Active Directory : joindre un ordinateur à un domaine en ligne de commande

**Windows 2008R2** **Windows 2012** **Windows Server 2012R2** **Windows Server 2016** **Windows Server 2019**

Sommaire

1. [Introduction](https://rdr-it.com/active-directory-joindre-ordinateur-domaine-ligne-commande/#introduction)
2. [Prérequis](https://rdr-it.com/active-directory-joindre-ordinateur-domaine-ligne-commande/#prrequis)
3. [Jonction avec NETDOM](https://rdr-it.com/active-directory-joindre-ordinateur-domaine-ligne-commande/#jonction-avec-netdom)
4. [DJOIN : jonction hors-ligne](https://rdr-it.com/active-directory-joindre-ordinateur-domaine-ligne-commande/#djoin-jonction-hors-ligne)
5. [Jonction avec PowerShell](https://rdr-it.com/active-directory-joindre-ordinateur-domaine-ligne-commande/#jonction-avec-powershell)

Introduction

La jonction au domaine [Active Directory](https://rdr-it.com/mise-en-place-environnement-active-directory/) d’un ordinateur peut être effectué soit à l’aide l’interface graphique ou à l’aide de ligne de commande et PowerShell.

Dans ce tutoriel, je vais vous expliquer comment joindre un ordinateur à un domaine à l’aide des différentes lignes de commande qui sont :

* NETDOM
* DJOIN pour la jonction hors-ligne
* Add-Computer en PowerShell

Il est aussi possible de le faire à l’aide l’[interface graphique](https://rdr-it.com/joindre-un-ordinateur-a-un-domaine-windows-10-2016/).

Prérequis

Avant de joindre un ordinateur à un domaine Active Directory, il faut s’assurer que celui-ci puisse communiquer avec un contrôleur, ce qui implique une configuration IP correcte (serveur DNS pointant sur un contrôleur). Afin d’opérer la jonction, il faut également être administrateur local du poste et avoir un compte dans le domaine autorisé à joindre des ordinateurs.

Jonction avec NETDOM

Les commandes NETDOM sont un ensemble de commandes qui permettent la gestion de l’Active Directoy, elles sont présentes nativement sur les versions servers de Windows.

La commande doit être exécutée sur le poste à joindre au domaine.

Syntaxe :

netdom join /domain: /ud: /pd:<\*|password-account-ud>

**computer-name** : nom de l’ordinateur à  joindre au domaine  
/**domain** : nom du domaine Active Directory  
/**ud** : Compte du domaine autorisé à joindre un ordinateur  
/**pd** : \* permet la saisie du mot de passe ou entrer le mot de passe en clair.  
/*OU* : il est possible d’indiquer l’OU  
/*reboot* : redémarrage du poste après la jonction

Exemple : Jonction du domaine SRVCORE2 au domaine lab.intra

netdom join SRVCORE2 /domain:lab.intra /ud:LAB\administrateur /pd:\*

DJOIN : jonction hors-ligne

La commande djoin permet de joindre un ordinateur en hors-ligne, cette jonction se fait en deux étapes :

* Provision du poste dans l’annuaire Active Directory avec la génération d’un fichier de métadonnées.
* Jonction de l’ordinateur au domaine avec le fichier.

Voici quelques exemples de cas d’utilisation :

* Déploiement de poste d’ordinateur par un prestataire extérieur pour éviter de lui fournir un compte Active Directory.
* Limiter le nombre de personnes autorisé à joindre des ordinateurs au domaine.
* Déploiement de Nano server.

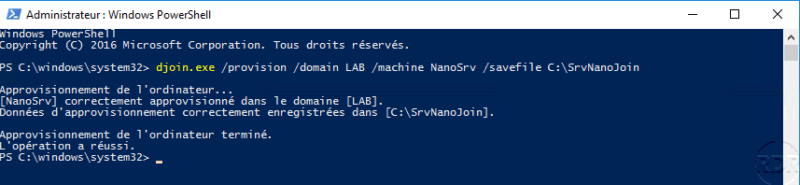
Provision du compte ordinateur, commande a exécuter sur un contrôleur de domaine où depuis un poste membre du domaine (minimum WIndows 7) avec un compte membre du groupe Admins du domaine :

djoin /provision /domain /machine /savefile

**domain** : nom du domaine Active Directory  
**machine** : nom du compte ordinateur  
**savefile** : emplacement et nom du fichier contenant les metadonnées.

