



# PowerShell with SharePoint Server and Office 365

# Shane Young

- 13 Year SharePoint MVP
- [Shane@powerapps911.com](mailto:Shane@powerapps911.com)
- @ShanesCows
- <http://www.youtube.com/ShaneYoungCloud>
- Cincinnati, Ohio
- <https://www.PowerApps911.com>
- <https://www.BoldZebras.com>



**SPTechCon**  
The SharePoint and  
Office 365 Conference

# Todd Klindt

12 Year SharePoint MVP

Writer, speaker, consultant, podcaster, SPDocKit marketer



[todd@toddklindt.com](mailto:todd@toddklindt.com)

@toddklindt

[www.toddklindt.com](http://www.toddklindt.com)



# Agenda

- Official Cmdlets
- Developerly Option
- The Goods

- Slides at <https://www.toddklindt.com/sptechcon2018Boston>



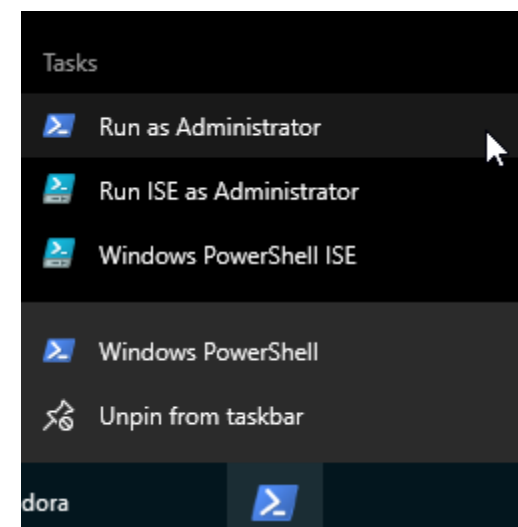
# Official Cmdlets

# There are 4 things to install

- Microsoft Official Office 365 PowerShell cmdlets
- [Install Sign-in Assistant](#) – 64bit
- [Install MSOnline Module](#) (v1) – GA
- [Install Azure AD Module](#) (v2) (Release or Preview)
- [Install SharePoint Online Module](#)
- [Install Skype for Business Online Module](#)
  
- [Connect to all Office 365 Services](#)

# Before you connect

- Have to be able to Run PowerShell as an Administrator
- Have to be an Office 365 Global Administrator
  - Except Exchange
- Should be running PowerShell 3.0 or later
  - `$PSVersionTable.PSVersion`
- Recommend 5.1 on your Windows desktop
  - Also consider adding [PSReadLine](#) if you are not on Win10
  - [Video walkthrough](#)
- Execution policy needs to be RemoteSigned



# Tangent: Talk about Passwords

- You will need your O365 username and password a lot so you have good and bad options:
- Annoying but secure

```
$MyAccount = Get-Credential
```

- Less annoying and way, way less secure

```
$username = admin@company.onmicrosoft.com
```

```
$password = "RightHereInPlainText"
```

```
$secure = $password | ConvertTo-SecureString -AsPlainText -Force
```

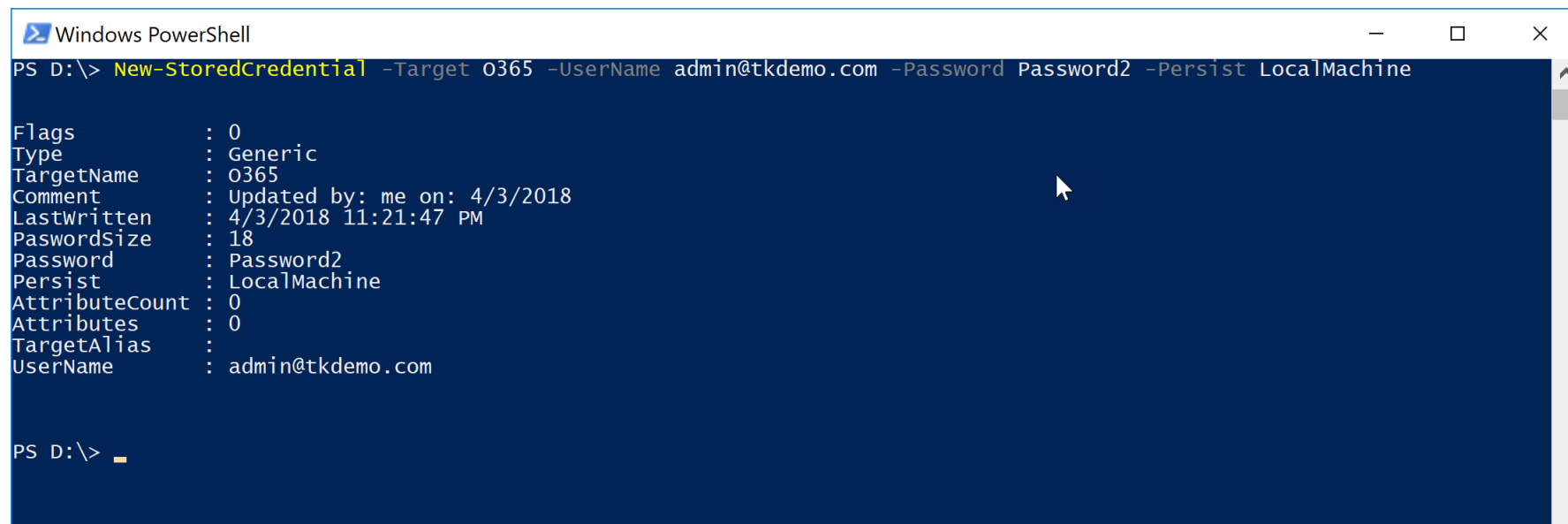
```
$MyAccount = New-Object System.Management.Automation.PSCredential ($username, $secure)
```

