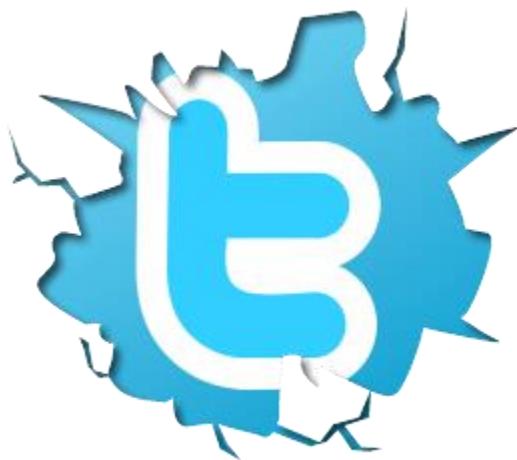


SQL Saturday Omaha

POWERSHELL FOR THE SQL SERVER DBA

Microsoft®
SQL Server®



Mike Fal - www.mikefal.net



Microsoft
CERTIFIED
Solutions Expert

Data Platform



@Mike_Fal

The fine print



Get-Agenda

Powershell Basics

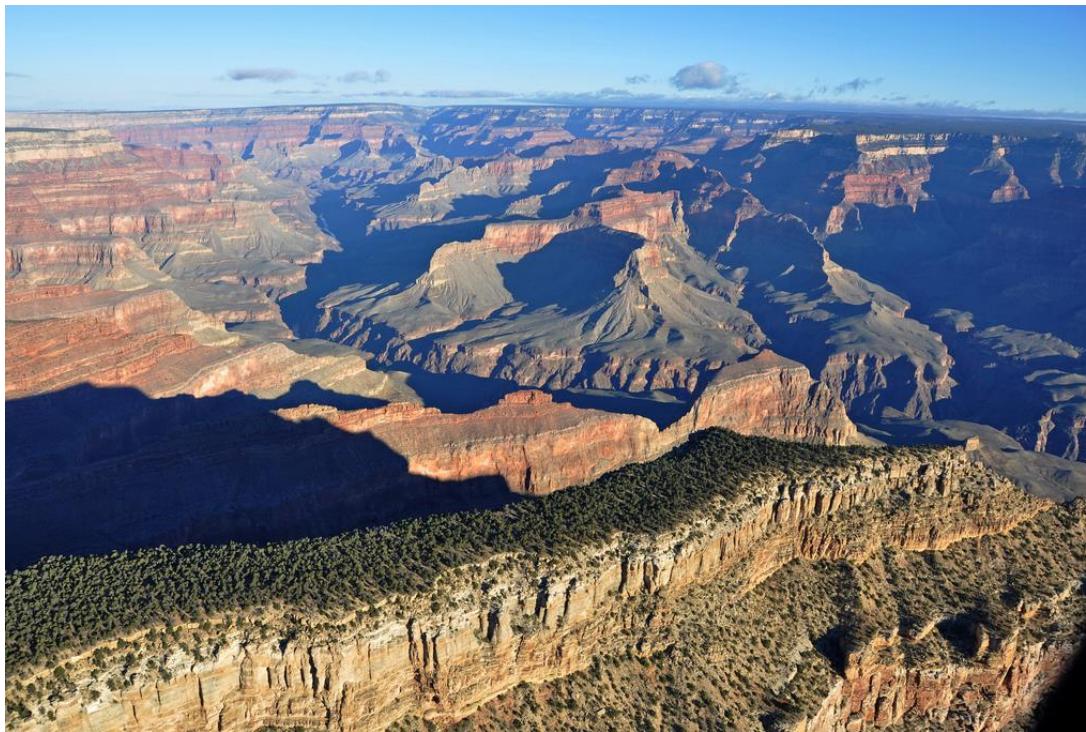
Powershell and SQL Server

Re-using Powershell

Practical Examples

And then what?

Ground Rules

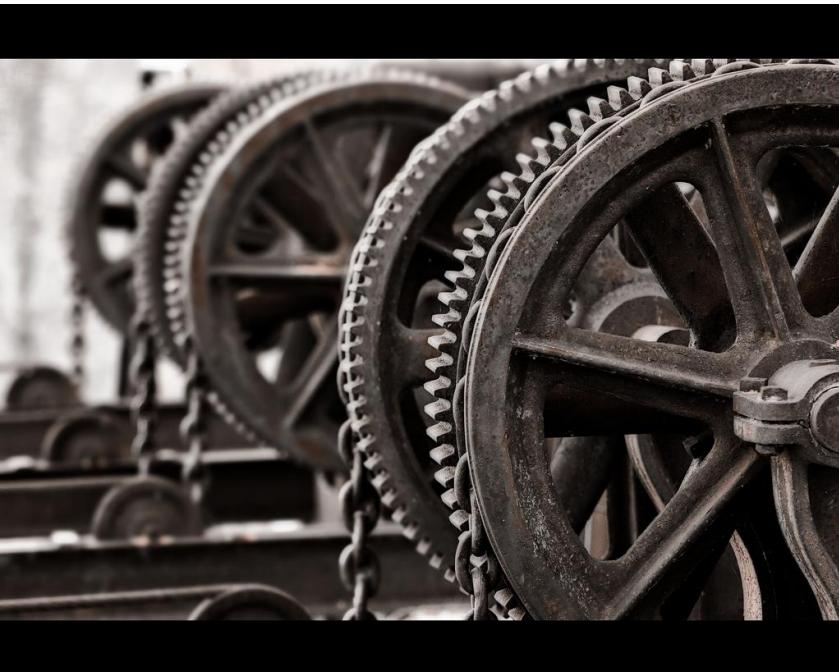


Don't focus on the code, focus on the concepts.
Ask questions!

What is Powershell?



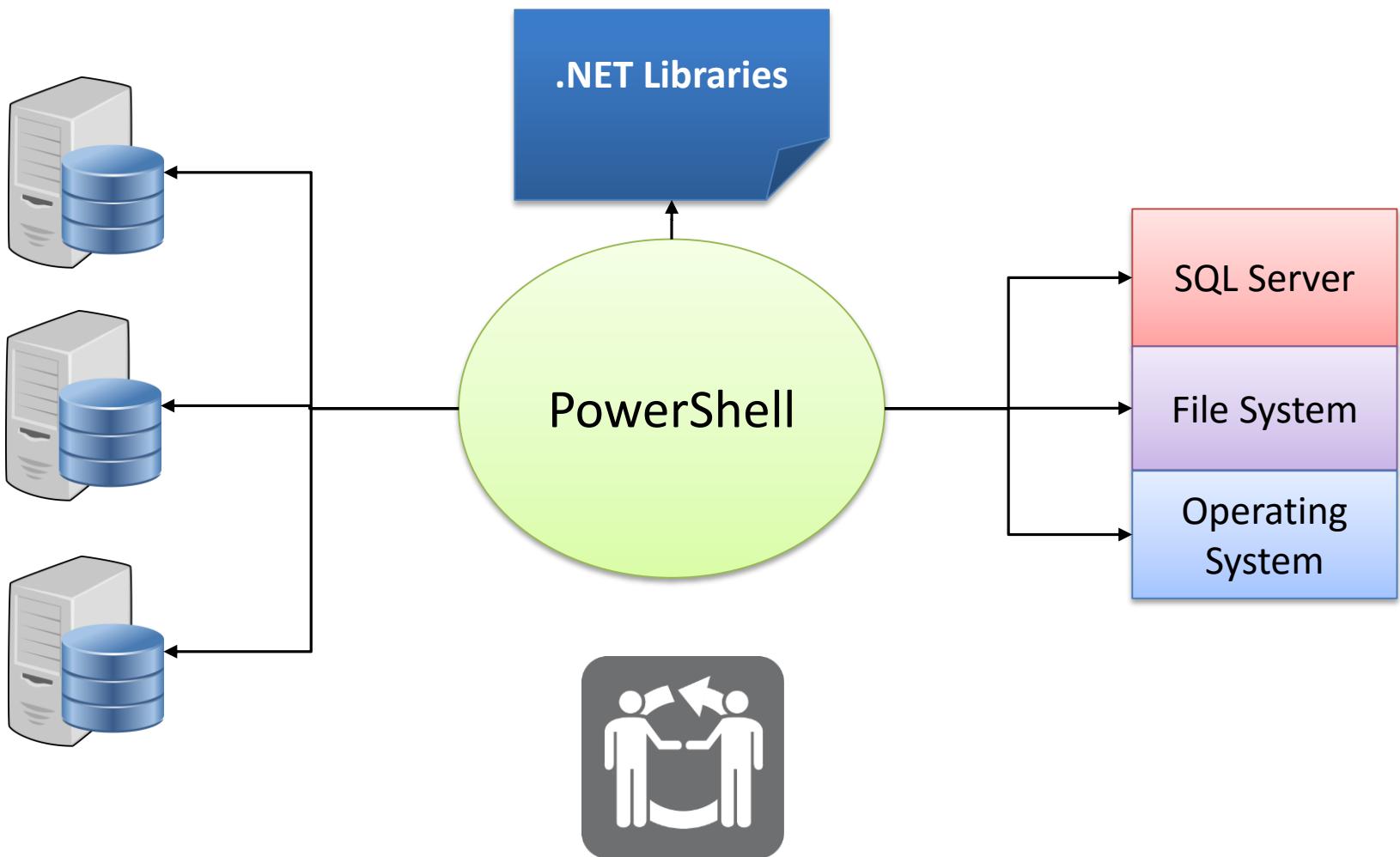
But for what?



The Facts

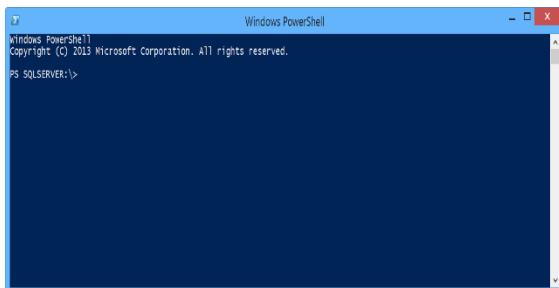
- Envisioned by Jeffery Snover – 2002
 - The Monad Manifesto
- Released as Powershell RC 1 – April 2006
 - Originally called Project “Monad”
- Current available version: 5.0
 - Just released, 4.0 has been out for a bit

Why PowerShell?

