

---

# EXPLORING THE SQL POWERSHELL MODULE



[www.mikefal.net](http://www.mikefal.net)



Mike Fal



**Microsoft**  
**CERTIFIED**  
Solutions Expert

Data Platform

# Get-Agenda

What is SqlServer (SQLPS)?

The Provider

Cmdlets/Functions

Practical Use

# Don't Panic!

```
    $trns = Get-ChildItem $dir -recurse | Where-Object {$_.name -like "*.trn"} | sort-object LastWriteTime
}
else{
    $full = Get-ChildItem $dir | Where-Object {$_.name -like "*.bak"} | Sort-Object LastWriteTime -desc | Select-Object -first 1
    $diff = Get-ChildItem $dir | Where-Object {$_.name -like "*.dff"} | sort-object LastWriteTime -desc | select-object -first 1
    $trns = Get-ChildItem $dir | Where-Object {$_.name -like "*.trn"} | sort-object LastWriteTime
}

#initialize and process full backup
$outputfile = Join-Path -Path $outputdir -ChildPath "restore_$database.sql"
$restore = Get-RestoreObject $database $full
$shfull = Get-Header $restore $smosrv
if($database.Length -eq 0)
{
    $database = $shfull.DatabaseName
    $restore.Database=$database
}

$LSNCheck = $shfull.CheckpointLSN
$files = $restore.ReadFileList($smosrv)
foreach($file in $files){
    $pfile = $file.PhysicalName
    if($newdata.Length -gt 0 -and $file.Type -eq "D"){
        $pfile=$newdata + $pfile.Substring($pfile.LastIndexOf("\"))
    }

    if($newdata.Length -gt 0 -and $file.Type -eq "L"){
        $pfile=$newlog + $pfile.Substring($pfile.LastIndexOf("\"))
    }

    $newfile = New-Object("Microsoft.SqlServer.Management.Smo.RelocateFile") ($file.LogicalName,$pfile)
    $restore.RelocateFiles.Add($newfile) | out-null
}

$ssqlout += "/******"
$ssqlout += "Restore Database Script Generated $(Get-Date)"
$ssqlout += "Database: "+$database
$ssqlout += "*****/"
$ssqlout += "--FULL RESTORE"
If($owner){$ssqlout += "EXECUTE AS LOGIN = '$owner';"}
$ssqlout += $restore.Script($smosrv)

#process differential backups
if($diff -ne $null){
    $restore = Get-RestoreObject $database $diff
    $hdiff = Get-Header $restore $smosrv

    if($hdiff.DatabaseBackupLSN -eq $LSNCheck){
        $ssqlout += "--DIFF RESTORE"
        $ssqlout += $restore.Script($smosrv)
        $LSNCheck = $hdiff.LastLSN
    }
    else{
        . . . . .
    }
}
```

**Don't focus on the code,  
focus on the concepts.**

**Ask questions!**



In the beginning...



# The Basics

## SQLPS

- Powershell module for SQL Server
- Includes cmdlets and the provider
- SQL Server 2012 or greater
- Powershell 2.0 or greater

`Import-Module SQLPS`





## SqlServer

- Introduced with SSMS July 2016 update!
- Includes everything from SQLPS along with fixes
- Not compatible with SQLPS!

