**Ajouter adresses ip sur une carte réseau d’un serveur :**

Le cmdlet **New-NetIPAddress** permet d'ajouter des adresses IP sur une carte réseau

Le cmdlet **Get-NetIPAddress** permet d'afficher des informations sur les adresses IP d'une carte réseau.

**$nic = "Ethernet0"**

**for ($i = 101; $i -le 110; $i++)**

**{**

**New-NetIPAddress -InterfaceAlias $nic `**

**-IPAddress 192.168.1.$i `**

**-PrefixLength 24**

**}**

**Get-NetIPAddress -InterfaceAlias $nic `**

**-AddressFamily IPv4 | Select-Object IPv4Address,PrefixLength**

Voici le résultat

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

Description générée automatiquement

**Get-LocalGroup -Name iis\***

Une image contenant texte, capture d’écran, Police, Bleu électrique

Description générée automatiquement

Le groupe **IIS\_IUSRS** a des autorisations NTFS de "Lecture et exécution" sur le dossier "**c:\inetpub\wwwroot**".

**Vider la cache de votre navigateur**

**Rafraîchissement des pages web affichées**

Afin de vous assurer que l'affichage est bien le résultat des modifications demandées à chaque exercice, vous devez effacer les fichiers qui sont dans la cache de votre navigateur.

En utilisant PowerShell, ajouter un nouveau site Web "par adresse" ayant les caractéristiques suivantes:

* Nom du site: **Web adresse 103**
* Chemin d’accès physique: **e:\\_web\adresse\_103**
* Adresse IP: **192.168.1.103**
* Port: **80**

Écrire le code PowerShell pour créer ce site Web.

**# Vous devez supposer que le dossier existe**

**$site = "Web adresse 103"**

**$dossier = "e:\\_web\adresse\_103"**

**$ip = "192.168.1.103"**

**$port = 80**

**New-Website -Name $site `**

**-PhysicalPath $dossier `**

**-IPAddress $ip `**

**-Port $port `**

En utilisant PowerShell, ajouter un nouveau site Web "par port" ayant les caractéristiques suivantes:

* Nom du site: **Web port 1280**
* Chemin d’accès physique: **e:\\_web\port\_1280**
* Adresse IP: **192.168.1.10**
* Port: **1280**

Écrire le code PowerShell pour créer ce site Web.

**# Vous devez supposer que le dossier existe**

**$site = "Web port 1280"**

**$dossier = "e:\\_web\port\_1280"**

**$ip = "192.168.1.10"**

**$port = 1280**

**New-Website -Name $site `**

**-PhysicalPath $dossier `**

**-IPAddress $ip `**

**-Port $port**

**Voici la commande qui permet de créer une règle de trafic entrant qui ouvre le port 1280.**

**New-NetFirewallRule -DisplayName "\_PORT\_1280" `**

**-Profile "Private,Public" `**

**-Direction Inbound `**

**-Action Allow `**

**-Protocol TCP `**

**-LocalPort 1280**

**Voici la commande pour vérifier la présence de la règle "\_PORT\_1280"**

**Get-NetFirewallRule -DisplayName "\_PORT\_1280"**

**Voci la commande pour afficher des informations sur les sites web**

**Get-Website**

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Voici la commande pour afficher des informations sur le site "Web adresse 103"**

**Get-Website –Name "Web adresse 103"**

Une image contenant texte, capture d’écran, Police, Bleu électrique

Description générée automatiquement

**Voici la commande pour arrêter un site web**

**Stop-Website –Name "Web adresse 103"**

**Voici la commande pour démarrer un site web**

**Start-Website –Name "Web adresse 103"**

**DNS**

La commande NSLOOKUP.EXE est une commande qui permet d'interroger un serveur DNS pour obtenir des informations sur un nom de domaine.

Par défaut, la commande NSLOOKUP.EXE utilise le serveur DNS préféré qui est configuré sur la carte réseau.