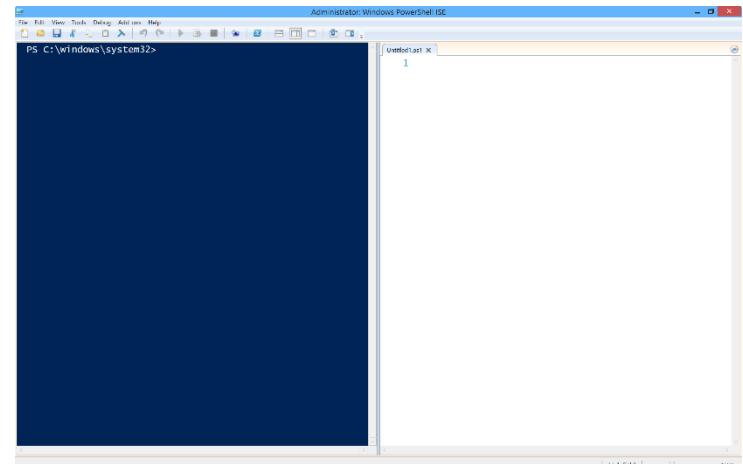


Pieces and Parts

Hosts



Powershell.exe



Powershell ISE

System.Management.Automation

Demo – The ISE

Administrator: Windows PowerShell ISE

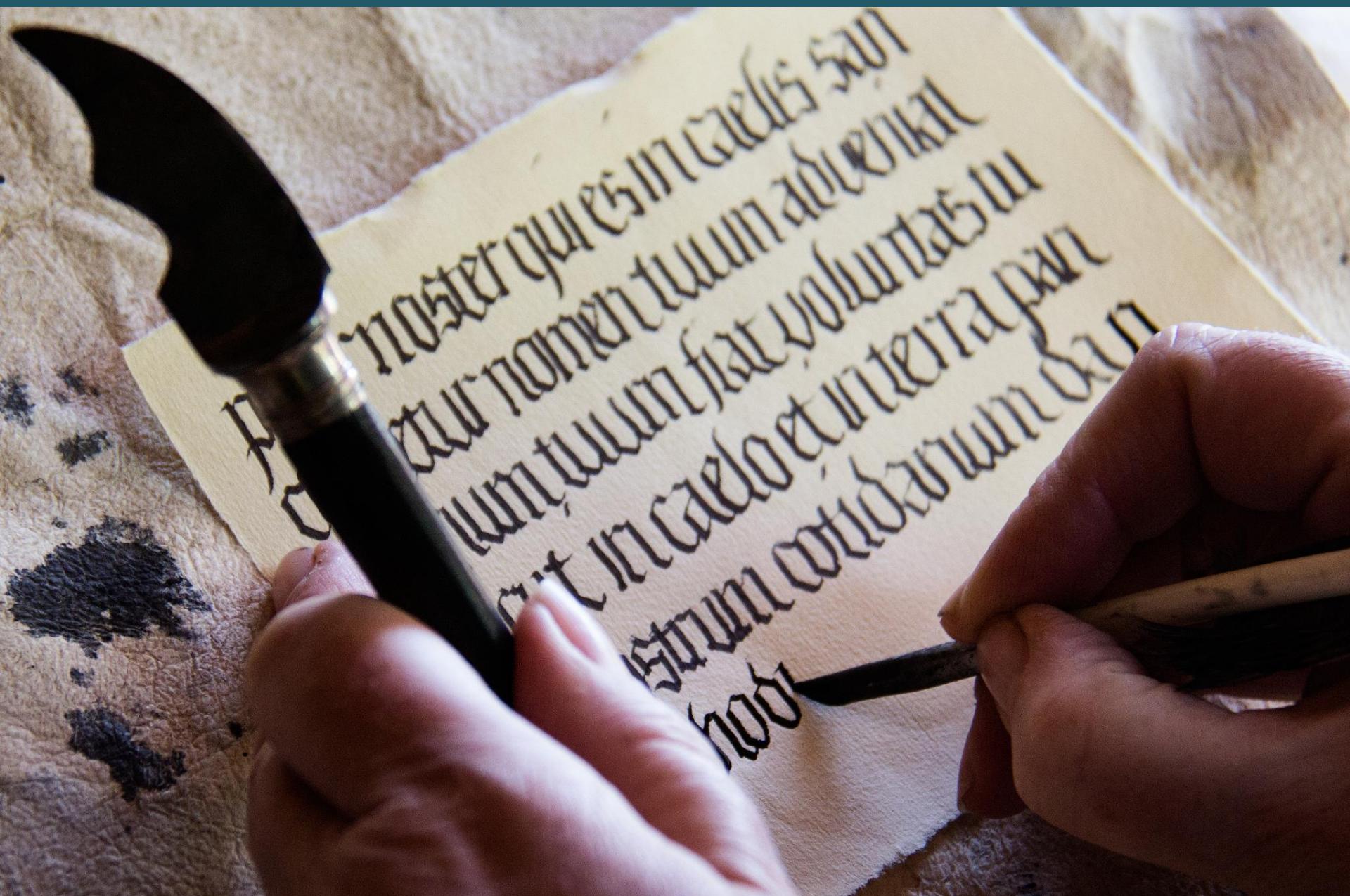
```
File Edit View Tools Debug Add-ons Help
BuildVMsScratch.ps1 X BuildAGscratch.ps1 Untitled1.ps1*(Recovered) Build-ClusterScratch.ps1*(Recovered) Build_HVLab.ps1 Powershell_tips_tricks.ps1*(Recovered) SQLBenchmarker.psm1
```

```
1 function New-LabVM{
2     param([parameter(Mandatory=$true)][string]$VMName
3           ,[ValidateSet('Full','Core')][string]$InstallType
4           )
5
6     $SourceVHD = switch($InstallType){
7         'Full'{'c:\vms\vhds\QM-Server2012Full.vhdx'}
8         'Core'{ 'c:\vms\vhds\QM-Server2012R2Core.vhdx'}
9     }
10
11 new-VM -Name $VMName -BootDevice CD -SwitchName "VMswitch"
12
13 Copy-Item $SourceVHD "C:\vms\vhds\$VMName.vhdx"
14 Add-VHHardDiskDrive -VMName $VMName -Path "C:\vms\vhds\$VMname.vhdx" -ControllerNumber 0 -ControllerLocation 0
15
16 Set-VMdvdDrive -vmname $VMName -Path C:\vms\ISOs\en_windows_server_2012_r2_with_update_x64_dvd_6052708.iso -ToControllerNumber 1 -ToControllerLocation 0
17 Set-VMMemory -VMName $VMName -DynamicMemoryEnabled $true
18 Rename-VMNetworkAdapter -VMName $VMName -Name 'Network Adapter' -NewName 'LocalNetwork'
19
20 Get-VM -Name $VMName | Start-VM
21
22
23 New-LabVM -VMName 'AD-Full' -InstallType Full
24 New-LabVM -VMName 'Picard-Core' -InstallType Core
25 New-LabVM -VMName 'Riker-Core' -InstallType Core
26
27 $pw='vanh0uten142' | ConvertTo-SecureString -AsPlainText -Force
28 $cred = New-Object System.Management.Automation.PSCredential ('$OFS\Administrator', $pw)
29 Add-Computer -DomainName 'SOF.local' -Credential $cred -NewName RIKER
30
31 $new='vanh0uten142' | ConvertTo-SecureString -AsPlainText -Force
```

PS C:\windows\system32>

Ln 1 Col 1 | 90%

Writing Powershell



The Lighting Tour of the Language

Cmdlets

Variables

Control Logic and Flow

Errors and Error Handling

cmdlets

Fundamental unit of “getting stuff done”

Verb-Noun

Limited by Microsoft

Unlimited values,
Should be descriptive

cmdlet examples

Get-Help

New-Item

Remove-Module

Learn Within Powershell

Get-Command

List Commands:

New

-Module SQLPS

Get-Help

Show help info:

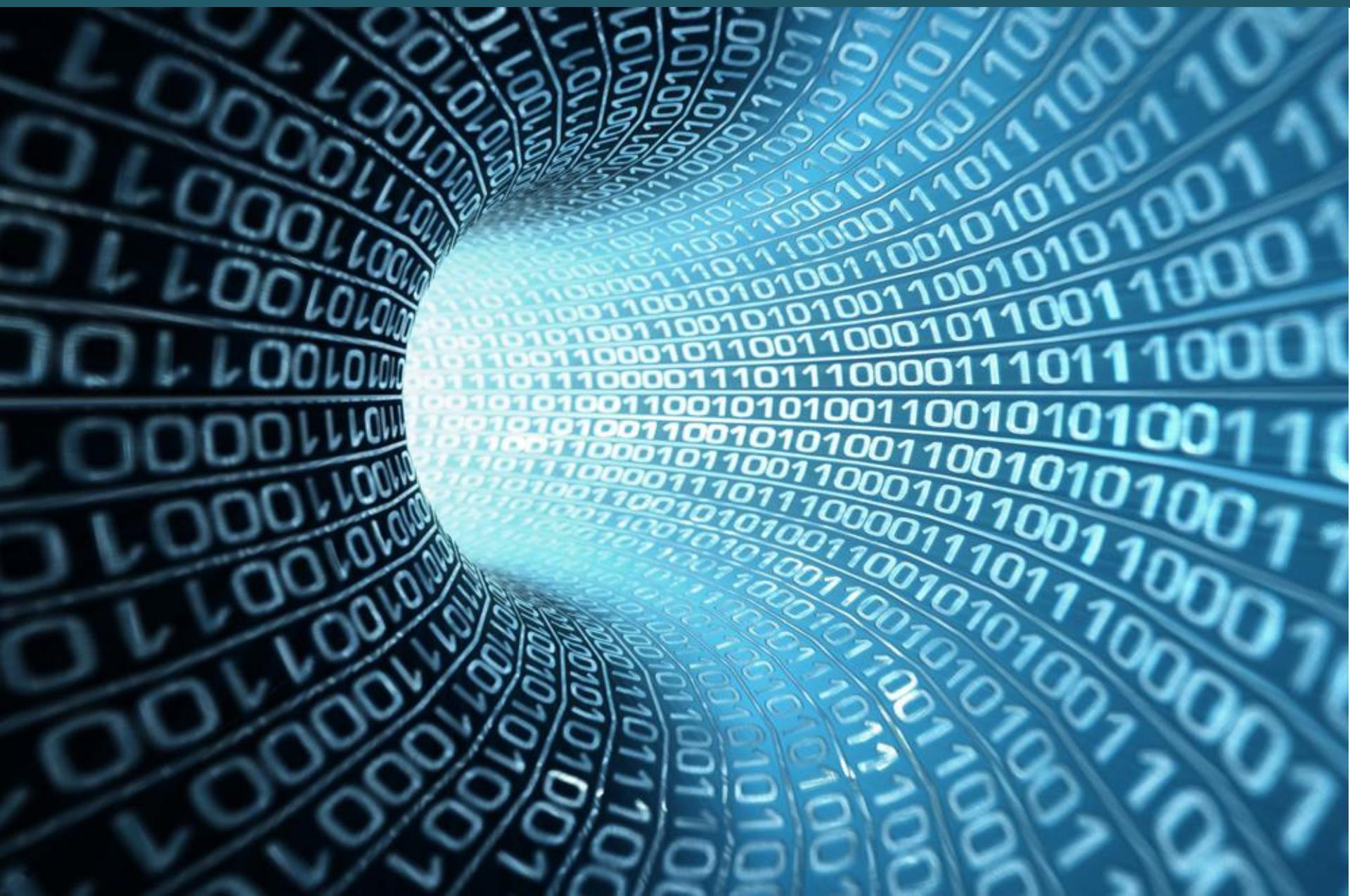
man, help

-ShowWindow

Get-Member

Methods and properties

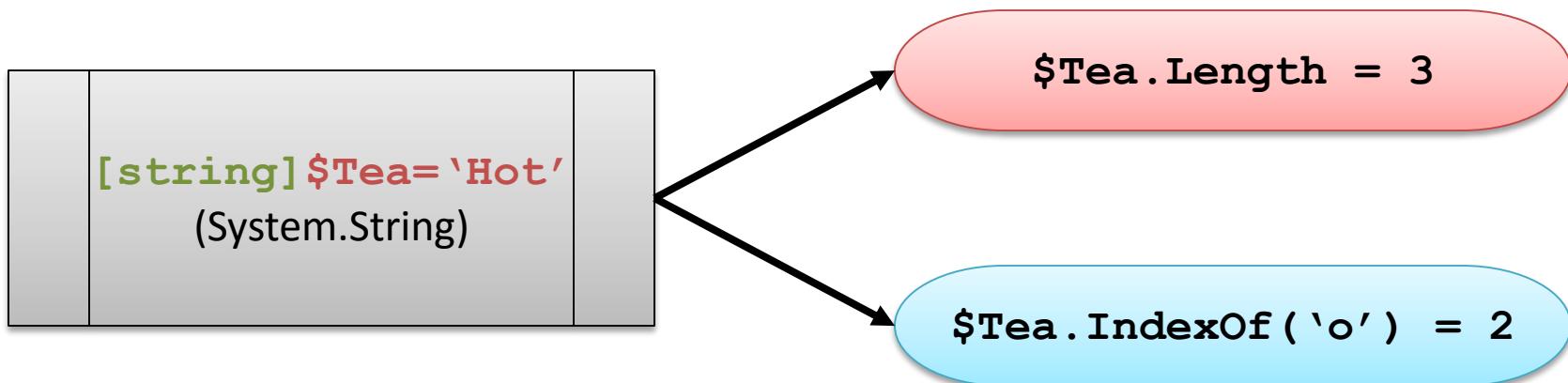
Demo - CMDLETS



Object Oriented Thinking

Everything is a .Net object!

- Properties (attributes)
- Methods (functions, do things)



Variables

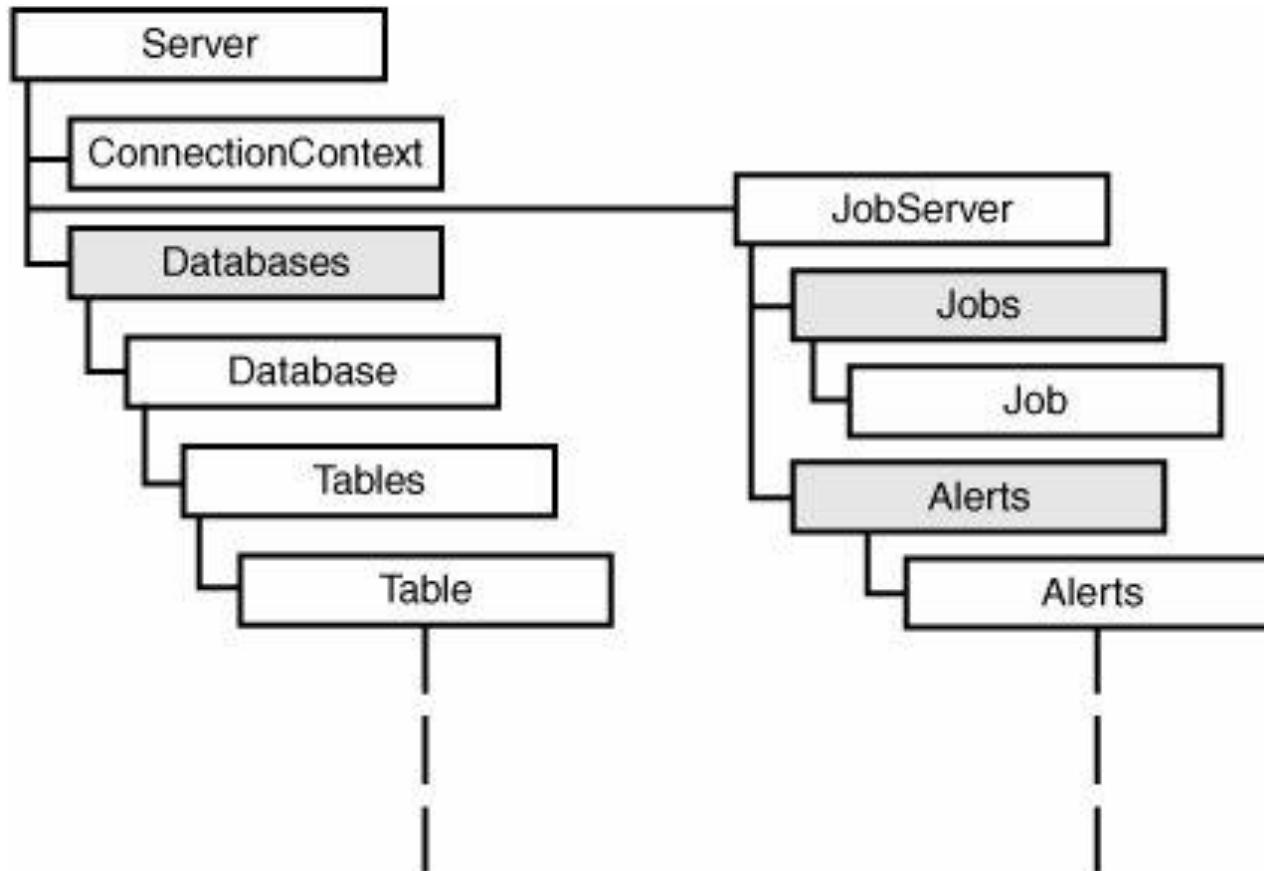
'\$' indicates a variable

```
[string]$Tea = 'Hot'
```

```
$Tea = 'Hot'
```

```
$TeaTime = Get-Date
```

Demo – Objects, and Variables

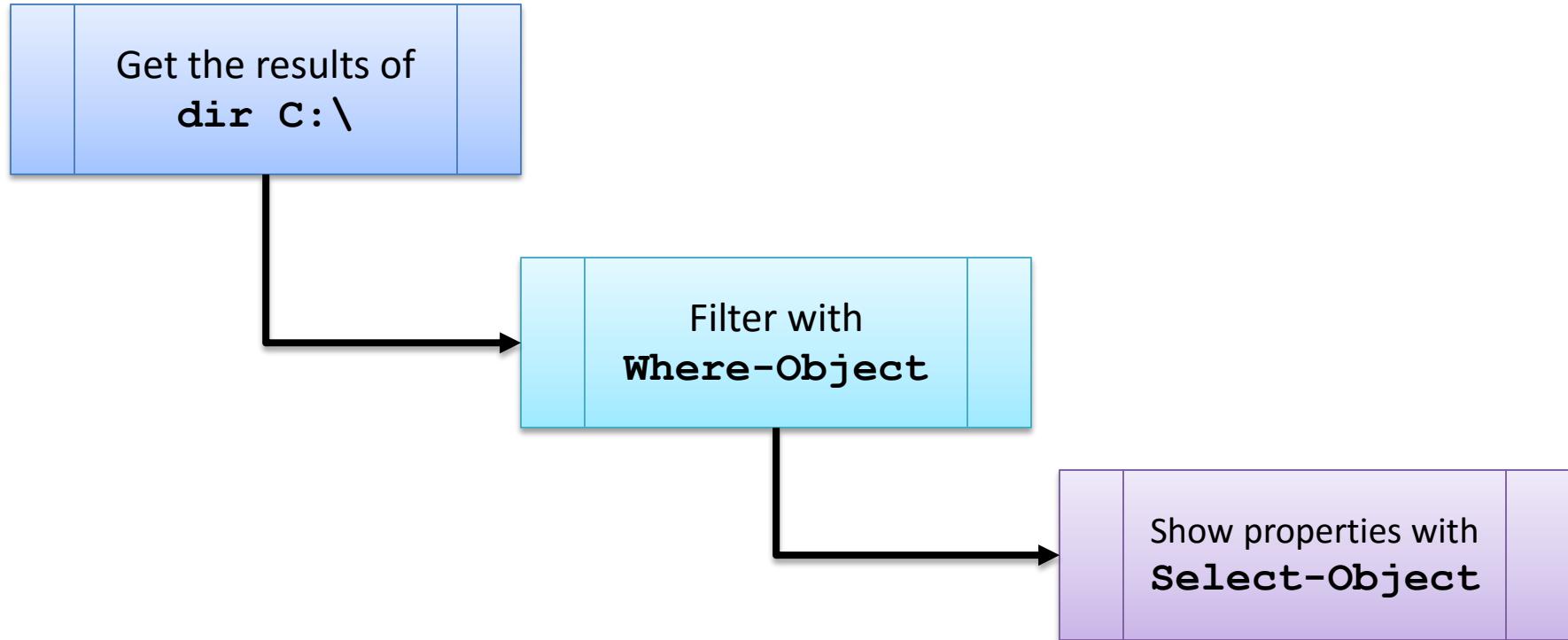


Pipeline



How does it work?

```
dir C:\ | Where-Object {$_._.PsIsContainer -eq $true} |  
Select-Object name
```

