**Commandes d’administration et réseau Bash**

**Assia EL GUERCH ID101**

1. **Administration :**

**Gestion de fichiers :**

1. **Navigation dans les fichiers et dossiers :**

* Get-ChildItem : Affiche les fichiers et les dossiers dans un répertoire spécifique.

*Exemple : Get-ChildItem C:\Dossier*

* Set-Location / cd : Change le répertoire courant.

*Exemple : Set-Location C:\Dossier*

* Push-Location : Empile le répertoire courant sur la pile de répertoires.

*Exemple : Push-Location C:\Dossier*

* Pop-Location : Supprime le répertoire courant de la pile de répertoires et déplace le répertoire courant vers le répertoire au sommet de la pile.

*Exemple : Pop-Location*

* Split-Path : Divise un chemin d'accès en plusieurs parties.

*Exemple : Split-Path C:\Dossier\Fichier.txt -Parent*

1. **Gestion de fichiers :**

* Copy-Item : Copie des fichiers d'un emplacement à un autre.

*Exemple : Copy-Item C:\Dossier\Fichier.txt D:\NouveauDossier*

* Move-Item : Déplace des fichiers d'un emplacement à un autre.

*Exemple : Move-Item C:\Dossier\Fichier.txt D:\NouveauDossier*

* Remove-Item : Supprime des fichiers ou des dossiers.

*Exemple : Remove-Item C:\Dossier\Fichier.txt*

* New-Item : Crée un nouveau fichier ou dossier.

*Exemple : New-Item C:\Dossier\NouveauDossier -ItemType Directory*

* Set-Content : Modifie le contenu d'un fichier.

*Exemple : Set-Content C:\Dossier\Fichier.txt -Value "Nouveau contenu"*

* Get-Content : Lit le contenu d'un fichier.

*Exemple : Get-Content C:\Dossier\Fichier.txt*

* Add-Content : Ajoute du contenu à la fin d'un fichier.

*Exemple : Add-Content C:\Dossier\Fichier.txt -Value "Contenu supplémentaire"*

* Rename-Item : Renomme un fichier ou un dossier.

*Exemple : Rename-Item C:\Dossier\Fichier.txt NouveauNom.txt*

* Get-Item : Obtient les informations sur un fichier ou un dossier.

*Exemple : Get-Item C:\Dossier\Fichier.txt*

* Test-Path : Vérifie si un chemin d'accès existe.

*Exemple : Test-Path C:\Dossier\Fichier.txt*

* Join-Path : Combine plusieurs parties de chemin en une seule.

*Exemple : Join-Path C:\Dossier "Fichier.txt"*

1. **Filtrage et tri des fichiers :**

* Where-Object : Filtre les objets dans une collection en fonction de conditions spécifiques.

*Exemple : Get-ChildItem | Where-Object { $\_.Extension -eq ".txt" }*

* Sort-Object : Trie les objets dans une collection par une propriété spécifique.

*Exemple : Get-ChildItem | Sort-Object LastWriteTime*

1. **Manipulation de fichiers compressés :**

* Compress-Archive : Crée une archive à partir de fichiers et de dossiers.

*Exemple : Compress-Archive C:\Dossier -DestinationPath C:\Archive.zip*

* Expand-Archive : Extrait le contenu d'une archive compressée.

*Exemple : Expand-Archive C:\Archive.zip -DestinationPath C:\Dossier*

1. **Manipulation de fichiers CSV :**

* Import-Csv : Importe les données d'un fichier CSV dans un objet PowerShell.

*Exemple : $data = Import-Csv C:\Chemin\vers\fichier.csv*

* Export-Csv : Exporte des données PowerShell vers un fichier CSV.

*Exemple : $data | Export-Csv C:\Chemin\vers\nouveau\_fichier.csv -NoTypeInformation*

* ConvertTo-Csv : Convertit des objets PowerShell en format CSV.

*Exemple : Get-Process | Select-Object Name, Id | ConvertTo-Csv | Out-File C:\Chemin\vers\processes.csv*

* ConvertFrom-Csv : Convertit des données CSV en objets PowerShell.