`Import-Module SqlServer`



# Installation

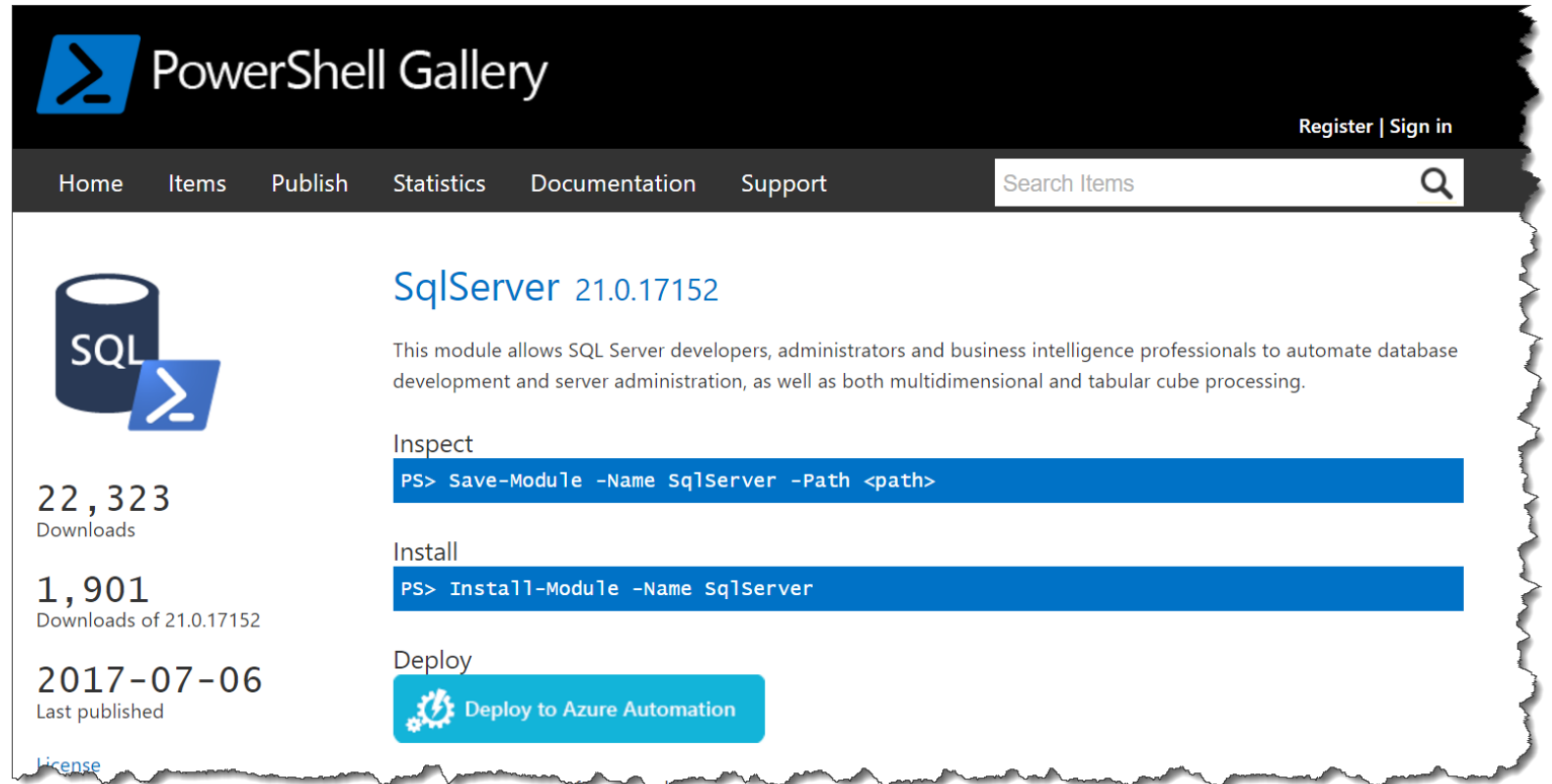
- Not installed specifically previous to SQL 2016 SP1
- Included anytime you install ANY SQL Server component
- SQLPS Installation Path:  
C:\Program Files (x86)  
  \Microsoft SQL Server  
  \**130**\Tools\PowerShell\Modules
- SqlServer Installation Path  
C:\Program Files\WindowsPowerShell\Modules



# Installation with SSMS 17/SQL 2016

PowerShell Gallery (<https://www.powershellgallery.com/>)

Install-Module SqlServer  
(PowerShell 5.0+)



The screenshot shows the PowerShell Gallery website. The header includes the PowerShell Gallery logo and navigation links: Home, Items, Publish, Statistics, Documentation, Support, and a search bar. The main content area displays the 'SqlServer 21.0.17152' module. On the left, there is a download statistics section showing 22,323 total downloads, 1,901 downloads for version 21.0.17152, and a last published date of 2017-07-06. The right side of the module page includes a description, an 'Inspect' section with a PowerShell command, an 'Install' section with a PowerShell command, and a 'Deploy' section with a button to 'Deploy to Azure Automation'.

**PowerShell Gallery**

Register | Sign in

Home Items Publish Statistics Documentation Support Search Items

**SqlServer 21.0.17152**

This module allows SQL Server developers, administrators and business intelligence professionals to automate database development and server administration, as well as both multidimensional and tabular cube processing.