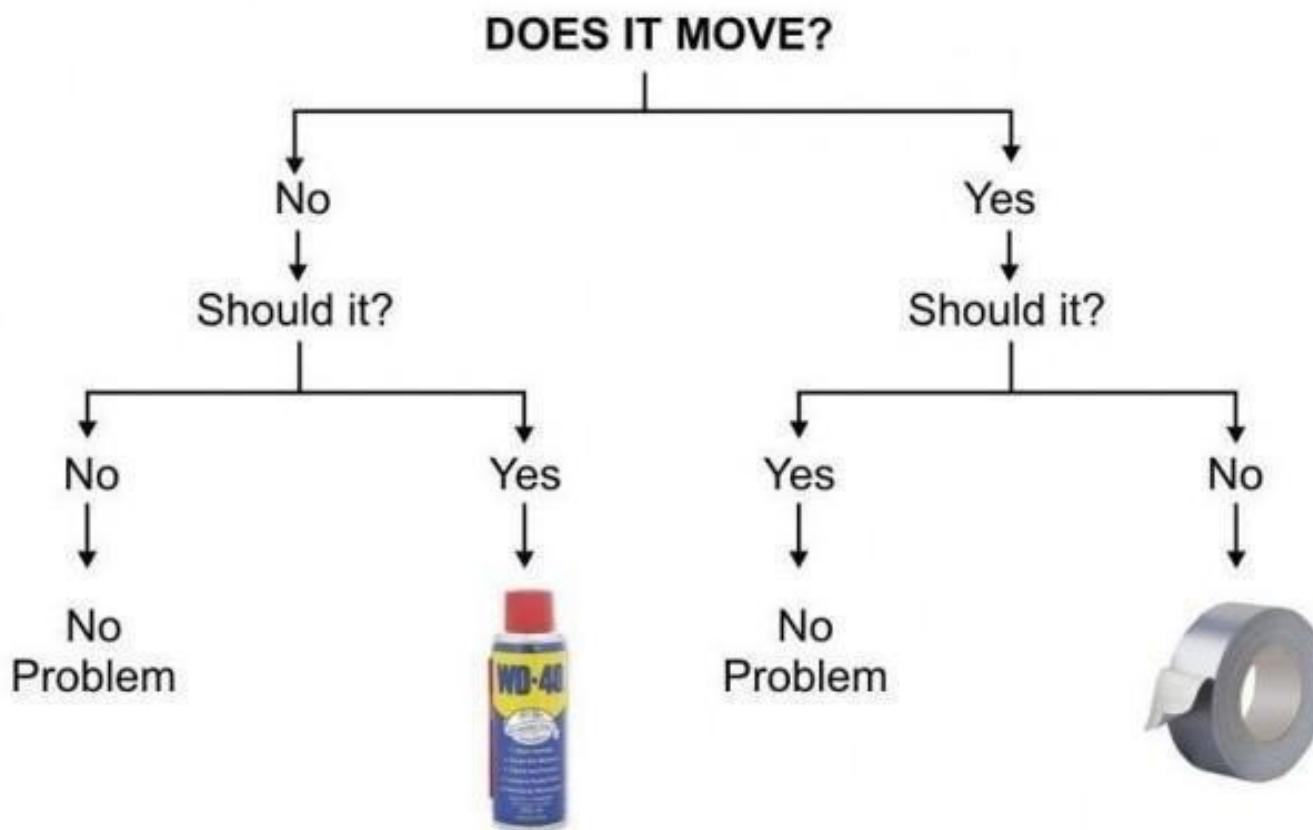


Demo - Pipeline



Control Logic

Engineering Flowchart



Operators

Shell Syntax

- >**: Redirect output, not greater than
- <**: Take input, not less than
- =**: Set value, not equality

Powershell Operators

- eq, -ne**: equality check
- lt, -le, -gt, -ge**: Less than and Greater than
- and, -or**: Combine conditions
- like, -notlike**: Wild card comparisons

Control Flow - Conditionals

```
If (<condition>) {  
    <script block>  
}  
  
Else {  
    <script block>  
}  
  
Switch (<item>) {  
    <value> { <script block> }  
    <value> { <script block> }  
    ...  
}
```

Control Flow - Loops

```
For (<start>;<check>;<iteration>) {  
    <script block>  
}
```

```
Do {  
    <script block>  
} While/Until (<condition>)
```

```
ForEach (<object> in <collection>) {  
    <script block>  
}
```

Break – Exit Loop

Error Handling

\$Error – System error collection

\$Error[0] – Always the last error

```
PS C:\windows\system32> 1/0
Attempted to divide by zero.
At Line:1 char:1
+ 1/0
+ ~~~
  + CategoryInfo          : NotSpecified: (:) [], RuntimeException
  + FullyQualifiedErrorId : RuntimeException
```

Error Handling

```
Try {  
    <script block>  
}  
Catch {  
    <script block>  
}  
Finally {  
    <script block>  
}
```

Throw – Create an exception

Demo – Control Flow/Error Handling



Review

What are the three key cmdlets for discovery in Powershell?

Get-Command, Get-Help, Get-Member

If you want to use a cmdlet to create something (file, directory, object), what verb will it use?

New

How can you extend the functionality of a command in one line?

Use the pipeline

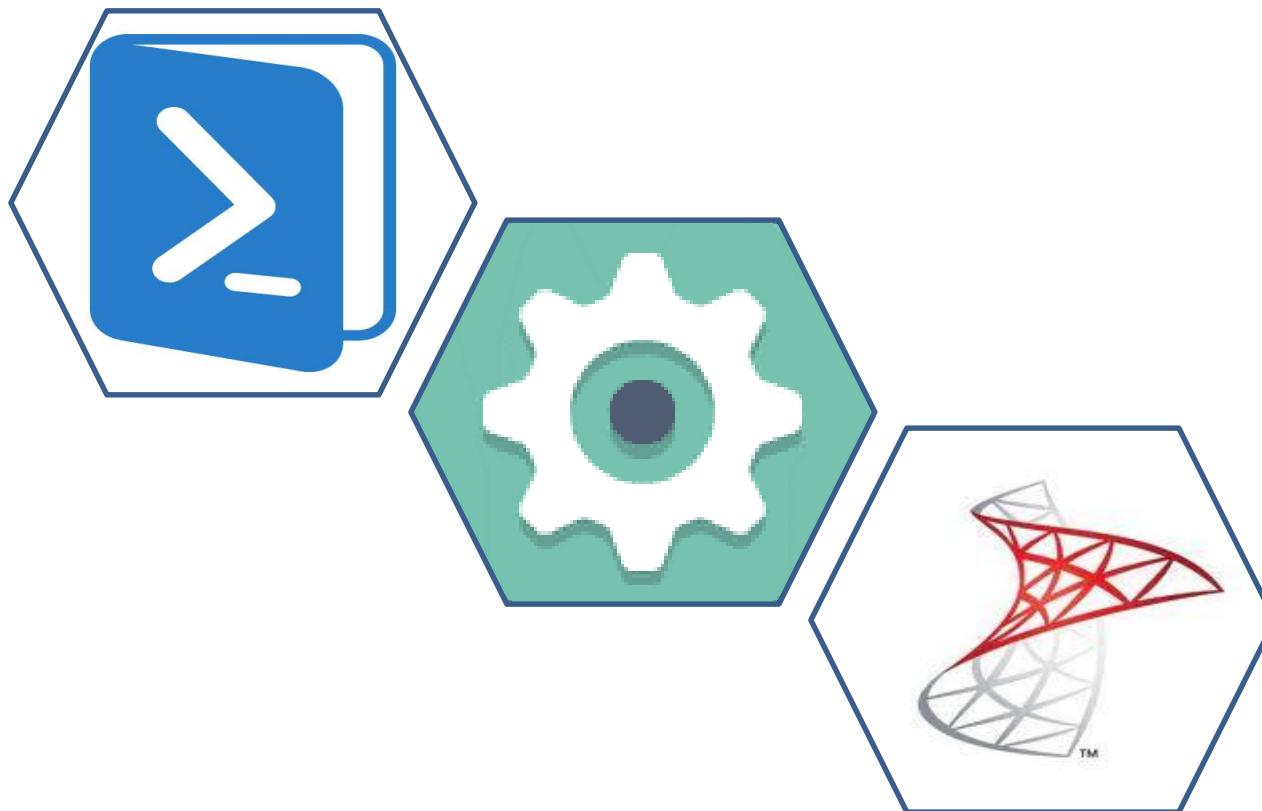
What would you use to loop through a collection of objects?

ForEach(){ }

Practical Use



Powershell And SQL Server



Ways to work with SQL Server

Command line (sqlcmd)

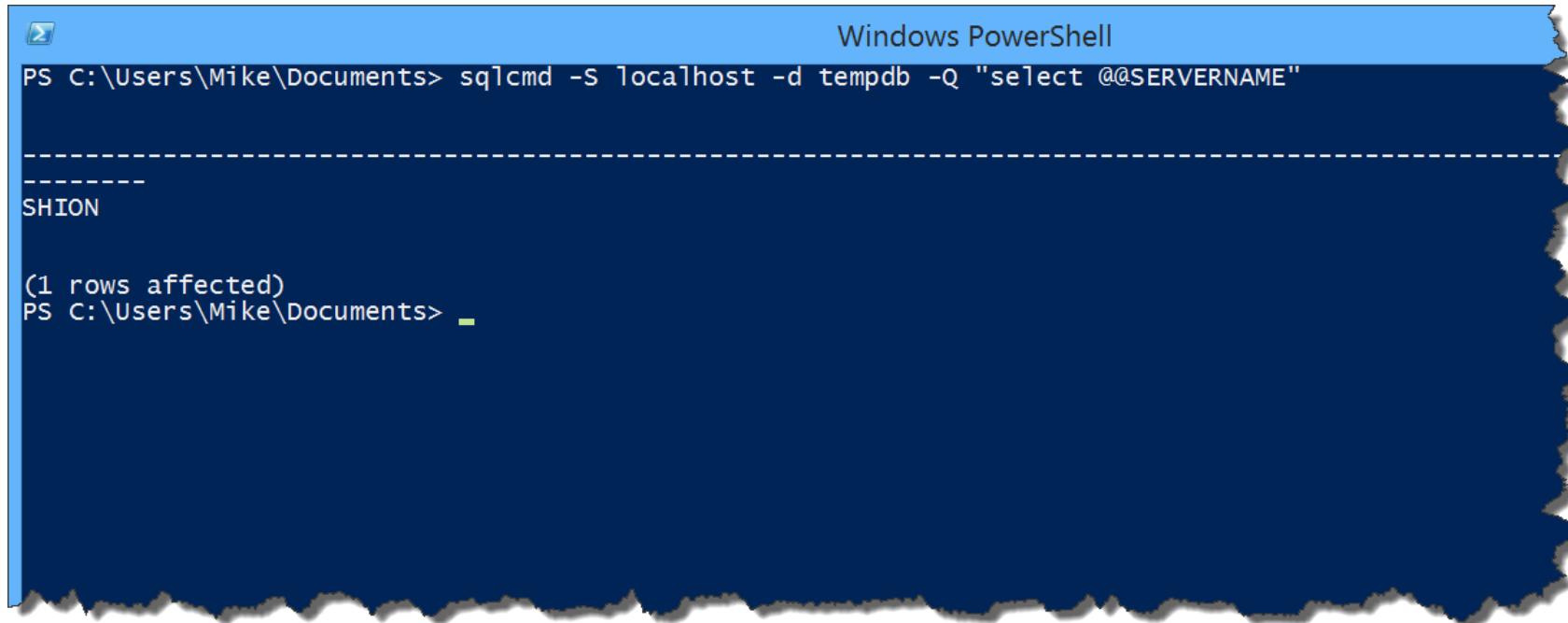
SQL Server provider (sqlps)

Server Management Objects (SMO)

Powershell in the SQL Agent



sqlcmd

A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window has a decorative torn paper effect along its right edge. The command entered is "sqlcmd -S localhost -d tempdb -Q "select @@SERVERNAME"" which returns the result "SHION" and "(1 rows affected)".