*Exemple : $csvData = Get-Content C:\Chemin\vers\fichier.csv | ConvertFrom-Csv*

* Select-Object : Sélectionne des propriétés spécifiques d'objets CSV.

*Exemple : $data | Select-Object Name, Age, City*

* Where-Object : Filtre les lignes d'un fichier CSV en fonction de conditions spécifiques.

*Exemple : $data | Where-Object { $\_.Age -gt 18 }*

* Sort-Object : Trie les lignes d'un fichier CSV par une colonne spécifique.

*Exemple : $data | Sort-Object Name*

* Group-Object : Regroupe les données en fonction des valeurs d'une colonne spécifique.

*Exemple : $data | Group-Object City*

* Measure-Object : Calcule des statistiques sur les données d'un fichier CSV.

*Exemple : $data | Measure-Object -Property Age -Average*

* Format-Table : Affiche les données CSV dans un format de tableau.

*Exemple : $data | Format-Table -AutoSize*

**Gestion de Processus et Services :**

1. **Gestion de Processus :**

* Get-Process : Affiche les processus en cours d'exécution sur l'ordinateur.

*Exemple : Get-Process*

* Stop-Process : Arrête un processus en cours d'exécution.

*Exemple : Stop-Process -Name "nomDuProcessus"*

* Start-Process : Démarre un nouveau processus.

*Exemple : Start-Process -Name "nomDuProcessus"*

* Wait-Process : Attends que l'exécution d'un processus se termine.

*Exemple : Wait-Process -Name "nomDuProcessus"*

* Get-WmiObject Win32\_Process : Obtient des informations détaillées sur les processus à l'aide de WMI.

*Exemple : Get-WmiObject Win32\_Process | Select-Object Name, ProcessId, CommandLine*

1. **Gestion de services:**

* Get-Service : Affiche les services en cours d'exécution sur l'ordinateur.

*Exemple : Get-Service*

* Start-Service : Démarre un service.

*Exemple : Start-Service -Name "nomDuService"*

* Stop-Service : Arrête un service en cours d'exécution.

*Exemple : Stop-Service -Name "nomDuService"*

* Restart-Service : Redémarre un service.

*Exemple : Restart-Service -Name "nomDuService"*

* Set-Service : Modifie les propriétés d'un service. Exemple :

*Set-Service -Name "nomDuService" -StartupType Automatic*

* New-Service : Crée un nouveau service. Exemple :

*New-Service -Name "MonService" -BinaryPathName "C:\Chemin\vers\service.exe"*

* Get-Service | Where-Object { $\_.Status -eq "Running" } : Filtre les services en cours d'exécution. *Exemple : Get-Service | Where-Object { $\_.Status -eq "Running" }*
* Get-Service | Sort-Object Status : Trie les services par leur état.

*Exemple : Get-Service | Sort-Object Status*

**la gestion des utilisateurs et des groupes**

1. **Gestion des utilisateurs :**

* Get-LocalUser : Affiche les utilisateurs locaux sur l'ordinateur.

*Exemple : Get-LocalUser*

* New-LocalUser : Crée un nouvel utilisateur local.

*Exemple : New-LocalUser -Name "nomUtilisateur"*

* Remove-LocalUser : Supprime un utilisateur local.

*Exemple : Remove-LocalUser -Name "nomUtilisateur"*

* Set-LocalUser : Modifie les propriétés d'un utilisateur local.

*Exemple : Set-LocalUser -Name "nomUtilisateur" -Description "Nouvelle description"*

* Disable-LocalUser : Désactive un utilisateur local. Exemple :

*Disable-LocalUser -Name "nomUtilisateur"*

* Enable-LocalUser : Active un utilisateur local.

*Exemple : Enable-LocalUser -Name "nomUtilisateur"*

1. **Gestion des groupes :**

* Get-LocalGroup : Affiche les groupes locaux sur l'ordinateur.