**Inspect**

```
PS> Save-Module -Name SqlServer -Path <path>
```

**Install**

```
PS> Install-Module -Name SqlServer
```

**Deploy**

Deploy to Azure Automation

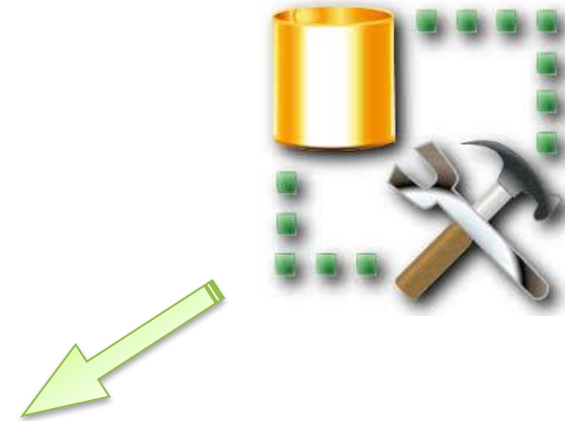
22,323 Downloads

1,901 Downloads of 21.0.17152

2017-07-06 Last published

[License](#)

# Built on the SMO



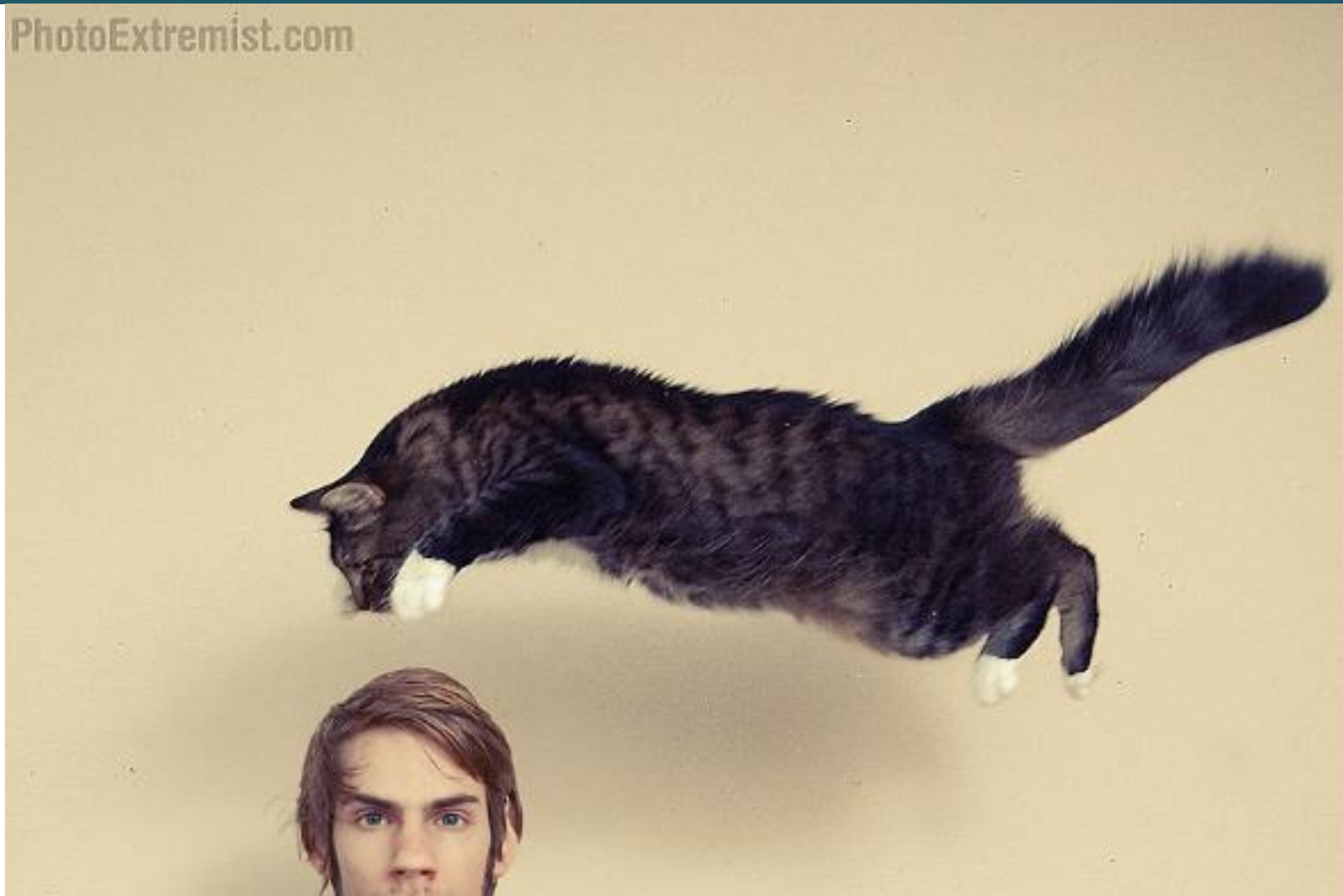
**.Net Library**  
*requires .Net 2.0*

# What Makes It Work?

SMO ASSEMBLIES	
Microsoft.SqlServer.Management.Common	Microsoft.SqlServer.SqlEnum
Microsoft.SqlServer.Smo	Microsoft.SqlServer.RegSvrEnum
Microsoft.SqlServer.Dmf	Microsoft.SqlServer.WmiEnum
Microsoft.SqlServer.Instapi	Microsoft.SqlServer.ServiceBrokerEnum
Microsoft.SqlServer.SqlWmiManagement	Microsoft.SqlServer.ConnectionInfoExtended
Microsoft.SqlServer.ConnectionInfo	Microsoft.SqlServer.Management.Collector
Microsoft.SqlServer.SmoExtended	Microsoft.SqlServer.Management.CollectorEnum
Microsoft.SqlServer.SqlTDiagM	Microsoft.SqlServer.Management.Dac