**Source de nom (SOA)**

**SOA signifie "Start of Authority"**

**L'enregistrement "SOA" indique quel serveur DNS est la meilleure source d'information pour le domaine spécifié. Chaque domaine doit avoir un enregistrement "SOA".**

**Serveur de noms (NS)**

**NS signifie "Name Server"**

**NOTE: Le nom du serveur de nom se termine par un POINT.**

Les serveurs DNS gardent en cachent des informations sur les sites consultés.

Voici comment vider la cache au niveau de la fenêtre "Invite de commandes".

**ipconfig.exe /flushdns**

**/flushdns Purge le cache de résolution DNS.**

Voici comment afficher le contenu de la cache au niveau de fenêtre "Invite de commandes".

**Ipconfig.exe /displaydns**

**/displaydns Affiche le contenu du cache de résolution DNS.**

Voici comment vider la cache au niveau du serveur DNS avec une commande PowerShell.

**Clear-DnsServerCache -Force**

**Utilisation de plusieurs cmdlets du module DNSserver**

Le rôle "Serveur DNS" est installé sur le SERVEUR2 donc le module DnsServer est présent.

La console de gestion DNS est installée sur le SERVEUR1 donc le module DnsServer est présent.

**NOTE: Les commandes PowerShell à exécuter sur le serveur virtuel "SERVEUR2"**

Voici la commande pour créer une zone directe principale

**Add-DnsServerPrimaryZone -Name domaine2.local `**

**-ZoneFile domaine2.local.dns `**

**-DynamicUpdate None**

Affiche des informations sur toutes les zones du serveur DNS

**Get-DnsServerZone**

Affiche des informations sur la zone "domaine2.local"

**Get-DnsServerZone -Name domaine2.local**

Création d’un enregistrement de type "A"

**Add-DnsServerResourceRecordA -Name www `**

**-IPv4Address 192.168.1.10 `**

**-ZoneName domaine2.local**

Création d’un enregistrement de type CNAME

**Add-DnsServerResourceRecordCName -Name support `**

**-HostNameAlias www.domaine2.local `**

**-ZoneName domaine2.local**

Affiche tous les enregistrements d'un domaine

**Get-DnsServerResourceRecord -ZoneName domaine2.local**

Affiche les enregistrements d'un domaine selon le type

**Get-DnsServerResourceRecord -ZoneName domaine2.local `**

**-RRType SOA**

**Get-DnsServerResourceRecord -ZoneName domaine2.local `**

**-RRType NS**

**Get-DnsServerResourceRecord -ZoneName domaine2.local `**

**-RRType A**

**Get-DnsServerResourceRecord -ZoneName domaine2.local `**

**-RRType CNAME**

Commande pour afficher le nom et les adresses IP des serveurs racines.

**Get-DnsServerRootHint**

Commande pour vider le contenu de la cache du serveur DNS

**Clear-DnsServerCache -Force**

**NOTE: Les commandes PowerShell à exécuter sur le serveur virtuel "SERVEUR1"**

Voici la commande pour créer une zone directe principale sur le SERVEUR2 à partir du SERVEUR1.

**Add-DnsServerPrimaryZone -Name domaine3.local `**

**-ZoneFile domaine3.local.dns `**

**-DynamicUpdate None `**

**-ComputerName SERVEUR2**

Création d’un enregistrement de type "A" sur le SERVEUR2 à partir du SERVEUR1.

**Add-DnsServerResourceRecordA -Name www `**

**-IPv4Address 192.168.1.10 `**

**-ZoneName domaine3.local `**

**-ComputerName SERVEUR2**

Création d’un enregistrement de type CNAME sur le SERVEUR2 à partir du SERVEUR1.