```
Windows PowerShell
PS C:\Users\Mike\Documents> sqlcmd -S localhost -d tempdb -Q "select @@SERVERNAME"
-----
SHION
(1 rows affected)
PS C:\Users\Mike\Documents>
```

Providers

Name	Provider	Root	CurrentLocation
A	Microsoft.PowerShell.Core\FileSystem	A:\	
Alias	Microsoft.PowerShell.Core\Alias		
C	Microsoft.PowerShell.Core\FileSystem	C:\	Users\Administrator
Cert	Microsoft.PowerShell.Security\Certificate	\	
D	Microsoft.PowerShell.Core\FileSystem	D:\	
Env	Microsoft.PowerShell.Core\Environment		
Function	Microsoft.PowerShell.Core\Function		
HKCU	Microsoft.PowerShell.Core\Registry	HKEY_CURRENT_USER	
HKLM	Microsoft.PowerShell.Core\Registry	HKEY_LOCAL_MACHINE	
Variable	Microsoft.PowerShell.Core\Variable		
WSMan	Microsoft.WSMan.Management\WSMan		

Windows Components as Drives

`Get-PSDrive`

`Explore/Use like a FileSystem`

SQL Server Provider

Import-Module SQLPS

Included with SQL 2012+ client

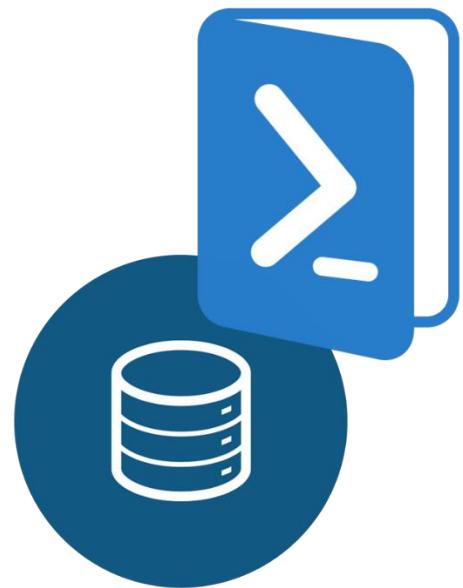
(Always use the most current client tools)

PS SQLSERVER:\sql\localhost\default



Provider Folder Host/Server Instance

Invoke-SqlCmd



- Works almost like sqlcmd
- Some differences
 - Returns objects
 - Not interactive
 - Not all sqlcmd switches supported
- More object oriented

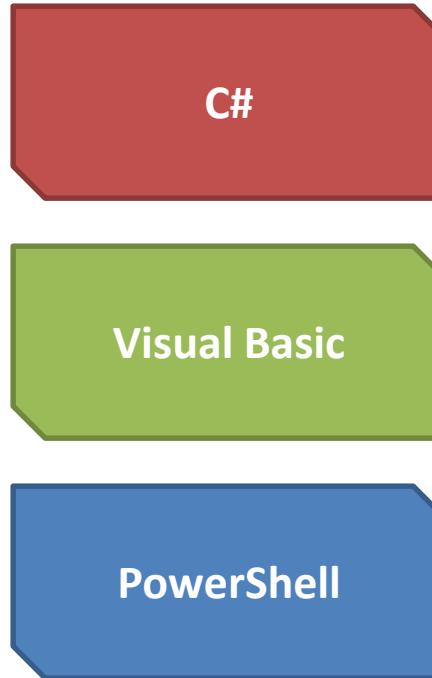
Demo - SQLPS and sqlcmd



SMO



Requires .NET framework 2.0



What does it give us?

- Robust, object aware feature set
 - Flexibility
 - Durability
- Increased complexity, but also increased re-usability



Demos – Using the SMO

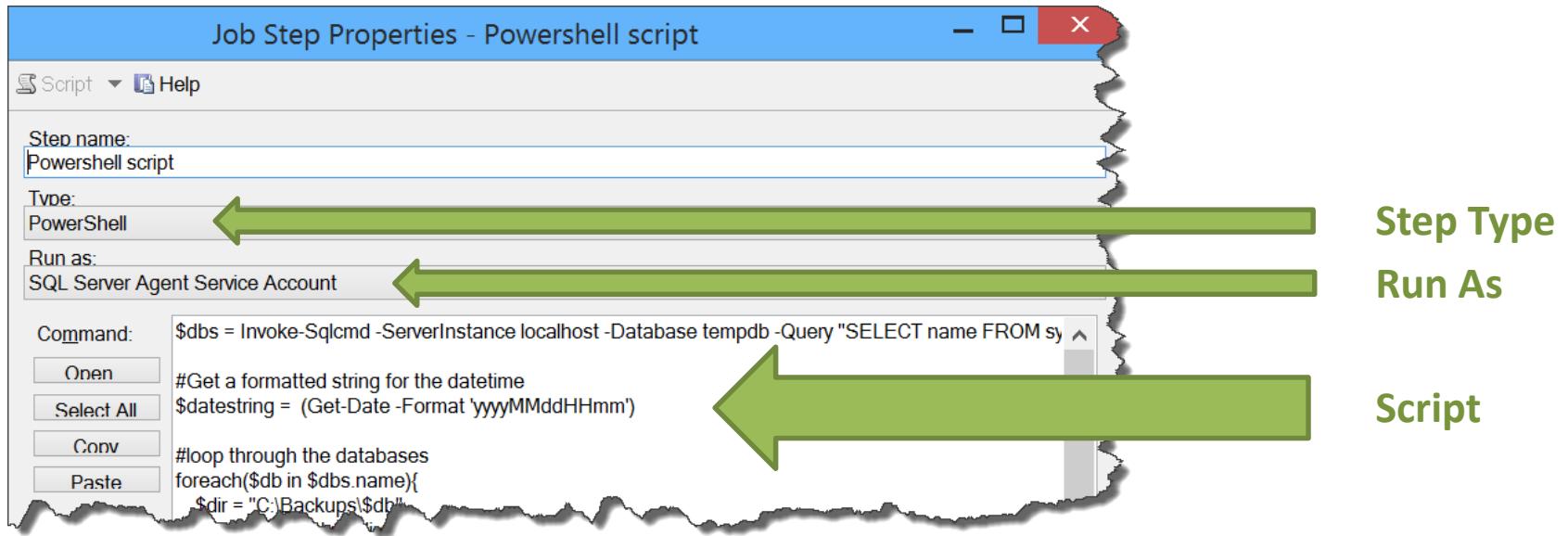


Powershell in SQL Agent Jobs



**Use SQL Agent to mange jobs
Easily integrate Powershell with your tasks**

What does it look like?



Uses SQL Server's “mini” shell
Powershell 2.0 (booooo)

Demo – Powershell and SQL Agent



Review

Once you've loaded the SQLPS, how would you start working with the provider interactively in Powershell?

cd SQLSERVER:

By default, what security account will a SQL Server Agent job run a Powershell script?

SQL Server Agent service account

Can you use Invoke-SqlCmd interactively?

No

To create a new SMO server object, what cmdlet will you use?

New-Object

Practical Use



Break!



Script re-use



Rinse and Repeat

Writing Scripts

Re-using Scripts

Your Powershell Profile

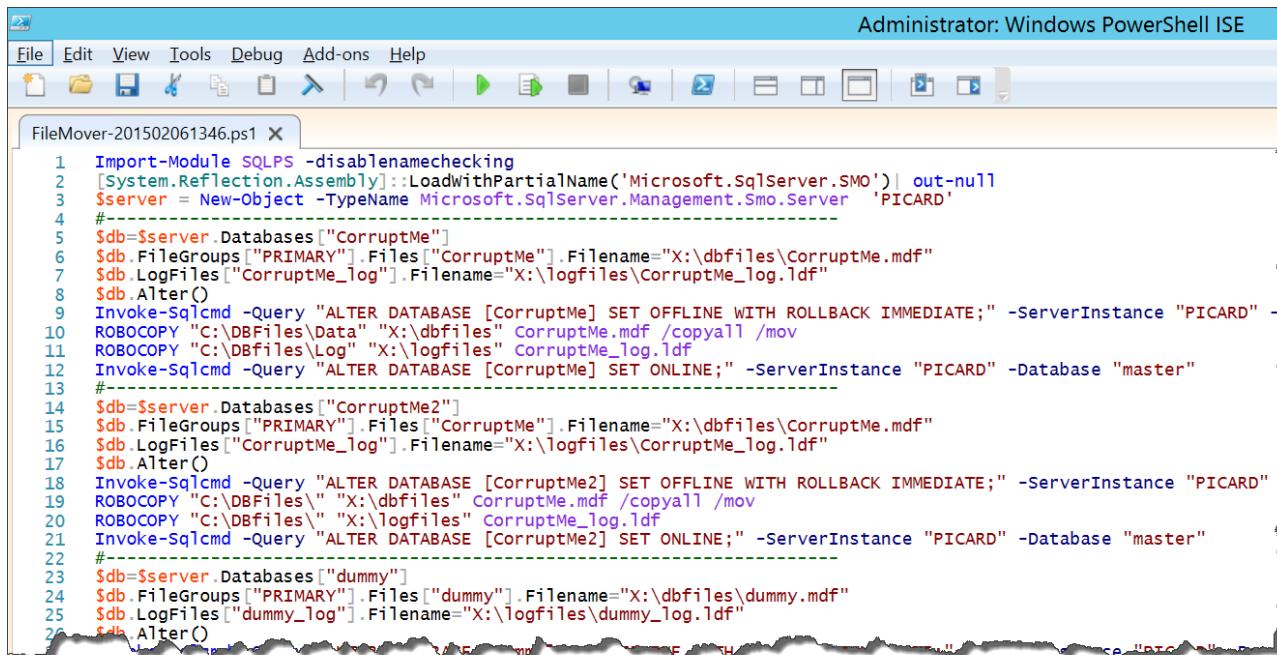
Functions and Modules

Best Practices and Documenting your scripts

Saving Scripts

Scripts are saved as .ps1 files

Can be run by calling them at the command line