Microsoft.SqlServer.SString	Microsoft.SqlServer.Management.DacEnum
Microsoft.SqlServer.Management.RegisteredServers	Microsoft.SqlServer.Management.Utility
Microsoft.SqlServer.Management.Sdk.Sfc	

DEMO!

PhotoExtremist.com





# SQL Server and the File System





# 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

Windows PowerShell

```
PS SQLSERVER:\> dir
```

Name	Root	Description
----	----	-----
DAC	SQLSERVER:\DAC	SQL Server Data-Tier Application Component
DataCollection	SQLSERVER:\DataCollection	SQL Server Data Collection
SQLPolicy	SQLSERVER:\SQLPolicy	SQL Server Policy Management
Utility	SQLSERVER:\Utility	SQL Server Utility
SQLRegistration	SQLSERVER:\SQLRegistration	SQL Server Registrations
SQL	SQLSERVER:\SQL	SQL Server Database Engine
SSIS	SQLSERVER:\SSIS	SQL Server Integration Services
XEvent	SQLSERVER:\XEvent	SQL Server Extended Events
DatabaseXEvent	SQLSERVER:\DatabaseXEvent	SQL Server Extended Events
SQLAS	SQLSERVER:\SQLAS	SQL Server Analysis Services

```
PS SQLSERVER:\sql\localhost\default
```