- [Use an encrypted file](#)



# Credential Manager

- [Use Credential Manager](#)
- `Install-Module credentialmanager -Scope CurrentUser`
- `New-StoredCredential -Target 0365 -UserName admin@tkdemo.com -Password Password2 -Persist LocalMachine`



```
Windows PowerShell
PS D:\> New-StoredCredential -Target 0365 -UserName admin@tkdemo.com -Password Password2 -Persist LocalMachine

Flags           : 0
Type            : Generic
TargetName      : 0365
Comment        : Updated by: me on: 4/3/2018
LastWritten     : 4/3/2018 11:21:47 PM
PasswordSize    : 18
Password       : Password2
Persist        : LocalMachine
AttributeCount  : 0
Attributes      : 0
TargetAlias     :
UserName       : admin@tkdemo.com

PS D:\>
```

# Connect to your Azure AD Tenant

- **MSOnline (v1)**

```
$MyAccount = Get-Credential  
Connect-MsolService -Credential $MyAccount  
Get-MsolUser  
Get-Command -Module msonline
```

- **AzureAD (v2)**

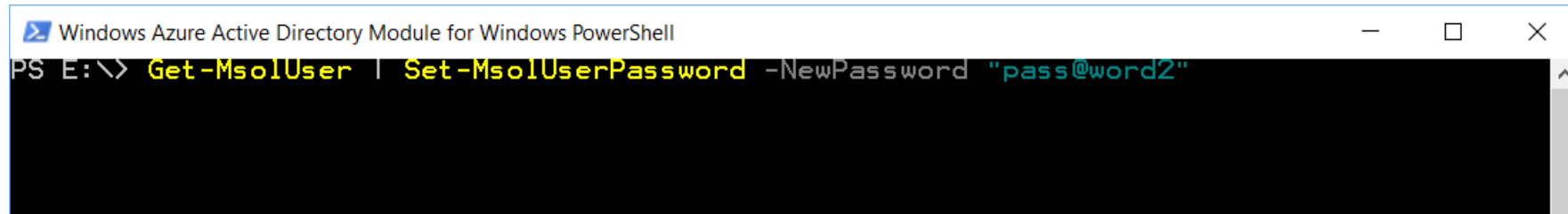
```
$MyAccount = Get-Credential  
Connect-AzureAD -Credential $MyAccount  
Get-AzureADUser  
Get-Command -Module AzureAD
```

- **Install-Module azuread**

# Fun Gotchas

```
Windows Azure Active Directory Module for Windows PowerShell
PS E:\> help Set-MsolUserPassword
NAME
    Set-MsolUserPassword
SYNOPSIS
    Resets the password for a user.
SYNTAX
    Set-MsolUserPassword -ObjectId <Guid> [-ForceChangePassword <Boolean>] [-NewPassword <string>] [-TenantId <Guid>] [<CommonParameters>]
    Set-MsolUserPassword -UserPrincipalName <string> [-ForceChangePassword <Boolean>] [-NewPassword <string>] [-TenantId <Guid>] [<CommonParameters>]
DESCRIPTION
    The Set-MsolUserPassword cmdlet is used to change the password of a user. This cmdlet can only be used for users with standard identities.
RELATED LINKS
REMARKS
    To see the examples, type: "get-help Set-MsolUserPassword -examples".
    For more information, type: "get-help Set-MsolUserPassword -detailed".
    For technical information, type: "get-help Set-MsolUserPassword -full".
    For online help, type: "get-help Set-MsolUserPassword -online"
PS E:\> help Set-MsolUserPassword -Examples
NAME
    Set-MsolUserPassword
SYNOPSIS
    Resets the password for a user.
----- EXAMPLE 1 -----
C:\PS>Set-MsolUserPassword -UserPrincipalName user@contoso.com
Returns the user's new password.
Description
-----
This command resets the password for user@contoso.com. A random password will be generated. The user will be required to reset the password on next sign in.
----- EXAMPLE 2 -----
C:\PS>Set-MsolUserPassword -userPrincipalName user@consoso.com -NewPassword "pa$$word"
Returns the user's new password.
Description
-----
This command resets the password for user@contoso.com. The user will be required to
-- More --
```