```
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
FileMover-201502061346.ps1 ×
1 Import-Module SQLPS -disallowNameChecking
2 [System.Reflection.Assembly]::LoadWithPartialName('Microsoft.SqlServer.SMO') | Out-Null
3 $server = New-Object -TypeName Microsoft.SqlServer.Management.Smo.Server 'PICARD'
4 #
5 $db=$server.Databases["CorruptMe"]
6 $db.FileGroups["PRIMARY"].Files["CorruptMe"].Filename="X:\dbfiles\CorruptMe.mdf"
7 $db.LogFiles["CorruptMe_Log"].Filename="X:\logfiles\CorruptMe_Log.ldf"
8 $db.Alter()
9 Invoke-Sqlcmd -Query "ALTER DATABASE [CorruptMe] SET OFFLINE WITH ROLLBACK IMMEDIATE;" -ServerInstance "PICARD" -P
10 ROBOCOPY "C:\DBfiles\Data" "X:\dbfiles" CorruptMe.mdf /copyall /mov
11 ROBOCOPY "C:\DBfiles\Log" "X:\logfiles" CorruptMe_Log.ldf
12 Invoke-Sqlcmd -Query "ALTER DATABASE [CorruptMe] SET ONLINE;" -ServerInstance "PICARD" -Database "master"
13 #
14 $db=$server.Databases["CorruptMe2"]
15 $db.FileGroups["PRIMARY"].Files["CorruptMe"].Filename="X:\dbfiles\CorruptMe.mdf"
16 $db.LogFiles["CorruptMe_Log"].Filename="X:\logfiles\CorruptMe_Log.ldf"
17 $db.Alter()
18 Invoke-Sqlcmd -Query "ALTER DATABASE [CorruptMe2] SET OFFLINE WITH ROLLBACK IMMEDIATE;" -ServerInstance "PICARD" -P
19 ROBOCOPY "C:\DBfiles\" "X:\dbfiles" CorruptMe.mdf /copyall /mov
20 ROBOCOPY "C:\DBfiles\" "X:\logfiles" CorruptMe_Log.ldf
21 Invoke-Sqlcmd -Query "ALTER DATABASE [CorruptMe2] SET ONLINE;" -ServerInstance "PICARD" -Database "master"
22 #
23 $db=$server.Databases["dummy"]
24 $db.FileGroups["PRIMARY"].Files["dummy"].Filename="X:\dbfiles\dummy.mdf"
25 $db.LogFiles["dummy_Log"].Filename="X:\logfiles\dummy_Log.ldf"
26 $db.Alter()
```

Code Re-use: Scripts

Powershell Scripts can be parameterized

```
param ($VariableName)
```

Parameters are referenced

- Positionally
- Variable name becomes parameter name

Code Re-Use: Your Profile

Script file that loads whenever you start
Different profiles

- For you in the current host
- For you in all hosts
- For everyone in the current host
- For everyone in all hosts

`$Profile + $PSHome`

`Get-Help about_Profiles`

Code Re-use: Functions

You can package code into functions

