Powershell + psExec

Launching Powershell Remotely via psExec

From Powershell: Launch Powershell Remotely via psExec

```
driveletter:\somepath\psexec.exe \\hostname cmd /c 'echo . | powershell.exe -command "$env:PROCESSOR_ARCHITECTURE; exit 100"'
```

From CMD: Launch Powershell Remotely via psExec

```
powershell.exe -command "& {driveletter:\somepath\psexec\psexec.exe \\hostname cmd /c 'echo . | powershell.exe -command "$env:PROCESSOR_ARCHITECTURE; exit 100"'}"
```

Case Use:
Remotely Stop an IIS Application Pool

```
powershell.exe -command "& {driveletter:\somepath\psexec.exe \\hostname cmd /c 'echo . | powershell.exe -command "set-executionpolicy Bypass -scope Process;Import-Module WebAdministration;Stop-WebAppPool -Name "MyAppPool"; exit 100"'}"
```

Opsview + Powershell

Powershell InvocationSyntax

The syntax for properly calling powershell scripts from the Opsview/NRPE agent is rather cryptic.

After much searching on the internets, this is the configuration syntax I found to work:
nsc.ini:

[nrpe handlers]

…

check_mycheck = cmd /c echo DriveLetter:\Path\Folder\MyScript.ps1; exit($lastexitcode) | powershell.exe -command -

Sources

Description	URL	Notes Check mit Powershell	http://www.monitoring-portal.org/wbb/index.php?page=Thread&threadID=15923	This website was in (German?); Google Search:"nsclient
.exe" "powershell" "socket timeout after" Nagios – check_veeam	https://www.angryadmin.co.uk/?tag=nagios	Google Search:"nsclient		
.exe" "$lastexitcode" Returning Exit Code from Script	http://powershell.com/cs/blogs/tips/archive/2009/05/18/returning-exit-code-from-script.aspx	Google Search: bat exit /b errorlevel Mailbox Health 2007.ps1	http://exchange.nagios.org/directory/Plugins/Email-and-Groupware/Microsoft-Exchange/Mailbox-Health-2007-2Eps1/details	Google Search: "nagios" "powershell" "exit code"

Automating IIS

This Article presents some examples of automating IIS using PowerShell and the appcmd.exe tool

References I used

Title	URL IIS AppCmd Quick Reference	http://blogs.msdn.com/b/mikezh/archive/2012/04/23/iis-appcmd-quick-reference.aspx ""	https://github.com/izabroski/OrchardBootstrap/blob/master/OrchardBootstrap.ps1 PowerShell Snap-in	http://www.iis.net/learn/manage/powershell/powershell-snap-in-creating-web-sites-web-applications-virtual-directories-and-application-pools

PowerShell Commands

This applies to my environment: Before you can utilize PowerShell commands for IIS automation, you'll need to ensure two things:

- PowerShell Script Execution is allowed for your session

[code language="powershell"]Set-Executionpolicy Bypass -Scope Process[/code]

- IIS Web Administration PowerShell module is properly loaded

[code language="powershell"]Import-Module WebAdministration[/code]

In the following end-to-end scenario we will execute the following steps:

1. Create new Application Pool
2. Create a new site
3. Create a new application
4. Assign the newly created Application to the Newly Created AppPool
5. Create two virtual directories
6. Assign permissions to these virtual directories

| No. | Action | Command | Create an IIS AppPool named MyAppPool | New-Item AppPools\MyAppPool Create an IIS Site named MySite with a physical path of C:\MySite | New-Item IIS:\Sites\MySite -bindings @{protocol="http";bindingInformation=":80:MySite"} -physicalPath C:\MySite Create an IIS Application named MyApplication with a physical path of C:\MySite\MyApplication | New-Item 'IIS:\Sites\MySite\MyApplication' -physicalPath C:\MySite\MyApplication -type Application Modify Application – Set AppPool for MyApplication to MyAppPool | Set-ItemProperty "IIS:\Sites\MySite\MyApplication" -name ApplicationPool -value MyAppPool Create a Virtual Directory MyVirtualDirectory under MySite\MyApplication pointing to C:\MySite\MyApplication\MyVirtualDirectory | New-Item 'IIS:\Sites\MySite\MyApplication\MyVirtualDirectory' -type VirtualDirectory -physicalPath "C:\MySite\MyApplication\MyVirtualDirectory" Assign permissions for MyApplication\MyVirtualDirectory | $file = $(get-item "IIS:\Sites\MySite\MyApplication\App_Data\") $dacl = $file.GetAccessControl() $newRule = New-Object Security.AccessControl.FileSystemAccessRule "BUILTIN\IIS_IUSRS", Write, Allow $modified = $false $dacl.ModifyAccessRule("Add", $newRule, [ref]$modified) $file.SetAccessControl($dacl) $file.GetAccessControl().GetAccessRules($true, $true, [System.Security.Principal.NTAccount])