**Add-DnsServerResourceRecordCName -Name support `**

**-HostNameAlias www.domaine3.local `**

**-ZoneName domaine3.local `**

**-ComputerName SERVEUR2**

**NOTE: Exemple pour détruire plusieurs zones DNS**

**ATTENTION: vous devez exécuter ce code en cas de nécessité seulement**

**$zones = (Get-DnsServerZone).ZoneName**

**foreach ($zone in $zones)**

**{**

**# Efface toutes les zones dont le nom se termine par ".local"**

**if ($zone.EndsWith(".local","CurrentCultureIgnoreCase"))**

**{**

**Write-Host "EFFACE LA ZONE DNS: $zone"**

**Remove-DnsServerZone -Name $zone -Force**

**}**

**}**

**Utilisation de plusieurs cmdlets du module DNSserver**

Le rôle "Serveur DNS" est installé sur le SERVEUR2 donc le module DnsServer est présent.

La console de gestion DNS est installée sur le SERVEUR1 donc le module DnsServer est présent.

**Création d’un domaine enfant par programmation PowerShell.**

**NOTE: Les commandes PowerShell à exécuter sur le serveur virtuel "SERVEUR2"**

Création de la zone directe principale

**Add-DnsServerPrimaryZone -Name parent2.local `**

**-ZoneFile parent2.local.dns `**

**-DynamicUpdate None**

**Par programmation, on ne peut pas créer un domaine enfant sans enregistrement de type "A".**

Création du domaine "enfant" contenant deux enregistrements de type "A".

**Add-DnsServerResourceRecordA -Name site1.enfant `**

**-IPv4Address 192.168.1.107 `**

**-ZoneName parent2.local**

**Add-DnsServerResourceRecordA -Name site2.enfant `**

**-IPv4Address 192.168.1.108 `**

**-ZoneName parent2.local**

La délégation de domaines permet de séparer les responsabilités entre les équipes.

# Cette commande affiche des informations sur les enregistrements de type "A"

# de la zone **delegation1.parent.local** qui sont sur le **SERVEUR1**.

**Get-DnsServerResourceRecord -ZoneName delegation1.parent.local `**

**-RRType A `**

**-ComputerName SERVEUR1**

# Cette commande affiche des informations sur la zone déléguée qui est sous PARENT.LOCAL

**Get-DnsServerZoneDelegation -Name parent.local `**

**-ComputerName SERVEUR2**

# Cette commande affiche des informations sur les enregistrements de type "A"

# de la zone **delegation1.parent.local** qui sont sur le **SERVEUR1**.

**Get-DnsServerResourceRecord -ZoneName delegation1.parent.local `**

**-RRType A `**

**-ComputerName SERVEUR1**

**Utilisation de plusieurs cmdlets du module DNSserver**

Le rôle "Serveur DNS" est installé sur le SERVEUR2 donc le module DnsServer est présent.

Le rôle "Serveur DNS" est installé sur le SERVEUR1 donc le module DnsServer est présent.

**Création d’une zone déléguée par programmation PowerShell.**

**NOTE: Les commandes PowerShell à exécuter sur le "SERVEUR2" si la zone principale "parent.local" existe**

**ÉTAPE 1 - Création d’un domaine délégué**

La zone parent se nomme **parent.local**

Le domaine délégué se nomme **delegation1**

Le domaine délégué sera sur le **SERVEUR1** à l’adresse IP **192.168.1.10**

**Add-DnsServerZoneDelegation -ChildZoneName delegation1 `**

**-Name parent.local `**

**-IPAddress 192.168.1.10 `**

**-NameServer serveur1.parent.local `**

**-ComputerName SERVEUR2**

**NOTE: Les commandes PowerShell à exécuter sur le "SERVEUR1"**

**ÉTAPE 2 - Création d’une zone principale directe sur le "SERVEUR1"**

**note: le nom de la zone doit avoir le même nom que sur le "SERVEUR2"**

**Add-DnsServerPrimaryZone -Name delegation1.parent.local `**

**-ZoneFile delegation1.parent.local.dns `**

**-DynamicUpdate None `**

**-ComputerName SERVEUR1**

Création de deux enregistrements de type "A"

**Add-DnsServerResourceRecordA -ZoneName delegation1.parent.local `**

**-Name r1 `**

**-IPv4Address 192.168.1.201 `**

**-ComputerName SERVEUR1**

**Add-DnsServerResourceRecordA -ZoneName delegation1.parent.local `**

**-Name r2 `**

**-IPv4Address 192.168.1.202 `**

**-ComputerName SERVEUR1**

**Utilisation de plusieurs cmdlets du module DNSserver**

Le rôle "Serveur DNS" est installé sur le SERVEUR2 donc le module DnsServer est présent.