```
function Function-Name { }
```

Functions can be parameterized like scripts

They can also be called within a script

Get-Help about_Functions

Code Re-use: Module

You can make your own modules to use

No special syntax, just save as a .psm1

Allows us to use custom functions direct from
the command line.

Install to either:

C:\Users\<user>\Documents\WindowsPowerShell

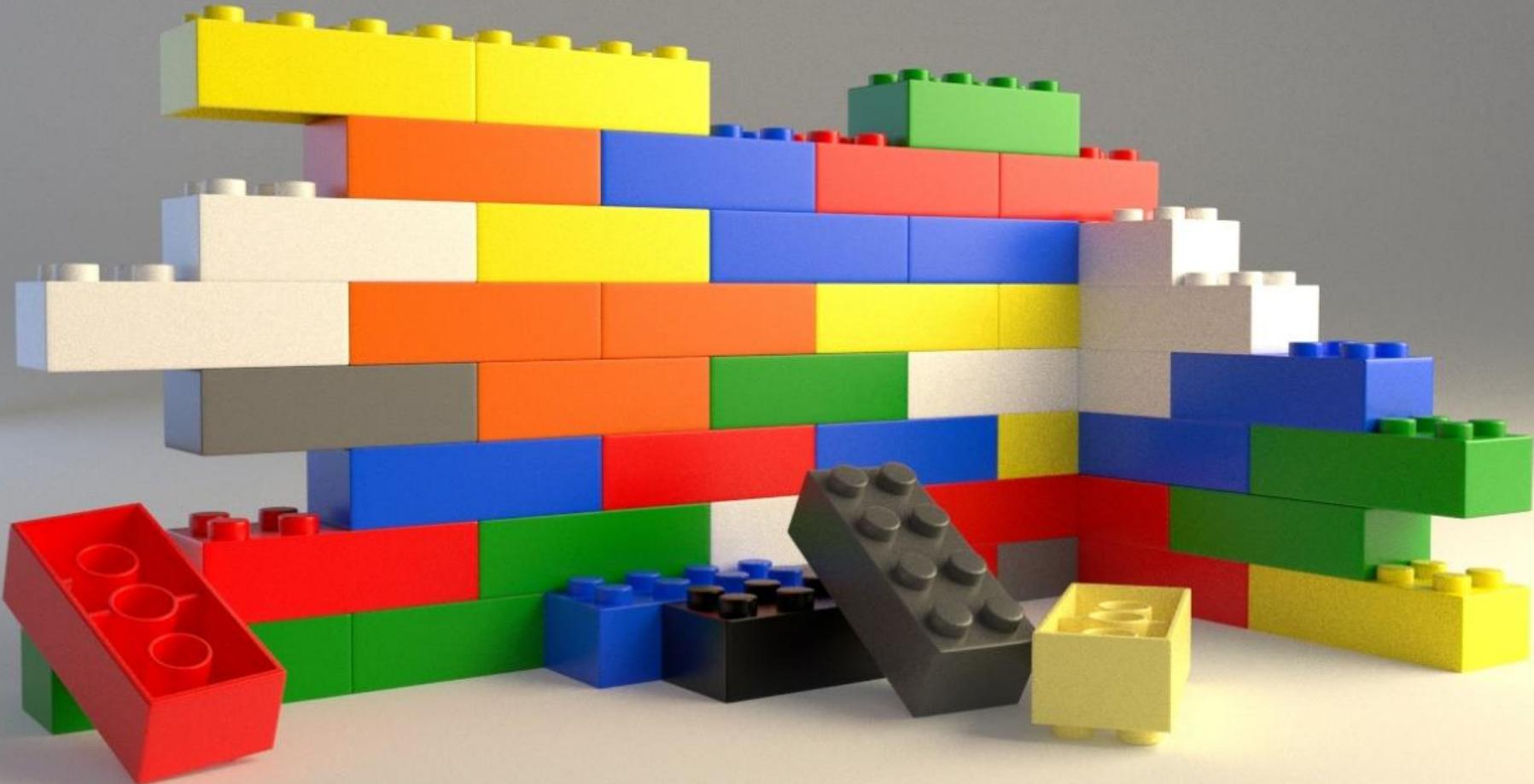
C:\Program Files\WindowsPowerShell

Get-Help about_Modules

Demo – Functions, Modules, Profile



Best Practices



Comment Based Help

<#

.SYNOPSIS

By using a specific formation, you can write your own Get-Help information

#>

Get-Help about_comment_based_help

Review

What would you use to load a standard set of Powershell modules every time you opened a session?

Your Powershell profile

What command should you use to load a custom module?

Import-Module

Where does the **param** keyword go when parameterizing a function or a script?

It must go at the beginning (can be after a comment block)

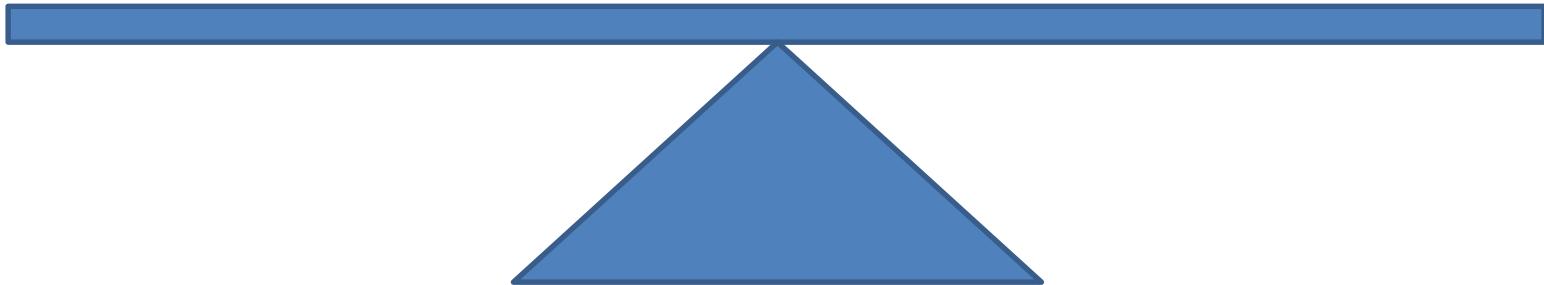
Break!



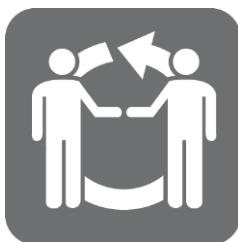
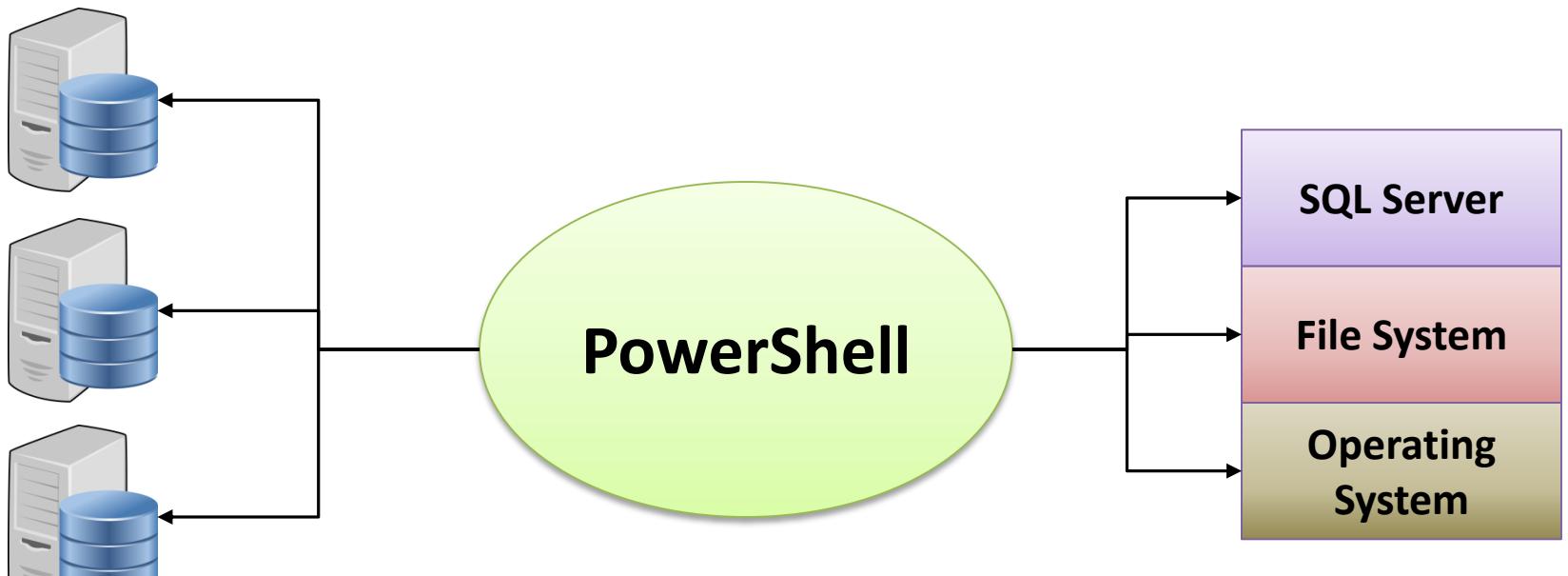
Practical Use



The Approach



What to Consider



Demo – Practical Scripts



Powershell and Server Core



What Server Core IS



Windows Server...
...with no GUI!

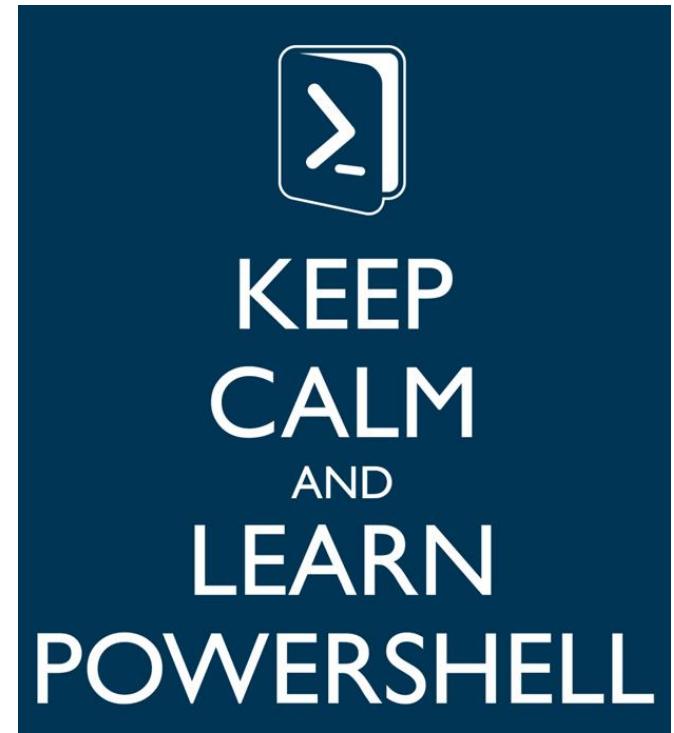
Lean – Smaller footprint and install

Stable – Fewer applications and services

Secure – Less security loopholes

Tools to Start With

SCONFIG



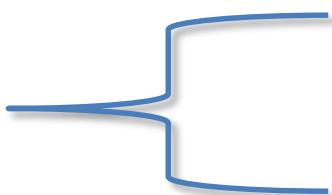
First Look



Server Core Support

Feature	Supported
Database Engine Services	Yes
SQL Server Replication	Yes
Full Text Search	Yes
Analysis Services	Yes
Reporting Services	No
SQL Server Data Tools (SSDT)	No
Client Tools Connectivity	Yes
Integration Services Server[1]	Yes
Client Tools Backward Compatibility	No
Client Tools SDK	No
SQL Server Books Online	No
Management Tools - Basic	Remote Only[2]
Management Tools – Complete	Remote Only[2]
Distributed Replay Controller	No
Distributed Replay Client	Remote Only[2]
SQL Client Connectivity SDK	No
Microsoft Sync Framework	Yes[3]
Master Data Services	No
Data Quality Services	No

SQL Server Requirements



Requirement	How to install