Exemple :

djoin /provision /domain LAB /machine NanoSrv /savefile C:\SrvNanoJoin

[](https://rdr-it.com/wp-content/uploads/2019/06/djoin-1.png)

Vérifier que le compte Ordinateur et que le fichier soit bien présent.

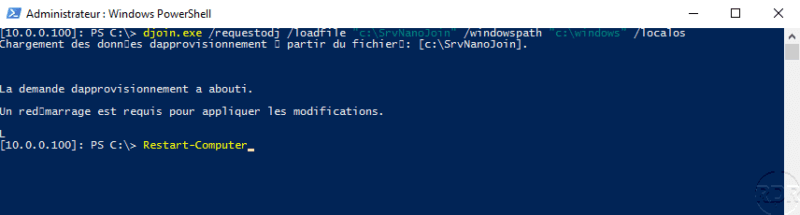
Sur le poste à joindre au domaine, copier le fichier et entrer la commande suivante pour effectuer la jonction du poste :

djoin /requestodj /loadfile <Folder\File> /windowspath <WindowsFolder> /localos

**requestodj** : jonction au domaine hors-ligne au prochain redémarrage  
**loadfile** : fichier contenant les métadonnées  
**windowspath** : dossier d’installation de Windows, il est d’utiliser les variables suivantes dans la commade %systemroot% ou %windir%  
**localos** : cible le système d’exploitation local

Exemple :

djoin /requestodj /localfile C:\SrvNanoJoin /windowspath C:\Windows /localos

[](https://rdr-it.com/wp-content/uploads/2019/06/djoin-3.png)

Au prochain redémarrage l’ordinateur sera joint au domaine.

Jonction avec PowerShell

Sur l’ordinateur à joindre au domaine ouvrir une invite de commande PowerShell et entrer  la commande suivante :

Add-Computer -DomainName -Credential

**DomainName** : domaine Active Directory  
**Credential** : Compte utilisateur avec les droits de joindre un ordinateur au domaine

À la validation de la commande, entrer le mot de passe du compte, une fois la commande validée redémarrer l’ordinateur pour la prise en compte.

Exemple :

Add-Computer -DomainName lab.intra -Credential administrateur@lab.intra

# Powershell #01| Installer un serveur AD/DNS en mode Core

*22 août 2019*[*0*](https://notamax.be/powershell-01-installer-un-serveur-ad-dns-en-mode-core/#respond)*Par*[MAIRIEN ANTHONY](https://notamax.be/author/it-anthony/)

Premier article d’une (je l’espère?) longue série parlant du fameux langage Powershell ! Ici nous allons voir une première introduction à propos de ce langage, puis voir comment installer un Windows Server en mode Core et comment installer un Active Directory & serveur DNS en ligne de commandes. On attaque fort dès le début !

### **1) Préface**

Il est vrai que je ne réalise pas énormément d’articles parlant du monde Windows, alors pourquoi vouloir me lancer dans l’aventure Powershell maintenant ? Et bien, tout simplement car au détour d’un article sur le net, j’ai découvert qu’il était possible d’installer Powershell sur une distribution GNU/Linux -ce que je me suis empressé de faire car j’aime toujours bidouiller- et effectivement, it works. Le tout fonctionne grâce aux fameux flatpacks et en réalité, quand on se penche un peu plus sur ce fameux langage qu’est le Powerhell, on se rend compte qu’il peut être assez intéressant… Après, je ne suis qu’un débutant avec ce dernier, et étant plus porté Linux que Windows, j’avoue lui préférer largement Bash (qui est plus simple/pratique à mon goût).

Il faut aussi dire qu’en entreprise, savoir manier la ligne de commande de chez Microsoft et savoir installer des serveurs en mode Core, ça peut être très utile et peut vous faire gagner des points sur votre CV car encore très peu de Sysadmins (dans mon entourage tout dû moins) n’osent pas s’aventurer dans ces méandres…

Bref, après cette petite préface, passons à l’introduction !

### **2) Introduction**

Le Powershell est donc à la fois un shell (invite de commandes), un langage de script, et aussi un kit de développement. Petit rappel rapide, le shell est ce qui permet à l’utilisateur d’interagir directement avec son OS, comme le fameux « cmd » ou encore le terminal de GNU/Linux.

Il est disponible chez Windows depuis la version 7 de son système d’exploitation, et est toujours présent dans la dernière build de Windows 10. Il est régulièrement mis à jour (version 6 actuellement) et vise à terme à remplacer le traditionnel invite de commandes avec son bon vieux MS-DOS.