# Don't Try This At Home



```
Windows Azure Active Directory Module for Windows PowerShell
PS E:\> Get-MsolUser | Set-MsolUserPassword -NewPassword "pass@word2"
```

## Connect to Skype for Business

```
$Skype = New-CsOnlineSession -Credential $MyAccount
```

```
Import-PSSession $Skype
```

```
Get-CsOnlineUser
```

```
Remove-PSSession $Skype
```

- This one can be confusing. Remember that Skype for Business, Lync, and Communication Server are all the same thing. The cmdlets and documentation tend to use them interchangeably. ☹

# Connect to Exchange

```
$Exchange = New-PSSession -ConfigurationName Microsoft.Exchange -  
ConnectionUri "https://outlook.office365.com/powershell-liveid/" -  
Credential $MyAccount -Authentication "Basic" -AllowRedirection  
  
Import-PSSession $Exchange  
  
Get-Mailbox  
  
Remove-PSSession $Exchange
```

- Skype and Exchange are limited to 3 sessions so always end your session.

- Just a little different

- No cmdlets, uses Remoting
- Limited to three sessions
- Requires port 80
- Close out gracefully
  - Remove-PSSession \$Session

- Supports MFA

```
Administrator: SharePoint Online Management Shell
PS C:\> Get-ExecutionPolicy
Restricted
PS C:\> Set-ExecutionPolicy RemoteSigned

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about_Execution_Policies help topic at http://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): y
PS C:\> $UserCredential = Get-Credential

cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:
Credential
PS C:\> $Session = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri https://outlook.office365.com/powerShell-liveid/ -Credential $UserCredential -Authentication Basic -AllowRedirection
PS C:\> Import-PSSession $Session
WARNING: The names of some imported commands from the module 'tmp_klzre0r4.jaj' include unapproved verbs that might make them less discoverable. To find the commands with unapproved verbs, run the Import-Module command again with the Verbose parameter. For a list of approved verbs, type Get-Verb.

ModuleType Version Name ExportedCommands
-----
Script 1.0 tmp_klzre0r4.jaj {Add-AvailabilityAddressSpace, Add-DistributionGroupMember...

PS C:\> Get-Mailbox

Name Alias ServerName ProhibitSendQuota
----
DiscoverySearchMailbox... DiscoverySearchMa... cy1pr18mb0155 50 GB (53,687,091,200 bytes)
todd todd sn2pr18mb0783 49.5 GB (53,150,220,288 bytes)

PS C:\>
```

Administrator: SharePoint Online Management Shell

```
PS C:\> Get-Mailbox todd | select *
```

```
RunspaceId           : 318562d5-0727-417a-8ac2-50d2e3de2806
Database             : NAMPR18DG049-db034
MailboxProvisioningConstraint :
MailboxRegion        :
MessageCopyForSentAsEnabled : False
MessageCopyForSendOnBehalfEnabled : False
MailboxProvisioningPreferences : {}
UseDatabaseRetentionDefaults : False
RetainDeletedItemsUntilBackup : False
DeliverToMailboxAndForward : False
IsExcludedFromServingHierarchy : False
IsHierarchyReady     : True
IsHierarchySyncEnabled : True
HasSnackyAppData     : False
LitigationHoldEnabled : False
SingleItemRecoveryEnabled : True
RetentionHoldEnabled : False
EndDateForRetentionHold :
StartDateForRetentionHold :
RetentionComment     :
RetentionUrl         :
LitigationHoldDate  :
LitigationHoldOwner  :
LitigationHoldDuration : Unlimited
ManagedFolderMailboxPolicy :
RetentionPolicy      : Default MRM Policy
AddressBookPolicy    :
CalendarRepairDisabled : False
ExchangeGuid        : 4c0b99fc-3938-4c91-94e7-072af3e7f2ea
MailboxContainerGuid :
UnifiedMailbox       :
MailboxLocations     : {1;4c0b99fc-3938-4c91-94e7-072af3e7f2ea;Primary;namprd18.prod.outlook.com;1d4140fa-da2c-4e16-bdde-ae0fe708ddb4}
AggregatedMailboxGuids : {}
```



```
Administrator: SharePoint Online Management Shell
PS C:\> New-Mailbox -Alias jill -Name jill -FirstName Jill -LastName Klindt -DisplayName "Jill Klindt" -MicrosoftOnlineS
ervicesID jill@tkclass.onmicrosoft.com -Password (ConvertTo-SecureString -String 'P@ssw0rd' -AsPlainText -Force) -ResetP
asswordOnNextLogon $false
WARNING: After you create a new mailbox, you must go to the Office 365 Admin Center and assign the mailbox a license,
or it will be disabled after the grace period.

Name                Alias                ServerName           ProhibitSendQuota
----                -
jill                 jill                 cy1pr18mb0664       49.5 GB (53,150,220,288 bytes)
WARNING: Failed to replicate mailbox:'jill' to the site:'namprd18.prod.outlook.com/Configuration/Sites/CY1PR18'. The
mailbox will be available for logon in approximately 15 minutes.

PS C:\> Get-Mailbox

Name                Alias                ServerName           ProhibitSendQuota
----                -
DiscoverySearchMailbox... DiscoverySearchMa... cy1pr18mb0155       50 GB (53,687,091,200 bytes)
jill                 jill                 cy1pr18mb0664       49.5 GB (53,150,220,288 bytes)
todd                 todd                 sn2pr18mb0783       49.5 GB (53,150,220,288 bytes)

PS C:\> Write-Host "Ta da!"
Ta da!
PS C:\>
```

