

Script

- Bash
 - Script Firewall
- PowerShell
 - Script Sauvegarde de partage

Bash

Script Firewall

Update d'un script bash FireWall :

Ancien Code FireWall :

```
@echo ===== SQL Server Ports =====
@echo Enabling SQLServer default instance port 1433
netsh advfirewall firewall add rule name="SQLServer" dir=in action=allow protocol=TCP localport=1433
@echo Enabling Dedicated Admin Connection port 1434
netsh advfirewall firewall add rule name="SQL Admin Connection" dir=in action=allow protocol=TCP
localport=1434
@echo Enabling conventional SQL Server Service Broker port 4022
netsh advfirewall firewall add rule name="SQL Service Broker" dir=in action=allow protocol=TCP
localport=4022
@echo Enabling Transact-SQL Debugger/RPC port 135
netsh advfirewall firewall add rule name="SQL Debugger/RPC" dir=in action=allow protocol=TCP localport=135
@echo ===== Analysis Services Ports =====
@echo Enabling SSAS Default Instance port 2383
netsh advfirewall firewall add rule name="Analysis Services" dir=in action=allow protocol=TCP localport=2383
@echo Enabling SQL Server Browser Service port 2382
netsh advfirewall firewall add rule name="SQL Browser" dir=in action=allow protocol=TCP localport=2382
@echo ===== Misc Applications =====
@echo Enabling HTTP port 80
netsh advfirewall firewall add rule name="HTTP" dir=in action=allow protocol=TCP localport=80
@echo Enabling SSL port 443
netsh advfirewall firewall add rule name="SSL" dir=in action=allow protocol=TCP localport=443
@echo Enabling port for SQL Server Browser Service's 'Browse' Button
netsh advfirewall firewall add rule name="SQL Browser" dir=in action=allow protocol=UDP localport=1434
@echo Allowing multicast broadcast response on UDP (Browser Service Enumerations OK)
netsh firewall set multicastbroadcastresponse ENABLE
@echo Ajout regle SQLServer 2019. Attention: le chemin contient le nom de l'instance (MSSQL15.SQLEXPRESS)
netsh advfirewall firewall add rule name="SqlServer 2019" dir=in action=allow program="C:\Program
Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\Binn\SqLservr.exe" enable=yes profile=private
```

Pause

Réponse du Script :

IMPORTANT: Command executed successfully. However, "**netsh** firewall" is deprecated; use "**netsh advfirewall firewall**" instead. For more information on using "**netsh advfirewall firewall**" commands instead of "**netsh firewall**", see KB article 947709 at <https://go.microsoft.com/fwlink/?linkid=121488>

Nouveau Code avec update de **netsh firewall** vers **netsh advfirewall firewall**

```
@echo off
set "ports=1433 1434 4022 135 2383 2382 80 443"
set "program=C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\Binn\SqlServr.exe"

for %%p in (%ports%) do (
    echo Enabling port %%p
    netsh advfirewall firewall add rule name="Port %%p" dir=in action=allow protocol=TCP localport=%%p
)

echo Enabling SQL Server Browser Service's 'Browse' Button for UDP 1434
netsh advfirewall firewall add rule name="SQL Browser UDP" dir=in action=allow protocol=UDP
localport=1434

echo Adding rule for %program%
netsh advfirewall firewall add rule name="SQL Server" dir=in action=allow program="%program%"
enable=yes profile=private

echo Allowing multicast broadcast response
netsh advfirewall firewall add rule name="Allow Multicast" dir=in action=allow protocol=UDP
localport=1434 enable=yes
Pause
```

Download : [firewall.bat](#)

Compte Rendu :

1. Liste de ports : Au lieu de taper manuellement chaque commande netsh pour chaque port, une liste de ports a été créée

```
set "ports=1433 1434 4022 135 2383 2382 80 443
```

Cette liste contient tous les ports que le script doit ouvrir.