Les commandes sont donc constituées d’un **verbe** ou **préfixe**et d’un **nom**séparés par un **tiret**. Elles peuvent être ensuite suivies de paramètres. On les appelle des **commandlets**pour command applets en anglais, mais on abrège souvent en **cmdlets**.

Bien, parlons maintenant du mode Core de Windows Server. Ce mode est ni plus ni moins qu’une version sans GUI (comprenez sans interface graphique) permettant donc essentiellement :

* De réduire la surface d’attaque, car il n’y a plus d’Internet Explorer ou autres logiciels du genre ;
* De réduire la consommation ressource, car Dieu sait qu’un Windows Server avec interface graphique est gourmand ;

\*Il est à noter aussi que depuis peu Microsoft a publié un nouveau type de serveur, appelé « Nano Server », nous n’en parlerons pas ici.

### **3) Installation du WS 2012 R2 en mode Core**

Bien, maintenant que l’on y voit un peu plus clair dans tout cela, on va pouvoir s’attaquer à la préparation de notre lab. Ici rien de sorcier, nous allons utiliser VMware Workstation et télécharger l’ISO d’un Windows Server 2012 R2 classique. Au niveau de la configuration, la voici :

Les administrateurs système qui savent se passer de l’interface graphique de Windows Server car ils maitrisent les commandes [PowerShell](https://www.windows8facile.fr/tag/powershell/), peuvent installer Windows Server 2016, 2019 et 2022 en mode minimal, c’est-à-dire l’édition Core / Nano Server du système d’exploitation serveur de Microsoft. Mais puisqu’il n’y a pas d’interface graphique et que l’OS n’embarque pas forcément les drivers de la carte réseau virtuelle d’un hyperviseur [VMware](https://www.windows8facile.fr/tag/vmware/) comme vSphere ou vCloud, difficile d’exécuter des commandes pour télécharger les pilotes adéquats sans connexion à internet.

Ce tutoriel explique comment **installer les drivers virtuels VMware Tools sur une VM Windows Server Core / Nano**, compatible avec [Windows Server 2022](https://www.windows8facile.fr/category/version/windows-server-2022/), [Windows Server 2019](https://www.windows8facile.fr/category/version/windows-server-2019/) et [Windows Server 2016](https://www.windows8facile.fr/category/version/windows-server-2016/). Une procédure un peu graphique mais indispensable pour avoir une connexion réseau sur la machine virtuelle serveur. **Ce guide est compatible avec les hyperviseurs VMware vSphere et vCloud** mais aussi avec les logiciels de virtualisation [**VMware Workstation Pro et Player**](https://www.windows8facile.fr/tag/vmware-workstation/).

**Installer les VMware Tools sur Windows Server Core / Nano**

1. **Créer une nouvelle machine virtuelle** Windows Server avec les paramètres désirés.

2. **Installer le système d’exploitation** Windows Server en version Server Core.

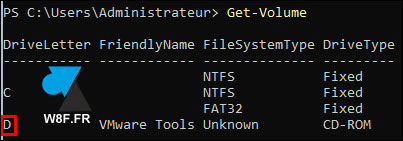
3. Une fois que le mot de passe Administrateur est défini, appuyer sur « **15** » pour ouvrir un PowerShell (selon version OS).

4. Sortir de la VM (**CTRL + ALT**) pour aller dans le menu **VM**, **Invité**, **Installer / Mettre à jour les VMware Tools** ou équivalent sur votre hyperviseur (en anglais : VM, Guest, Install VMware Tools), ou **File**, **Manage**, **Install VMware Tools** sur VMware Workstation. Le CD virtuel des drivers est chargé sur la VM.

5. Revenir dans la VM Windows Core

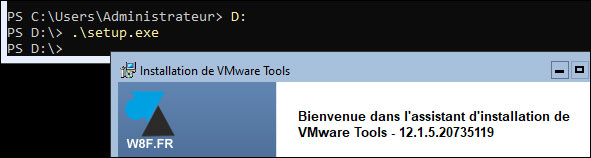
Sur Windows Server 2016 Nano (Core), taper « **powershell** » pour passer en mode PowerShell (PS). Sur 2019 et 2022, on y est directement.

6. Taper « **Get-Volume** » (ou Get-PSDrive) pour noter la lettre de lecteur du disque VMware Tools. On voit ici que c’est la lettre D qui a été attribuée.