Le rôle "Serveur DNS" est installé sur le SERVEUR1 donc le module DnsServer est présent.

**NOTE: Les commandes à exécuter sur le serveur "SERVEUR2"**

**ÉTAPE 1 - CRÉATION DE LA ZONE PRINCIPALE**

**Add-DnsServerPrimaryZone -Name principale.local `**

**-ZoneFile principale.local.dns `**

**-DynamicUpdate None**

Affiche les propriétés des enregistrements de type "NS" pour la zone principale.local

**Get-DnsServerResourceRecord -ZoneName principale.local `**

**-RRType NS**

**HostName RecordType Type Timestamp TimeToLive RecordData**

**-------- ---------- ---- --------- ---------- ----------**

**@ NS 2 0 01:00:00 serveur2.**

**Par défaut, l'enregistrement "NS" n'utilise pas un nom pleinement qualifié (FQDN).**

**La propriété RecordData est un objet DnsServerResourceRecordNS.**

**ÉTAPE 2 - CRÉATION DES ENREGISTREMENTS DE TYPE "A" DANS LA ZONE PRINCIPALE**

Création de l’enregistrement "A" pour le nom du serveur qui héberge la zone principale

**Add-DnsServerResourceRecordA -Name serveur2 `**

**-IPv4Address 192.168.1.20 `**

**-ZoneName principale.local**

Création de l’enregistrement de type "A" pour le nom du serveur qui hébergera la zone secondaire

**Add-DnsServerResourceRecordA -Name serveur1 `**

**-IPv4Address 192.168.1.10 `**

**-ZoneName principale.local**

Création d'un enregistrement "A" supplémentaire

**Add-DnsServerResourceRecordA -Name www `**

**-IPv4Address 192.168.1.250 `**

**-ZoneName principale.local**

**ÉTAPE 3 - SUPPRIME TOUS LES ENREGISTREMENT "NS" DANS LA ZONE PRINCIPALE**

**note 1: à l'étape suivante, ils seront remplacés par des noms FQDN**

**note 2: c'est plus facile de supprimer les enregistrements "NS" existants et de les remplacer**

**$infos = Get-DnsServerResourceRecord -ZoneName principale.local -RRType NS**

**foreach ($info in $infos)**

**{**

**$name = $info.RecordData.NameServer**

**Write-Host "Suppression de l'enregistrement NS: $name"**

**Remove-DnsServerResourceRecord -ZoneName principale.local `**

**-InputObject $info `**

**-Force**

**}**

**ÉTAPE 4 - CRÉATION DES ENREGISTREMENTS "NS" DANS LA ZONE PRINCIPALE**

**note: on utilise un nom pleinement qualifié (FQDN) pour chaque enregistrement "NS"**

**Add-DnsServerResourceRecord -ZoneName principale.local `**

**-Name "@" `**

**-NS `**

**-NameServer serveur2.principale.local.**

**Add-DnsServerResourceRecord -ZoneName principale.local `**

**-Name "@" `**

**-NS `**

**-NameServer serveur1.principale.local.**

Affiche les propriétés des enregistrements de type "NS" pour la zone principale.local

**Get-DnsServerResourceRecord -ZoneName principale.local `**

**-RRType NS**

**HostName RecordType Type Timestamp TimeToLive RecordData**

**-------- ---------- ---- --------- ---------- ----------**

**@ NS 2 0 01:00:00 serveur2.principale.local.**

**@ NS 2 0 01:00:00 serveur1.principale.local.**

Chaque enregistrement "NS" a un nom pleinement qualifié (FQDN).

