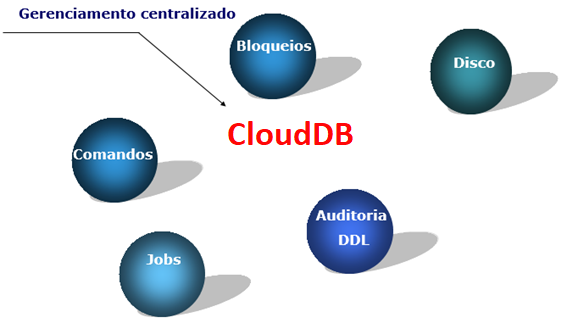


****

****

# Excluir arquivos por extensão com mais de X dias com o PowerShell

Iremos demonstrar um método para excluir arquivos de log em uma idade específica com uma extensão de arquivo específica no Windows Server. Isso parece uma tarefa excelente para o PowerShell e o Agendador de Tarefas do Windows.

Uma rápida pesquisa no Google rendeu muitas opções. O ponto crítico para este caso de uso foi o fato de que a pasta de log continha vários logs (diferentes nomes de arquivo e extensões) para várias finalidades. A maioria dos scripts e métodos que descobri levaram em consideração apenas a data (mais de X dias), mas não o nome ou a extensão do arquivo.

Get-ChildItem $Path | Where-Object { $\_.Name -like '\*.ext' } | Where-Object { $\_.LastWriteTime -lt $DatetoDelete } | Remove-Item

Isso adiciona um segundo “filtro” (Where-Object), baseado no nome do arquivo. A extensão do arquivo faz parte do atributo Name - no exemplo acima, usei .ext.

O script concluído ficou assim:

$Path = "C:\temp"

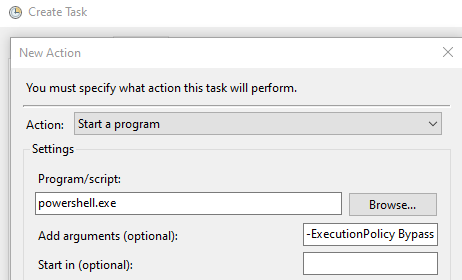
$Daysback = "-60"

$CurrentDate = Get-Date

$DatetoDelete = $CurrentDate.AddDays($Daysback)

Get-ChildItem $Path | Where-Object { $\_.Name -like '\*.ext' } | Where-Object { $\_.LastWriteTime -lt $DatetoDelete } | Remove-Item

Depois de criado, agendei no Agendador de Tarefas do Windows da seguinte forma:



**Program/script:** powershell.exe

**Add arguments (optional):** -ExecutionPolicy Bypass -File C:\scripts\clean-logs.ps1

Isso funcionou bem para o caso de uso em questão, mas provavelmente não seria dimensionado para várias extensões ou nomes de arquivo. Observe que, se você tiver caminhos/nomes de arquivos longos, Get-ChildItem lançará erros como os seguintes:

Get-ChildItem : Could not find a part of the path ''.

At

+ CategoryInfo: ReadError: (:String) [Get-ChildItem], DirectoryNotFoundException

+ FullyQualifiedErrorId : DirIOError,Microsoft.PowerShell.Commands.GetChildItemCommand

Get-ChildItem : The specified path, file name, or both are too long. The fully qualified file

name must be less than 260 characters, and the directory name must be less than 248 characters.

At

+ CategoryInfo: ReadError: (:String) [Get-ChildItem], PathTooLongException

+ FullyQualifiedErrorId : DirIOError,Microsoft.PowerShell.Commands.GetChildItemCommand

Você pode usar a solução alternativa -LiteralPath da seguinte maneira (dependendo da versão do Windows e do PowerShell):

$LiteralPath = "\\?\C:\temp"

$Daysback = "-60"

$CurrentDate = Get-Date

$DatetoDelete = $CurrentDate.AddDays($Daysback)

Get-ChildItem -LiteralPath $LiteralPath | Where-Object { $\_.Name -like '\*.ext' } | Where-Object { $\_.LastWriteTime -lt $DatetoDelete } | Remove-Item