7. Taper « **D:** » pour se placer sur le lecteur DVD.

8. Taper « **.\setup.exe** » pour ouvrir l’assistant d’installation des VMware Tools en mode graphique par dessus la console PowerShell.



Si rien ne se passe après quelques minutes, c’est sûrement parce que la fenêtre d’installation des VMware Tools est située derrière l’Invite de commandes. Déplacer, à la souris, le cmd pour voir le setup graphique VMware, ou avec ALT + TAB.

9. Avancer dans l’installation des VMTools par de simples **Suivant** (installation « **Typique**« ).

10. A la fin, valider le **redémarrage**du système pour appliquer les modifications.

11. Une fois redémarré et connecté à la session administrateur, un simple « **ipconfig** » montrera que la carte réseau est active.

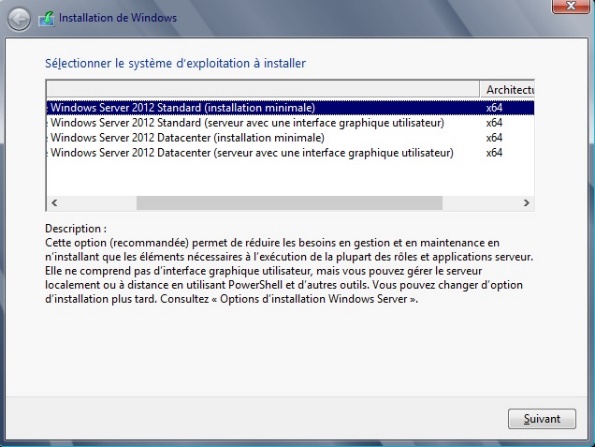
La suite se passe en PowerShell pour gérer Windows Server Core / Nano avec connexion réseau.

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Pardon ? Pas de 8Go de mémoire vive ou de double/quadruple  
processeurs ? Et bien non ! Car comme dit précédemment, cette version est bien plus légère.

Bien, une fois la machine virtuelle démarrée, nous allons simplement suivre l’installation de manière classique, en choisissant cependant la version sans interface graphique au moment opportun :



Ensuite l’installation (bien plus rapide forcément) se déroule de manière standard.

### **4) Commandes Powershell de base et configuration de l’hôte**

Et c’est ici que nous mettons les mains dans le cambouis. Bien, ici je ne vais pas vous détailler entièrement la syntaxe de Powershell, car comme dit en début d’article, elle se compose de verbe ou préfixe suivi d’un nom après un tiret, donc je vais me contenter de vous décrire les commandes que nous utiliserons ici. Plus vous pratiquerez ces commandes, plus celles-ci rentreront vite.

Première étape, installer les Vmware tools si nous installons notre serveur en tant que VM :

cd D:

setup.exe

Ici, malgré le fait que nous n’ayons pas d’interface graphique, l’installation des additions invités se fait de manière classique ; on clique sur **VM** puis **Install vmware tools** et ensuite on se rend sur le lecteur D: (par défaut) puis on démarre le setup, comme si de rien n’était !

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Nous pouvons ensuite redémarre notre hôte et hop, tout est prêt !

La véritable première étape est donc le changement du nom d’hôte :

Rename-Computer -NewName "srv-core"

shutdown /r

Ici, nous spécifions simplement l’argument -NewName à la commande Rename-Computer puis le tour est joué !

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Ensuite, un simple coup de hostname nous indiquera le nouveau nom d’hôte.

La prochaine étape (classique) est donc d’attribuer une adresse IP statique à notre serveur, toujours en Powershell. Pour cela, nous allons déjà utiliser le verbe **Get** pour obtenir des infos sur le module **NetAdapter**de façon à lister les interfaces réseaux disponibles :

Get-NetAdapter

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Ici nous pouvons donc voir le nom de notre interface, sa description, son numéro d’index, puis son état (up/down). Son numéro d’index est très important, car c’est ce qui va permettre de désigner réellement l’interface en question. Bien, voici donc la fameuse commande pour attribuer une adresse IP statique :

Set-NetIPAddress –InterfaceIndex 12 –IPAddress 192.168.1.100 –PrefixLength 24 –DefaultGateway 192.168.1.1

Bien, nous utilisons donc le verbe **Set**sur le module **NetIPAddress** en donnant ensuite les arguments suivants :