**NOTE: Les commandes à exécuter sur le serveur "SERVEUR1"**

**ÉTAPE 5 - CRÉATION DE LA ZONE SECONDAIRE SUR LE SERVEUR1**

Création de la zone directe secondaire

**Add-DnsServerSecondaryZone -Name principale.local `**

**-ZoneFile principale.local.dns `**

**–MasterServers 192.168.1.20**

**Étape 1c - Informations sur le pare-feu Windows**

**Par défaut, le port 3389 est ouvert pour les protocoles TCP et UDP.**



**Étape 1d - PowerShell et le pare-feu Windows**

**$nomTCP = "Bureau à distance — Mode utilisateur (TCP-Entrant)"**

**Get-NetFirewallRule -DisplayName $nomTCP | `**

**Get-NetFirewallPortFilter | Select-Object Protocol,LocalPort**

**Protocol LocalPort**

**-------- ---------**

**TCP 3389**

**$nomUDP = "Bureau à distance — Mode utilisateur (UDP-Entrant)"**

**Get-NetFirewallRule -DisplayName $nomUDP | `**

**Get-NetFirewallPortFilter | Select-Object Protocol,LocalPort**

**Protocol LocalPort**

**-------- ---------**

**UDP 3389**

**Si vous modifiez le numéro du port pour le protocole RDP dans le registre, vous devez ajouter des règles de trafic entrant pour autoriser ce port en TCP et UDP.**

**Voici la commande qui permet de créer une règle de trafic entrant qui ouvre le port 3390 en TCP.**

**New-NetFirewallRule -DisplayName "\_RDP\_TCP\_3390" `**

**-Profile "Private,Public" `**

**-Direction Inbound `**

**-Action Allow `**

**-Protocol TCP `**

**-LocalPort 3390**

**Voici la commande qui permet de créer une règle de trafic entrant qui ouvre le port 3390 en UDP.**

**New-NetFirewallRule -DisplayName "\_RDP\_UDP\_3390" `**

**-Profile "Private,Public" `**

**-Direction Inbound `**

**-Action Allow `**

**-Protocol UDP `**

**-LocalPort 3390**

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

Description générée automatiquement

**$nomTCP = "\_RDP\_TCP\_3390"**

**Get-NetFirewallRule -DisplayName $nomTCP | `**

**Get-NetFirewallPortFilter | Select-Object Protocol,LocalPort**

**Protocol LocalPort**

**-------- ---------**

**TCP 3390**

**$nomUDP = "\_RDP\_UDP\_3390"**

**Get-NetFirewallRule -DisplayName $nomUDP | `**

**Get-NetFirewallPortFilter | Select-Object Protocol,LocalPort**

**Protocol LocalPort**

**-------- ---------**

**UDP 3390**

**Le cmdlet Get-Disk de PowerShell permet de vérifier que le disque 2 n'est pas un disque local**

**Get-Disk | Select-Object DiskNumber,PartitionStyle,BusType**

**DiskNumber PartitionStyle BusType**

**---------- -------------- -------**

**0 GPT SAS**

**1 GPT SAS**

**2 GPT iSCSI**

**ANNEXE**

Voici comment trouver le nom IQN de l'initiateur iSCSI du SERVEUR1.

**IQN signifie (iSCSI Qualified Name)**

**En utilisant la console "Initiateur iSCSI"**

L'onglet "**Configuration**" de la console "**Initiateur iSCSI**" contient le nom de l'initiateur iSCSI.

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

Description générée automatiquement

**En utilisant une commande PowerShell**

**Get-InitiatorPort | Select-Object -Property NodeAddress,ConnectionType**

**NodeAddress ConnectionType**

**----------- --------------**

**iqn.1991-05.com.microsoft:serveur1 iSCSI**