Provider

Folder

Host/Server

Instance

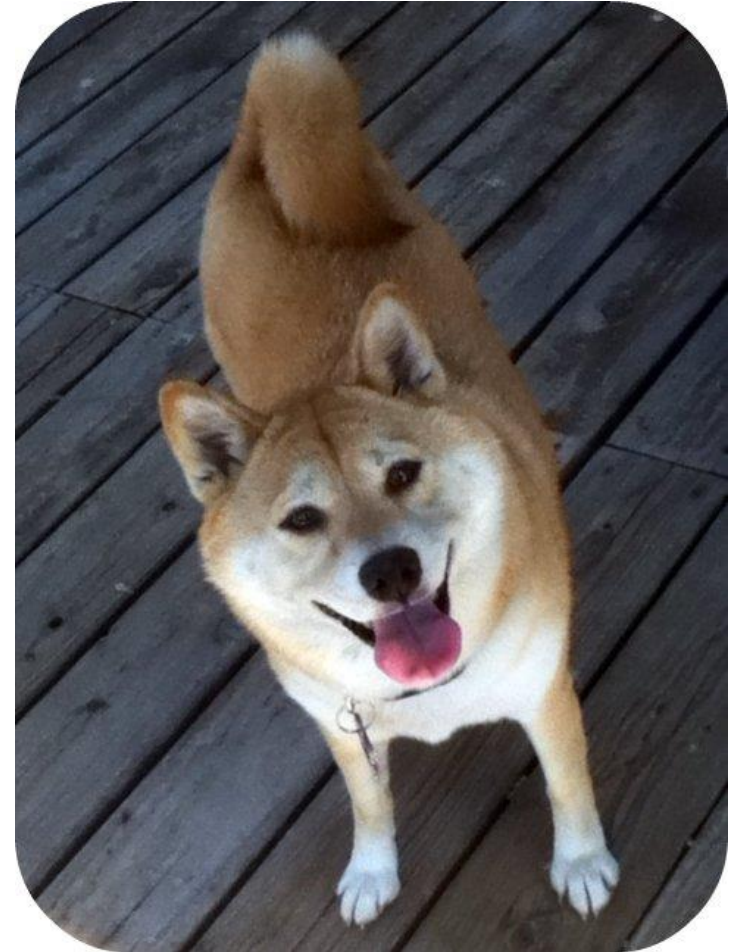
DEMO!