*Exemple : Get-LocalGroup*

* New-LocalGroup : Crée un nouveau groupe local.

*Exemple : New-LocalGroup -Name "nomGroupe"*

* Remove-LocalGroup : Supprime un groupe local.

*Exemple : Remove-LocalGroup -Name "nomGroupe"*

* Add-LocalGroupMember : Ajoute un membre à un groupe local.

*Exemple : Add-LocalGroupMember -Group "nomGroupe" -Member "nomUtilisateur"*

* Remove-LocalGroupMember : Supprime un membre d'un groupe local.

*Exemple : Remove-LocalGroupMember -Group "nomGroupe" -Member "nomUtilisateur"*

* Set-LocalGroup : Modifie les propriétés d'un groupe local.

*Exemple : Set-LocalGroup -Name "nomGroupe" -Description "Nouvelle description"*

* Get-LocalGroupMember : Obtient les membres d'un groupe local.

*Exemple : Get-LocalGroupMember -Group "nomGroupe"*

1. **Reseaux :**
2. **Configuration réseau :**

* Get-NetIPAddress : Affiche les adresses IP configurées sur l'ordinateur.

*Exemple : Get-NetIPAddress*

* Get-NetAdapter : Affiche les informations sur les adaptateurs réseau disponibles sur l'ordinateur.

*Exemple : Get-NetAdapter*

* Set-NetIPAddress : Configure une adresse IP pour un adaptateur réseau spécifique.

*Exemple : Set-NetIPAddress -InterfaceIndex 3 -IPAddress 192.168.1.100 -PrefixLength 24*

* Set-DnsClientServerAddress : Configure les serveurs DNS pour un adaptateur réseau.

*Exemple : Set-DnsClientServerAddress -InterfaceIndex 3 -ServerAddresses ("192.168.1.1", "8.8.8.8")*

1. **Gestion des connexions réseau :**

* Test-NetConnection : Teste la connectivité réseau vers un hôte spécifié.

*Exemple : Test-NetConnection -ComputerName example.com -Port 80*

* New-NetTCPConnection : Crée une nouvelle connexion TCP vers un hôte et un port spécifiés.

*Exemple : New-NetTCPConnection -ComputerName example.com -Port 80*

* Get-NetTCPConnection : Affiche les connexions TCP actives sur l'ordinateur.

*Exemple : Get-NetTCPConnection*

* Get-NetUDPEndpoint : Affiche les informations sur les points de terminaison UDP ouverts sur l'ordinateur.

*Exemple : Get-NetUDPEndpoint*

1. **Gestion des pare-feux :**

* Obtenir l'état du pare-feu Windows :
* Get-NetFirewallProfile | Select-Object Name, Enabled
* Description : Cette commande récupère l'état du pare-feu Windows pour différents profils réseau (Domaine, Privé, Public).
* Explication : Elle indique si le pare-feu est activé ou désactivé pour chaque profil réseau.
* Lister toutes les règles de pare-feu :
* Get-NetFirewallRule
* Description : Cette commande répertorie toutes les règles de pare-feu configurées sur le système.
* Explication : Elle affiche une liste complète des règles de pare-feu avec des détails tels que le nom, la direction, l'action et l'état activé.
* Activer ou désactiver le pare-feu Windows :
* Set-NetFirewallProfile -Profile <NomProfil> -Enabled <True/False>
* Description : Cette commande active ou désactive le pare-feu Windows pour un profil réseau spécifique.
* Explication : Remplacez <NomProfil> par 'Domaine', 'Privé' ou 'Public', et <True/False> par $true pour activer ou $false pour désactiver.
* Créer une nouvelle règle de pare-feu entrante :
* New-NetFirewallRule -DisplayName "MaRègle" -Direction Inbound -Action Allow -Protocol TCP -LocalPort 80
* Description : Cette commande crée une nouvelle règle de pare-feu entrante permettant le trafic sur le port TCP 80.
* Explication : Remplacez "MaRègle" par un nom descriptif pour votre règle. Cet exemple autorise le trafic entrant sur le port 80.
* Supprimer une règle de pare-feu :
* Remove-NetFirewallRule -Name "MaRègle"
* Description : Cette commande supprime une règle de pare-feu avec le nom spécifié.
* Explication : Remplacez "MaRègle" par le nom réel de la règle que vous souhaitez supprimer.
* Afficher des informations détaillées sur une règle de pare-feu :
* Get-NetFirewallRule -DisplayName "MaRègle" | Format-List
* Description : Cette commande fournit des informations détaillées sur une règle de pare-feu spécifique.
* Explication : Remplacez "MaRègle" par le nom de la règle dont vous souhaitez obtenir des détails.
* Réinitialiser les paramètres du pare-feu Windows par défaut :
* Reset-NetFirewallRule
* Description : Cette commande réinitialise toutes les règles de pare-feu à leurs paramètres par défaut.
* Explication : Utilisez avec prudence, car cela supprimera toutes les règles personnalisées que vous avez créées.