* **InterfaceIndex**, étant l’index de la carte en question ;
* **IPAddress**, étant l’adresse IP souhaitée ;
* **PrefixLength**, étant le masque de sous-réseau ;
* **DefaultGateway**, étant l’adresse IP de la gateway ;

Ensuite, un petit **Get-NetIPConfiguration** nous démontre bien que la nouvelle adresse est prise en compte :

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

A noter que si notre carte était en DHCP, celui-ci est automatiquement désactivé (logique). Pour le ré-activer, nous pouvons effectuer la commande suivante :

Set-NetIPInterface -InterfaceIndex 12 -Dhcp {Enabled/Disabled}

Et je rajouterai aussi qu’il est possible d’assigner plusieurs adresses IP à une même interface, pour cela nous remplaçons simple le verbe **Set** par **New** et la démarche reste identique !

Puis nous pouvons enfin renseigner notre serveur DNS (qui est donc le localhost) :

Set-DnsClientServerAddress -InterfaceIndex 12 -ServerAddresses 127.0.0.1

Bien, maintenant que notre hôte a accès à Internet, nous allons pouvoir l’activer ! Pour cela, c’est très simple, nous allons nous servir d’un utilitaire présent même sur les versions Desktop.

Pour désinstaller une clé de licence (d’essai, par exemple) :

slmgr –upk

L’agument -upk signifie uninstall product key, et donc pour activer une clé, nous utiliserons l’agument -ipk pour install product key :

slmgr -ipk XXXXX-XXXXX-XXXXX-XXXXX-XXXXX

Et enfin, pour tenter l’activation en ligne :

slmgr.vbs /ato

### **5) Installation du rôle Active Directory et du DNS**

Une fois notre Windows Serveur activé, nous allons pouvoir passer à l’installation du rôle AD et DNS !

Pour cela, rien de plus simple, on va utiliser le verbe **Install** sur le module **WindowsFeature** en lui renseignant ensuite le nom du service à installer, ici ce sera donc **AD-Domain-Services**. L’option **-IncludeManagementTools** est optionnelle, mais celle-ci pourra nous servir éventuellement par la suite :

Install-WindowsFeature AD-Domain-Services -IncludeManagementTools

\*Il est à noter que le rôle DNS ne s’installera que lors de la création de la forêt de base.

On va donc créer notre domaine de cette manière, en utilisant le verbe **Install** sur le module **ADDSForest**et en renseignant l’agument **-DomaineName**, tout simplement !

Install-ADDSForest -DomainName notamax.local

Puis la configuration se lance après avoir renseigné le mot de passe du SafeMode, et pour terminer notre serveur redémarrera :

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

A la connexion, nous pouvons d’ailleurs voir que nous serons connecté en domaine (NOTAMAX\) :

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Désormais, nous allons pouvoir créer des groupes ainsi que des utilisateurs, le tout en Powershell !

Pour commencer, on créer une OU ( Organizational Unit), permettant pour rappel de classer différents objets tel que des utilisateurs, des imprimantes, des ordinateurs…

New-ADOrganizationalUnit -Name Siège\_principal -Path "DC=notamax,DC=local"

Donc ici, comme d’habitude, nous allons utiliser le verbe **New** sur le module **ADOrganizationalUnit** pour en créer un nouveau, en lui spécifiant ensuite les arguments :

* **-Name** : Le nom de l’OU en question ;
* **-Path** : Le chemin vers le domaine (Domain Component) ;

Vient ensuite la commande pour créer un groupe classique, qui est la suivante :

New-ADGroup -Name "Commercial" -GroupCategory Security -GroupScope Global -DisplayName "Commercial" -Path "OU=Siège\_principal,DC=Notamax,DC=local" -Description "Groupe contenant les commerciaux"

Nous allons détailler tout ceci ensemble :

* On utilise le verbe **New** sur le module **ADGroup**, pour créer un nouveau groupe ;
* L’argument **-Name** spécifie le nom de ce groupe ;
* L’argument **-GroupCategory** spécifie le type de groupe, à savoir **Security** ou **Distribution** ;
* L’argument **-GroupScope** spécifie la portée du groupe, **Local**, **Global**, ou **Universal** ;
* Le **Displayname** est le nom que ce groupe affichera ;
* Le **Path** est identique à ce que l’on a vu au dessus, c’est « le chemin vers le domaine » sauf qu’ici nous allons l’incorporer dans notre OU créée précédemment (Siège\_principal) ;
* L’argument **-Description** est optionnel, il indique simplement une description du groupe, comme son nom l’indique ;

\*Si certaines choses vous échappent, notamment concernant la portée ou le type de groupe, je vous renvois à mon travail de fin d’année disponible [*ici*](https://notamax.be/scripts/TFE.pdf)où je décris chacun de ces points

Bien, à ce niveau là, nous avons déjà réalisé une OU, un groupe, et nous allons continuer en créant un utilisateur ! (Rassurez-vous, nous arrivons doucement à la fin).