**Timeouts!**

**Tab completion and Intellisense**

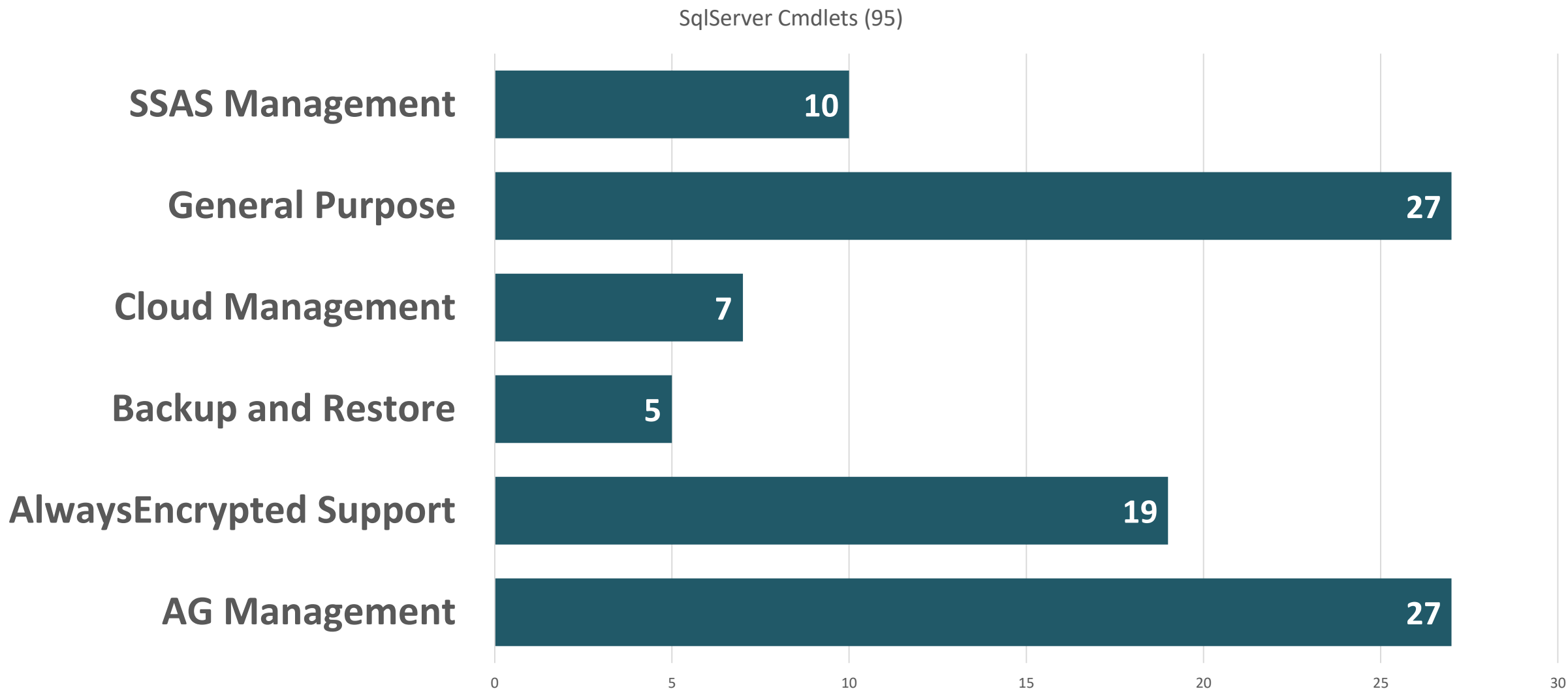
**Contextual behavior**



Verb-Noun



# Cmdlet Overview



# Cmdlets: General Purpose

Add-RoleMember	Get-SqlLogin
Add-SqlLogin	Invoke-PolicyEvaluation
ConvertFrom-EncodedSqlName/Decode-SqlName	Invoke-Sqlcmd
ConvertTo-EncodedSqlName/Encode-SqlName	New-SqlCredential
Convert-UrnToPath	Read-SqlTableData
Get-SqlAgent	Read-SqlViewData
Get-SqlAgentJob	Remove-RoleMember
Get-SqlAgentJobHistory	Remove-SqlCredential
Get-SqlAgentJobSchedule	Remove-SqlLogin
Get-SqlAgentJobStep	Save-SqlMigrationReport
Get-SqlAgentSchedule	Set-SqlCredential
Get-SqlCredential	Set-SqlErrorLog
Get-SqlDatabase	Write-SqlTableData
Get-SqlErrorLog	