1. **Surveillance réseau :**

* Get-NetEventSession : Affiche les sessions de journalisation des événements réseau.

*Exemple : Get-NetEventSession*

* Start-NetEventSession : Démarre une session de journalisation des événements réseau.

*Exemple : Start-NetEventSession -Name "NetworkLogging" -CaptureMode DomainJoined -LocalFilePath C:\Logs*

* Stop-NetEventSession : Arrête une session de journalisation des événements réseau.

*Exemple : Stop-NetEventSession -Name "NetworkLogging"*

1. **Gestion des VPN (Réseau Privé Virtuel) :**

* Add-VpnConnection : Ajoute une nouvelle connexion VPN.

*Exemple : Add-VpnConnection -Name "VPNConnection" -ServerAddress "vpn.example.com" -TunnelType "SSTP"*

* Get-VpnConnection : Affiche les connexions VPN configurées sur l'ordinateur.

*Exemple : Get-VpnConnection*

* Remove-VpnConnection : Supprime une connexion VPN configurée sur l'ordinateur.

*Exemple : Remove-VpnConnection -Name "VPNConnection"*

1. **Gestion des paramètres réseau avancés :**

* Get-NetIPInterface : Affiche les interfaces IP configurées sur l'ordinateur.

*Exemple : Get-NetIPInterface*

* Get-NetIPv4Protocol : Affiche les paramètres IPv4 configurés sur l'ordinateur.

*Exemple : Get-NetIPv4Protocol*

* Set-NetIPv4Protocol : Modifie les paramètres IPv4 configurés sur l'ordinateur.

*Exemple : Set-NetIPv4Protocol -DHCP Disable*

1. **Autre Commandes :**

• Ping : teste la connexion avec une autre machine

• Netstat : fournit les informations sur les connexions réseau d’une machine

• Arp : gestion du cache ARP de la machine

• Hostname : fournit le nom de la machine sur le réseau

• Tracert : montre les chemin utilisé par un paquet IP entre 2 machines

• IPConfig : affiche la configuration IP de la machine

• NSLookup : fournit les informations sur le DNS

• Route : affiche la table de routage

• NetDiag : fournit un diagnostic réseau

**Commandes de base :**

* Get-Help : Affiche l'aide pour les cmdlets PowerShell.

*Exemple : Get-Help Get-Process*

* Get-Command : Affiche la liste de toutes les commandes PowerShell disponibles.

*Exemple : Get-Command*

* Get-Date : Affiche la date et l'heure actuelles.

*Exemple : Get-Date*

* Get-Location : Affiche le répertoire de travail actuel.

*Exemple : Get-Location*

* Set-Location : Modifie le répertoire de travail actuel.

*Exemple : Set-Location C:\Windows*

* Clear-Host : Efface la fenêtre de la console.

*Exemple : Clear-Host*

* Write-Host : Affiche du texte dans la console.

*Exemple : Write-Host "Hello, World!"*

* Get-Process : Affiche les processus en cours d'exécution sur l'ordinateur.