Déjà, nous allons exécuter la fameuse commande mais en rajoutant l’argument -Syntax, permettant d’afficher les différentes options qui s’offrent à nous :

Get-Command New-ADUser –Syntax

Une image contenant texte, capture d’écran, Police, information

Description générée automatiquement

Comme on peut le voir ici, il y en a un bon paquet… que je ne peux malheureusement tous vous décrire ici ! Nous allons voir comment créer un User classique, ce qui sera déjà un bon début.

New-ADUser -Name "Thomas Andersen" -GivenName "Thomas" -Surname "Andersen" -SamAccountName "thomas-andersen" -UserPrincipalName "thomas-andersen@notamax.local" -Path "OU=Siège\_Principal,DC=notamax,DC=local" -AccountPassword(Read-Host -AsSecureString "Rentrez le mot de passe svp") -Enabled $true

Alors alors, comme d’habitude, décortiquons un peu tout ça…

* **-Name**, le nom complet de l’utilisateur ;
* **-GivenName**, le prénom de l’utilisateur ;
* **-Surname**, son nom de famille ;
* **-SamAccountName**, le nom d’utilisateur qui sera inscrit dans le gestionnaire de comptes de sécurité (SAM). C’est une sorte de base de données du système d’exploitation Windows qui contient les noms d’utilisateur et les mots de passe. Pour faire simple, ce sera le nom utilisé pour ouvrir la session sous forme **NOTAMAX\thomas-andersen** ;
* **-UserPrincipalName**, nom d’utilisateur pour ouvrir la session sous forme **thomas-andersen@notamax.local** ;
* **-Path**, le chemin vers le domaine, classique ;
* **-AccountPassword(Read-Host – AsSecureString**), ici nous allons donc setup un mot de passe pour le compte de l’utilisateur, en renseignant à Powershell l’argument **Read-Host** (permettant de demander à l’utilisateur de rentrer un mot de passe) mais aussi l’argument **AsSecureString**, permettant tout simplement de cacher les lettres tapées au clavier ;

Une fois fait, nous pouvons réaliser la commaned **Get-ADUser**pour obtenir des infos sur l’utilisateur nouvellement créé :

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

Et voilà, nous pouvons voir ici que le compte est bien activé et que les différentes informations sont correctes !

\*Vous pouvez d’ailleurs essayer la commande ***Get-ADGroup Commercial*** qui vous renverra elle aussi des informations, mais concernant le groupe que nous avons créé précédemment.

Pour passer notre utilisateur dans ce fameux groupe, voici la commande à effectuer :

Add-AdGroupMember ‘Commercial’ thomas-andersen

Ici nous spécifions donc d’abord le groupe, et ensuite l’utilisateur.

Et pour en lister les membres :

Get-AdGroupMember -identity Commercial

### **6) Connexion au domaine et conclusion**

On touche au but ! Ici il nous suffit de lancer une petite VM Windows 10, de renseigner l’adresse IP de notre serveur AD comme serveur DNS primaire, puis nous renseignons ensuite le compte Administrateur du domaine et nous entrons, procédure classique donc.

Ici nous allons donc nous connecter avec notre utilisateur **thomas-andersen** membre du groupe **Commercial**, et tadaaaa !

Une image contenant texte, capture d’écran, ordinateur, logiciel

Description générée automatiquement

Nous sommes connecté à notre domaine ! Pas mal non ?

Bien entendu, tout cela reste néanmoins très basique… pas encore de GPO, pas d’expiration de mot de passe, pas de Reverse DNS… mais pour une première approche de Powershell, je trouve que l’on s’est bien débrouillé

Je vais donc m’arrêter ici pour ce premier article, qui est déjà assez conséquent. Dans le second, nous verrons l’utilisation du menu interactif de configuration de serveur en Powershell nommé Sconfig, puis nous verrons comment appliquer quelques GPOs de base à certains groupes, et éventuellement comment créer et utiliser un script relié à un fichier Excel pour créer de multiples utilisateurs d’un coup.