.NET Framework 2.0 SP2	Not included in Windows Server 2012, must be installed.
.NET Framework 3.5 SP1 Full Profile	Not included in Windows Server 2012, must be installed.
.NET Framework 4 Server Core Profile	Shipped with Windows Server 2012.
Windows Installer 4.5	Shipped with Windows Server 2012
Windows PowerShell 2.0	Powershell 3.0 shipped with Windows Server 2012.

Unattended Install

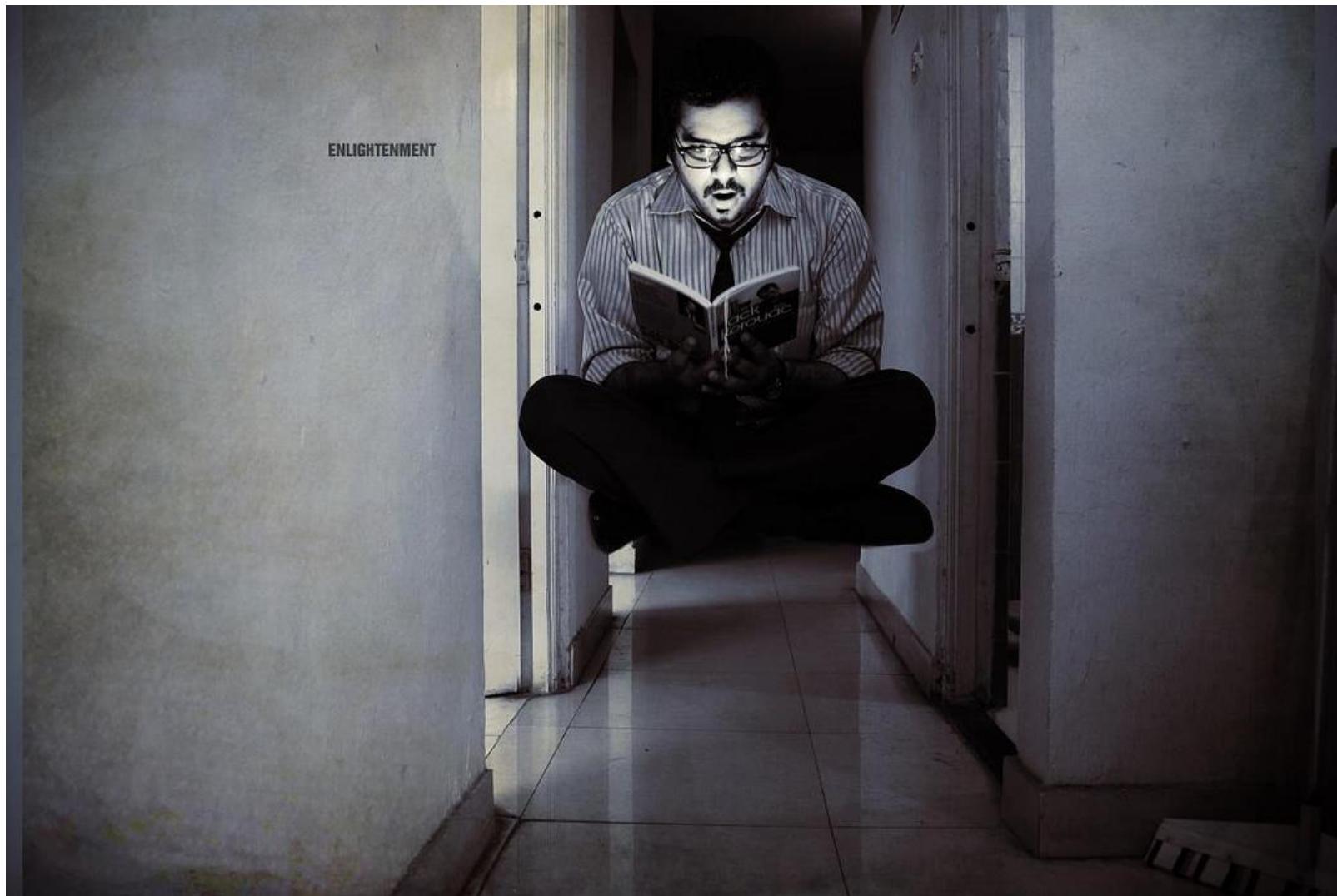
Well established tool

Allows for consistent SQL builds

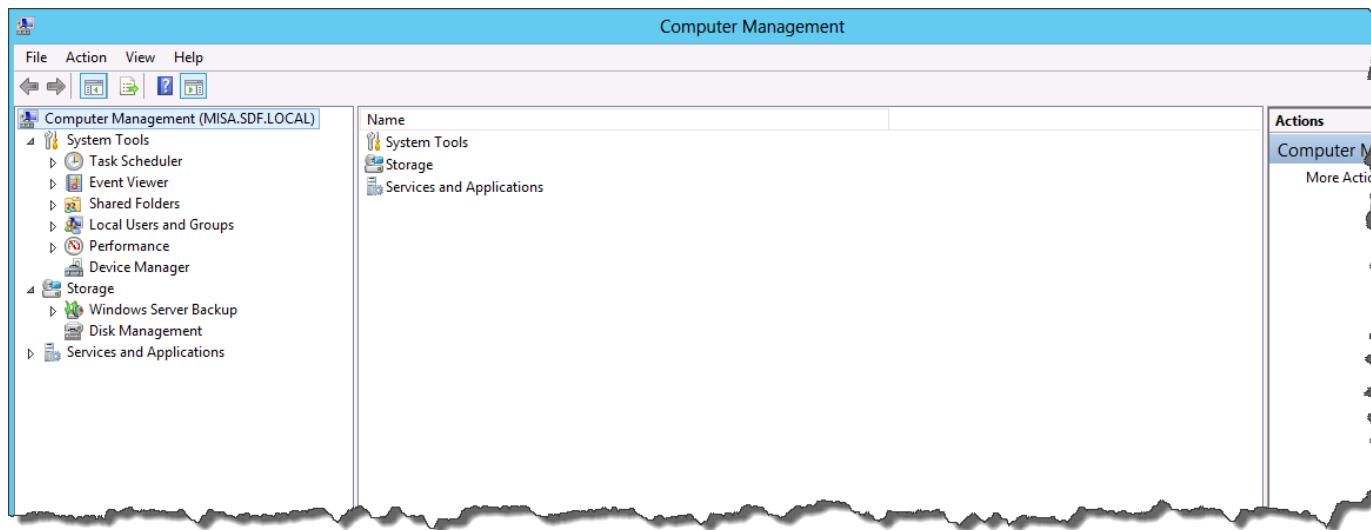
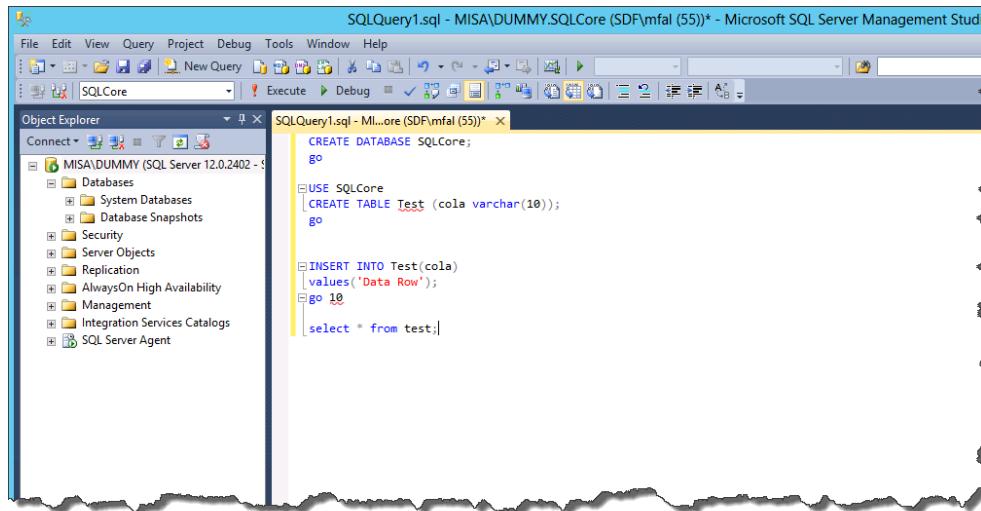
Infrastructure as code



Demo - SQL Install



Now What?



Moment of Zen



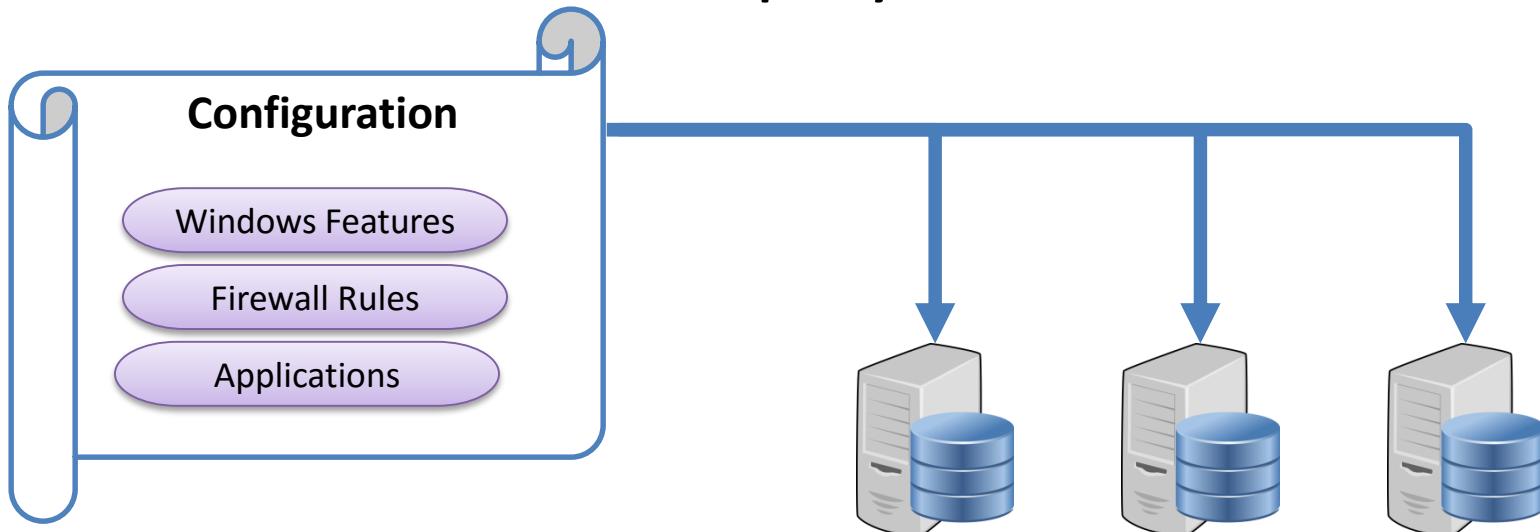
The Tools – Powershell DSC

“Killer Feature” in Powershell v4.0

Declarative Configurations

Resources Define What and How

Create. Deploy. Execute.



Demo - Install using DSC



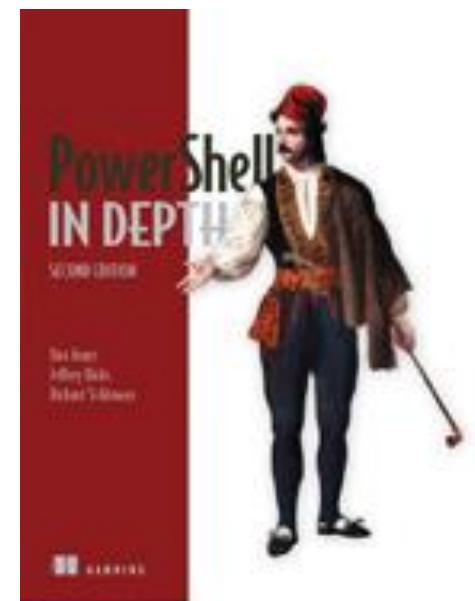
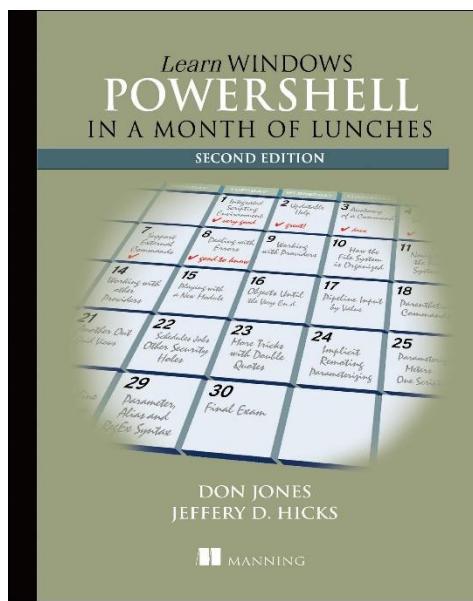
WWW.MNPHOTOBLOG.COM

So now what?



Books

- [Powershell In A Month of Lunches](#)
- [Powershell in Depth, 2nd Edition](#)



General Powershell

- The Scripting Guys
(<http://blogs.technet.com/b/heyscriptingguy/>)
- Jeff Hicks
(<http://jdhitsolutions.com/blog/>)
- Powershell.org
(<http://powershell.org/>)

Bloggers

- [Ben Miller](http://www.dbaduck.com/)
(<http://www.dbaduck.com/>)
- [Allen White](http://sqlblog.com/blogs/allen_white/)
([http://sqlblog.com/blogs/allen white/](http://sqlblog.com/blogs/allen_white/))
- [Kendal Van Dyke](http://www.kendalvandyke.com/)
(<http://www.kendalvandyke.com/>)
- [Laerte Junior](https://www.simple-talk.com/author/laerte-junior/)
(<https://www.simple-talk.com/author/laerte-junior/>)
- [Aaron Nelson](http://sqlvariant.com/)
(<http://sqlvariant.com/>)

Online

[Microsoft Virtual Academy](#)

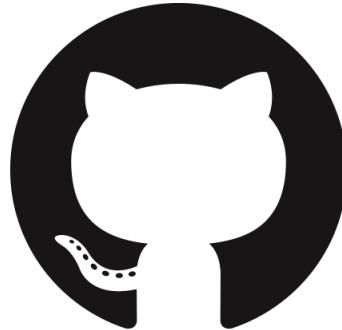
[PluralSight \(\\$\\$\)](#)



Other scripts

<https://github.com/MikeFal>

- Powershell repository (all my scripts, including WIP)
- Intro To Powershell repository (all the scripts from THIS class)



Most of all, USE IT

Find tasks to automate

Manage the file system ONLY through
Powershell

Rewrite a T-SQL or other task using
Powershell

And so on...

Questions



mike@mikefal.net



www.mikefal.net



[@Mike_Fal](https://twitter.com/Mike_Fal)