*Exemple : Get-Process*

* Stop-Process : Arrête un processus en cours d'exécution.

*Exemple : Stop-Process -Name "nomDuProcessus"*

* Start-Process : Démarre un nouveau processus.

*Exemple : Start-Process -FilePath "C:\Chemin\vers\application.exe"*

* Get-Service : Affiche les services en cours d'exécution sur l'ordinateur.

*Exemple : Get-Service*

* Stop-Service : Arrête un service en cours d'exécution.

*Exemple : Stop-Service -Name "nomDuService"*

* Start-Service : Démarre un service.

*Exemple : Start-Service -Name "nomDuService"*

* Get-ChildItem : Affiche les fichiers et dossiers dans un répertoire spécifié.

*Exemple : Get-ChildItem C:\*

* Copy-Item : Copie un fichier ou un dossier vers un nouvel emplacement.

*Exemple : Copy-Item C:\Fichier.txt D:\*

* Move-Item : Déplace un fichier ou un dossier vers un nouvel emplacement.

*Exemple : Move-Item C:\Fichier.txt D:\*

* Remove-Item : Supprime un fichier ou un dossier.

*Exemple : Remove-Item C:\Fichier.txt*

* Get-Content : Affiche le contenu d'un fichier.

*Exemple : Get-Content C:\Fichier.txt*

* Set-Content : Modifie le contenu d'un fichier.

*Exemple : Set-Content C:\Fichier.txt "Nouveau contenu"*

* Clear-Content : Efface le contenu d'un fichier.

*Exemple : Clear-Content C:\Fichier.txt*

1. **Creation d’un script Powershell**
2. **Variables :**

* $variable : Définit une variable.
* $null : Représente une valeur nulle ou non définie.

1. **Entrées et sorties :**

* Read-Host : Lit une entrée de l'utilisateur dans la console.
* Write-Host : Affiche du texte dans la console.
* Write-Output : Envoie un objet à la sortie.

1. **Structures de contrôle :**

* if/else : Exécute des blocs de code conditionnellement.

if ($condition) {

# Bloc de code si la condition est vraie

} else {

# Bloc de code si la condition est fausse

}

* switch : Exécute des blocs de code en fonction de la valeur d'une expression.

switch ($variable) {

"valeur1" { # Bloc de code si la valeur est "valeur1" }

"valeur2" { # Bloc de code si la valeur est "valeur2" }

default { # Bloc de code par défaut }

}

1. **Boucles :**

* for : Exécute un bloc de code un nombre spécifié de fois.

for ($i = 1; $i -le 5; $i++) {

# Bloc de code à répéter

}

* foreach : Parcourt les éléments d'une collection et exécute un bloc de code pour chaque élément.

foreach ($item in $collection) {

# Bloc de code à exécuter pour chaque élément

}

* while : Exécute un bloc de code tant qu'une condition spécifiée est vraie.

while ($condition) {

# Bloc de code à répéter tant que la condition est vraie

}

1. **Tableaux :**

* @() : Définit un tableau.

$tableau = @("élément1", "élément2", "élément3")

* $tableau[index] : Accède à un élément spécifique du tableau.

$premierElement = $tableau[0]

* foreach ($element in $tableau) : Parcourt tous les éléments du tableau.

foreach ($element in $tableau) {

Write-Output $element

}

* $tableau.Length : Obtient le nombre d'éléments dans le tableau.

$nombreElements = $tableau.Length

1. **Opérations logiques :**

* -and : Opérateur logique ET.
* -or : Opérateur logique OU.
* -not : Opérateur logique NON.

1. **Opérations de comparaison :**

* -eq : Égal à.
* -ne : Différent de.
* -gt : Supérieur à.
* -lt : Inférieur à.
* -ge : Supérieur ou égal à.
* -le : Inférieur ou égal à.

1. **Opérations mathématiques :**

* + : Addition.
* - : Soustraction.
* \* : Multiplication.
* / : Division.
* % : Modulo (reste de la division).