# Cmdlets: Backup and Restore

Backup-SqlDatabase

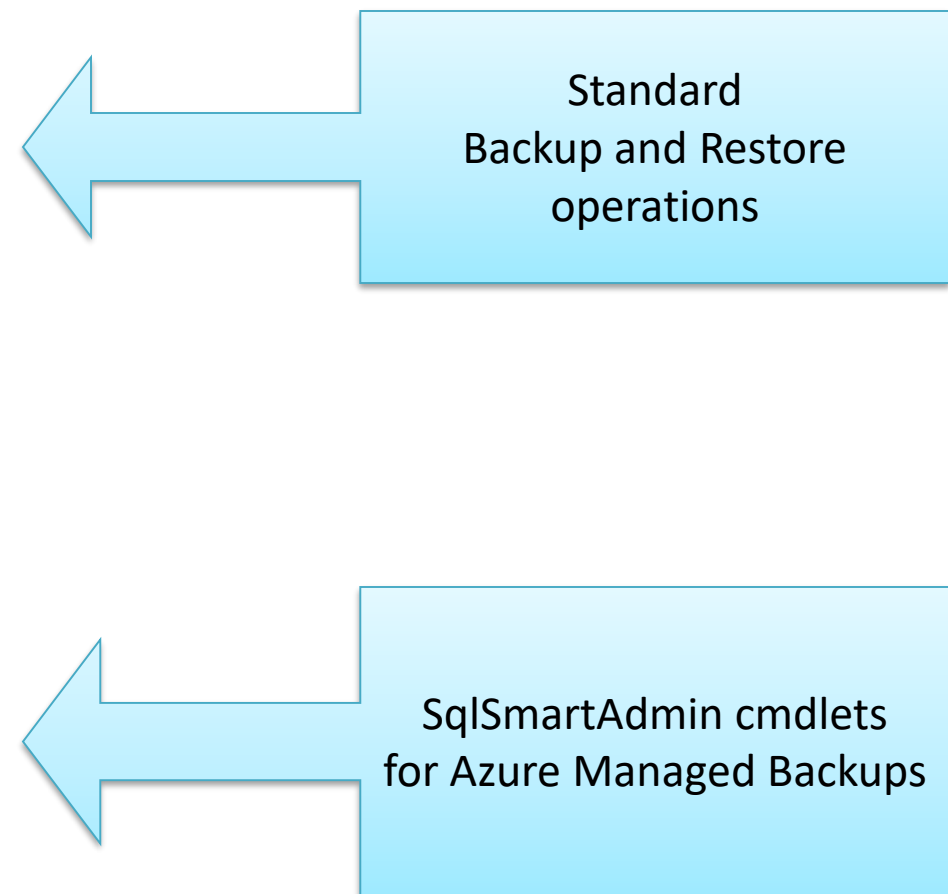
Restore-SqlDatabase

New-SqlBackupEncryptionOption

Get-SqlSmartAdmin

Set-SqlSmartAdmin

Test-SqlSmartAdmin



Standard  
Backup and Restore  
operations

SqlSmartAdmin cmdlets  
for Azure Managed Backups

# Cloud Management

Add-SqlFirewallRule

Remove-SqlFirewallRule

Get-SqlInstance

Set-SqlAuthenticationMode

Set-SqlNetworkConfiguration

Start-SqlInstance

Stop-SqlInstance

These cmdlets all require the  
Cloud Adapter Service

Careful, this doesn't return  
what you think it does...

These do the same as Start-  
Service and Stop-Service

# Cmdlets: Advanced Cmdlets

- Availability Group Management
- AlwaysEncrypted Management
- SSAS Management (merged from SQLAS)





# DEMO!

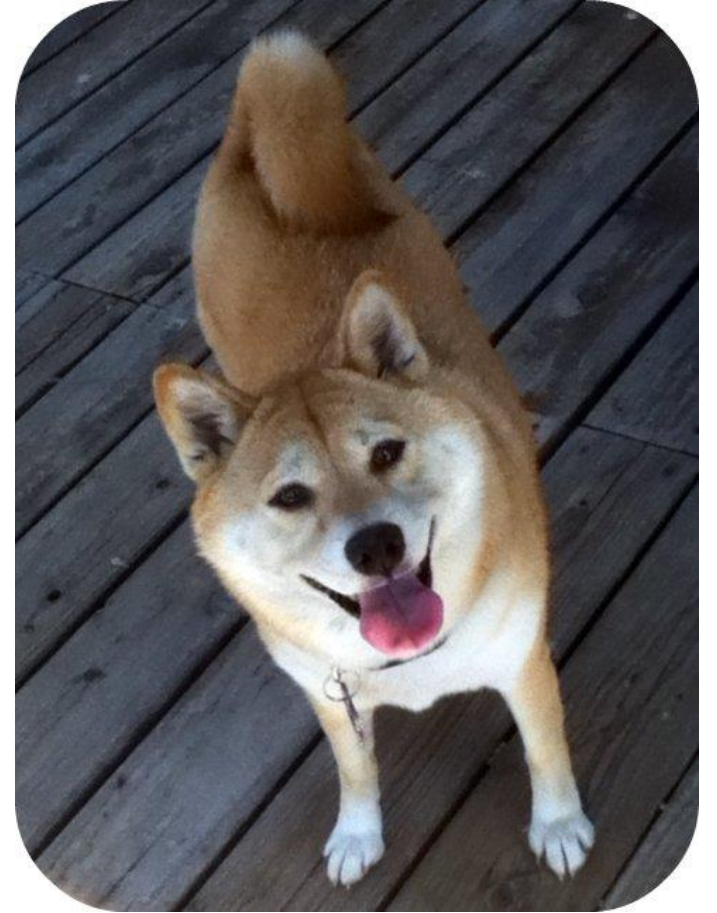
*Mike Fal - [www.mikefal.net](http://www.mikefal.net)*



# Gotchas

**Poorly Documented  
(*Changing!*)**

**Unexpected behavior**





So now what?

A close-up photograph of a concrete wall with the words "WHAT'S NEXT?" painted in large, bold, black, hand-painted capital letters. The paint is thick and has a slightly irregular, brush-stroke texture. The wall is light-colored and shows some signs of wear and discoloration. The background is out of focus, showing some dark, leafy vegetation.

WHAT'S  
NEXT?



DEMO!



# Wrapping up!

- ✓ What is the SqlServer/SQLPS Module?
- ✓ How do we use it?
- ✓ The Provider, what is that all about?
- ✓ What kind of cmdlets (functions) are there?
- ✓ How do I use this darn thing, anyway?





# Questions



[mike@mikefal.net](mailto:mike@mikefal.net)



[www.mikefal.net](http://www.mikefal.net)



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



<https://github.com/MikeFal>