Additional Commands

Action	Command Copy IIS AppPool MyAppPool to MyNewAppPool forcing overwrite	Copy IIS:\AppPools\MyAppPool IIS:\AppPools\MyNewAppPool –force Remove the site	Remove-Item IIS:\Sites\MySite Copy IIS Application MyApplication to MyNewApplication forcing overwrite	Copy "IIS:\Sites\MySite\MyApplication" "IIS:\Sites\MySite\MyNewApplication" -force Start IIS MySite	Start-WebItem 'IIS:\Sites\MySite' Stop IIS MySite	Stop-WebItem 'IIS:\Sites\MySite' Start IIS Application Pool	Start-WebAppPool -Name "MyAppPool" Stop IIS Application Pool	Stop-WebAppPool -Name "MyAppPool" Get Application Pool Status	Get-WebAppPoolState $appPoolName

Note: For the sake of example: Application Pool Name is **MyAppPool** and Site Name is **MySite**

IIS AppCmd Quick Reference

Action	Command Add Site	appcmd add site /name:MySite /bindings:http:/*:80 /physicalpath:"d:\MySitePath" Add App Pool	appcmd add apppool /name:MyAppPool /managedRuntimeVersion:v4.0 (e.g. targeting .NET 4.0) Set App Pool Credential	appcmd set config /section:applicationPools /[name='MyAppPool'].processModel.identityType:SpecificUser /[name='MyAppPool'].processModel.userName:MyDomain\MyAccount /[name='MyAppPool'].processModel.password:MyAccountPassword Add App	appcmd add app /site.name:"MySite" /path:/MyApp /physicalpath:"d:\MySitePath\MyApp" Assign/Change App Pool to an App	appcmd set app "MySite/MyApp" /applicationpool:MyAppPool List (App, Site, AppPool etc.)	appcmd list app appcmd list site appcmd list apppool /enable:Disable Anonymous Authentication (True to Enable, False to Disable)	appcmd set config "MySite/MyApp" -section:system.webServer/security/authentication/anonymousAuthentication /enabled:"True" /commit:apphost Enable Windows Authentication (True to Enable, False to Disable)	appcmd.exe set config "MySite/MyApp" -section:system.webServer/security/authentication/windowsAuthentication /enabled:"True" /commit:apphost Change Windows Authentication Providers (NTLM or Negotiate)	appcmd set config MySite/MyApp -section:system.webServer/security/authentication/windowsAuthentication /-providers /commit:apphost (clear provider list) appcmd set config MySite/MyApp -section:system.webServer/security/authentication/windowsAuthentication /+providers.[value='NTLM'] /commit:apphost (set to NTLM) appcmd set config MySite/MyApp -section:system.webServer/security/authentication/windowsAuthentication /providers:"NTLM,Negotiate"
/-providers.[value='Negotiate'] /commit:apphost (set to Negotiate) Add Custom Header – for example, noscript header or IE 9 compatable header	appcmd set config MySite /section:system.webServer/httpProtocol /customHeaders.[name='X-Content-Type-Options',value='noscript'] /commit:apphost									
appcmd set config MySite /section:httpProtocol /customHeaders.[name='X-UA-Compatible',value='IE=EmulateIE9'] /commit:apphost Add Default Document – error if it exists already	appcmd set config "MySite/MyApp" /section:defaultDocument /files.[value='default.aspx'] Delete App and Site – error if it doesn't exist	appcmd delete app "MySite/MyApp" appcmd delete site "MySite" Delete AppPool- error if it doesn't exist or it is used by app	appcmd delete apppool MyAppPool Backup and Restore IIS Settings	appcmd add backup MyBackup appcmd restore backup MyBackup HTTPS Binding If you are using HTTP over SSL	appcmd set site /site.name:"MyApp" /					
bindings.[protocol='https',bindingInformation='*:443:MySSLCertificate']										