2. Boucle for pour les ports : Une boucle **for** a été créée pour parcourir tous les ports de la liste définie précédemment. Pour chaque port, la commande netsh est exécutée pour permettre le trafic entrant pour ce port.

```
for %%p in (%ports%) do (  
    echo Enabling port %%p  
    netsh advfirewall firewall add rule name="Port %%p" dir=in action=allow protocol=TCP localport=%%p  
)
```

3. Service de navigation SQL Server : Cette commande a été laissée inchangée. Elle permet le trafic sur le port 1434 UDP, qui est utilisé par le service de navigation SQL Server.

```
echo Enabling SQL Server Browser Service's 'Browse' Button for UDP 1434  
netsh advfirewall firewall add rule name="SQL Browser UDP" dir=in action=allow protocol=UDP  
localport=1434
```

4. Programme SQL Server : Au lieu d'être divisée en plusieurs parties par la boucle for, la commande netsh pour le programme SQL Server est maintenant une seule commande qui utilise une variable pour le chemin du programme.

```
set "program=C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\Binn\SqIsvr.exe"  
echo Adding rule for %program%  
netsh advfirewall firewall add rule name="SQL Server" dir=in action=allow program="%program%"  
enable=yes profile=private
```

5. Réponse de diffusion multicast : La commande qui permet la réponse de diffusion multicast a été corrigée pour inclure l'argument "new", et cette commande n'autorise que le trafic sur le port 1434 UDP, au lieu de permettre tout le trafic comme auparavant.

```
echo Allowing multicast broadcast response  
netsh advfirewall firewall add rule name="Allow Multicast" dir=in action=allow protocol=UDP localport=1434  
enable=yes
```

Ces modifications rendent le script plus facile à comprendre et plus facile à maintenir à l'avenir. Au lieu de modifier plusieurs commandes si un port doit être ajouté ou supprimé, vous n'avez qu'à modifier la liste de ports. De même, si le chemin du programme SQL Server change, vous n'avez qu'à modifier la variable program.

PowerShell

Script Sauvegarde de partage

Basic :

export (basic) :

```
# Export-Shares.ps1
$shares = Get-WmiObject -Class Win32_Share
$shares | Select-Object Name, Path, Description, MaximumAllowed, InstallDate, Caption, Status, Type,
AllowMaximum | Export-Csv -Path "C:\exports\shares_export.csv" -NoTypeInfo
```

import (basic) :

```
# Import-Shares.ps1
$shares = Import-Csv -Path "C:\exports\shares_export.csv"

foreach ($share in $shares) {
    New-SmbShare -Name $share.Name -Path $share.Path -Description $share.Description -FullAccess "Everyone"
    if ($share.MaximumAllowed -ne $null) {
        Set-SmbShare -Name $share.Name -ConcurrentUserLimit $share.MaximumAllowed
    }
}

# Remarque : Ajustez les permissions d'accès selon vos besoins
```

Avec droits :

export :


```
# Export-Shares.ps1

$shares = Get-WmiObject -Class Win32_Share | Where-Object { $_.Type -eq 0 }

$shareInfo = foreach ($share in $shares) {
    $permissions = (Get-SmbShareAccess -Name $share.Name) | Select-Object Name, AccessControlType,
AccountName
    [PSCustomObject]@{
        Name          = $share.Name
        Path           = $share.Path
        Description    = $share.Description
        MaximumAllowed = $share.MaximumAllowed
        InstallDate    = $share.InstallDate
        Caption        = $share.Caption
        Status         = $share.Status
        Type           = $share.Type
        AllowMaximum    = $share.AllowMaximum
        Permissions    = $permissions
    }
}

$shareInfo | Export-Csv -Path "C:\exports\shares_export.csv" -NoTypeInfoInformation
```

import :

```
# Import-Shares.ps1

$shares = Import-Csv -Path "C:\exports\shares_export.csv"

foreach ($share in $shares) {
    New-SmbShare -Name $share.Name -Path $share.Path -Description $share.Description

    if ($share.MaximumAllowed -ne $null) {
        Set-SmbShare -Name $share.Name -ConcurrentUserLimit $share.MaximumAllowed
    }

    $permissions = ConvertFrom-Json $share.Permissions
    foreach ($permission in $permissions) {
        Grant-SmbShareAccess -Name $share.Name -AccountName $permission.AccountName -AccessRight
$permission.AccessControlType
    }
}
```

