalUser : Impossible de mettre à jour le mot de passe. Le nouveau mot de passe entré ne respecte pas les
spécifications de longueur, de complexité ou d'historique du domaine.
Au caractère Ligne:2 : 1
+ New-LocalUser "p.tieng" -Password $password -FullName "Paul Tieng" -D ...
+ ~~~~~
+ CategoryInfo          : InvalidArgument : (:) [New-LocalUser], InvalidPasswordException
+ FullyQualifiedErrorId : InvalidPassword,Microsoft.PowerShell.Commands.NewLocalUserCommand

PS C:\Users\Administrateur> $password = Read-Host -AsSecureString "ingétis2024!"
>> New-LocalUser "p.tieng" -Password $password -FullName "Paul Tieng" -Description "Elève de bts sio"
>>
ingétis2024!: *****

Name      Enabled Description
-----
p.tieng   True      Elève de bts sio

PS C:\Users\Administrateur>
```

```
$password = Read-Host -AsSecureString "Entrez le nouveau mot de
passe"
```

```
Set-LocalUser -Name "NomUtilisateur" -Password $password
```

Dans notre exemple :

```
$password = Read-Host -AsSecureString "ingétis2024!"
New-LocalUser "p.tieng" -Password $password -FullName "Paul
Tieng" -Description "Elève de bts sio"
```



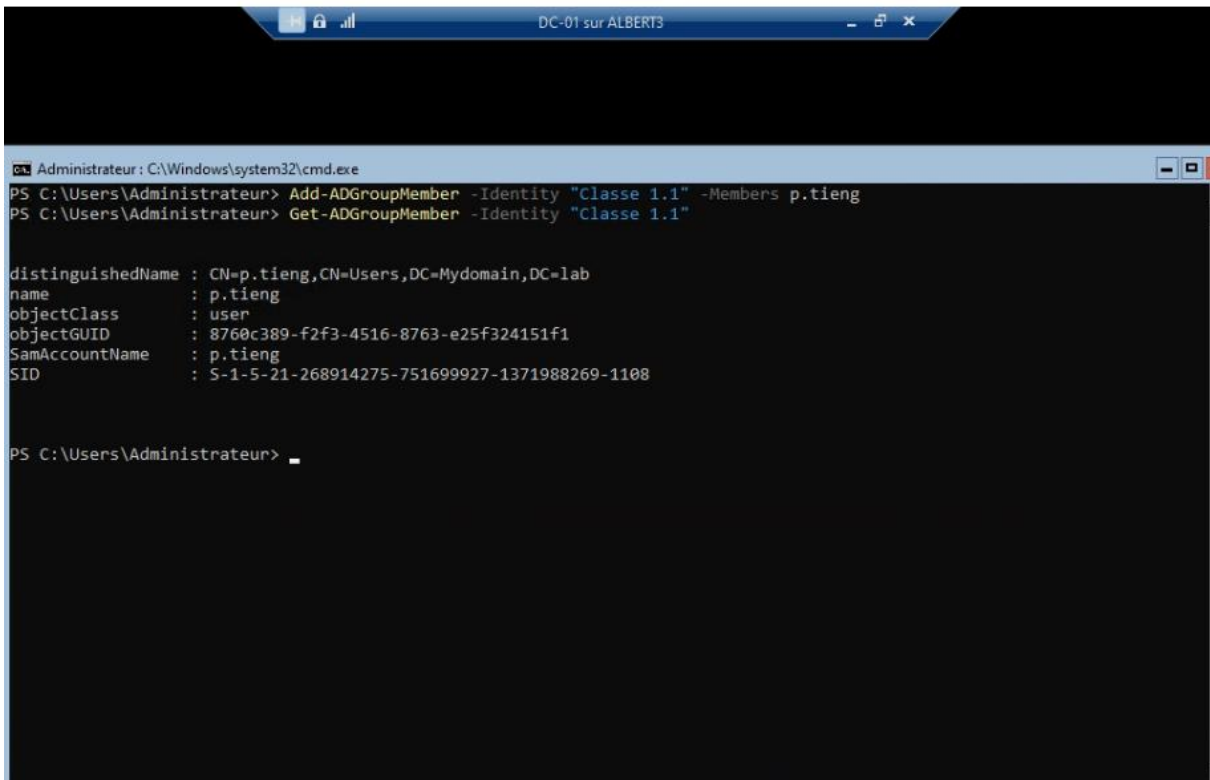
## XI. AFFILIATION

Pour ensuite ajouter cet utilisateur dans un groupe créer :

```
Add-ADGroupMember -Identity "NomDuGroupe" -Members  
jdoe
```

Vérifier que l'utilisateur à bien été mit dans le groupe :

```
Get-ADGroupMember -Identity "NomDuGroupe"
```



```
DC-01 sur ALBERT3  
Administrateur: C:\Windows\system32\cmd.exe  
PS C:\Users\Administrateur> Add-ADGroupMember -Identity "Classe 1.1" -Members p.tieng  
PS C:\Users\Administrateur> Get-ADGroupMember -Identity "Classe 1.1"  
  
distinguishedName : CN=p.tieng,CN=Users,DC=Mydomain,DC=lab  
name               : p.tieng  
objectClass        : user  
objectGUID         : 8760c389-f2f3-4516-8763-e25f324151f1  
SamAccountName     : p.tieng  
SID                : S-1-5-21-268914275-751699927-1371988269-1108  
  
PS C:\Users\Administrateur> _
```