New-Mailbox  
DisplayName  
jill@tkclass.onmicrosoft.com  
'P@ssw0rd'

# License Up That New Mailbox

```
Set-MsolUser -UserPrincipalName jill@tkclass.onmicrosoft.com -UsageLocation
```

The image shows a PowerShell terminal window and the Office Admin Center interface. The PowerShell window displays the command `Set-MsolUser -UserPrincipalName jill@tkclass.onmicrosoft.com -UsageLocation "US"`. The Office Admin Center interface shows the 'Active users' page with a table of users.

	Display name ^	User name	Status
<input type="checkbox"/>	Jill Klindt	jill@tkclass.onmicrosoft.com	Office 365 Business Premium
<input type="checkbox"/>	Todd Klindt	todd@tkclass.onmicrosoft.com	Office 365 Business Premium

# PowerShell with SharePoint Online

- Be prepared for disappointment
- Allows basic manipulation of SharePoint Online
  - Users and groups
  - Tenants
  - Site Collections
- [Download here](#)



Useful SharePoint things with all of that

<This Slide Intentionally Left Blank>

# Connect to SharePoint online

```
Connect-SPOService -URL https://Tenant-admin.sharepoint.com  
    -Credential $MyAccount  
Get-SPOSite  
Get-Command -Module Microsoft.Online.SharePoint.PowerShell
```



# Example Script

```
Param(  
    [Parameter(Mandatory=$true)]  
    [ValidateNotNullOrEmpty()]  
    [string] $User  
)  
  
# Add the Active Directory bits and not complain if they're already there  
Import-Module ActiveDirectory -ErrorAction SilentlyContinue
```

```
# Add the Azure Active Directory module

Import-Module MSOnline

# Import-Module AzureAD

# Define AD group that is synced to AAD and is used for ODFB audience

$syncgroupname = "CloudSync"

$syncgroup = Get-ADGroup $syncgroupname
```



```
# Name of the Azure License to apply
```

```
$license = "tkclass:O365_BUSINESS_PREMIUM"
```

```
# Get-AzureADSubscribedSku
```

```
# Azure AD domain suffix
```

```
$aadsuffix = "tkclass.org"
```

```
# $aadsuffix = (Get-AzureADDomain | Where-Object -Property IsDefault -Value  
$true -EQ).name
```

```
# First, add the user to the group
```

```
Add-ADGroupMember -Identity $syncgroupname -Members $User
```

```
# Remind them to recompile their SharePoint audience
```

```
Write-Host "You'll need to recompile your SharePoint audience to reflect the  
group change"
```

```
# Sync up to Azure AD
```

```
Start-ADSyncSyncCycle
```

```
# Now tweak the user in Azure AD
```

```
# First connect
```

```
Connect-MsolService
```

```
# Connect-AzureAD
```

```
# Get the user
```

```
$aaduser = "$user@$aadsuffix"
```

```
# Set the user's location. Without that the license will fail
```

```
Set-MsolUser -UserPrincipalName $aaduser -UsageLocation "SI"
```

```
# Set-AzureADUser
```

```
# Set the user's license
```

```
Set-MsolUserLicense -UserPrincipalName $aaduser -AddLicenses $license
```

```
# Set-AzureADUserLicense
```

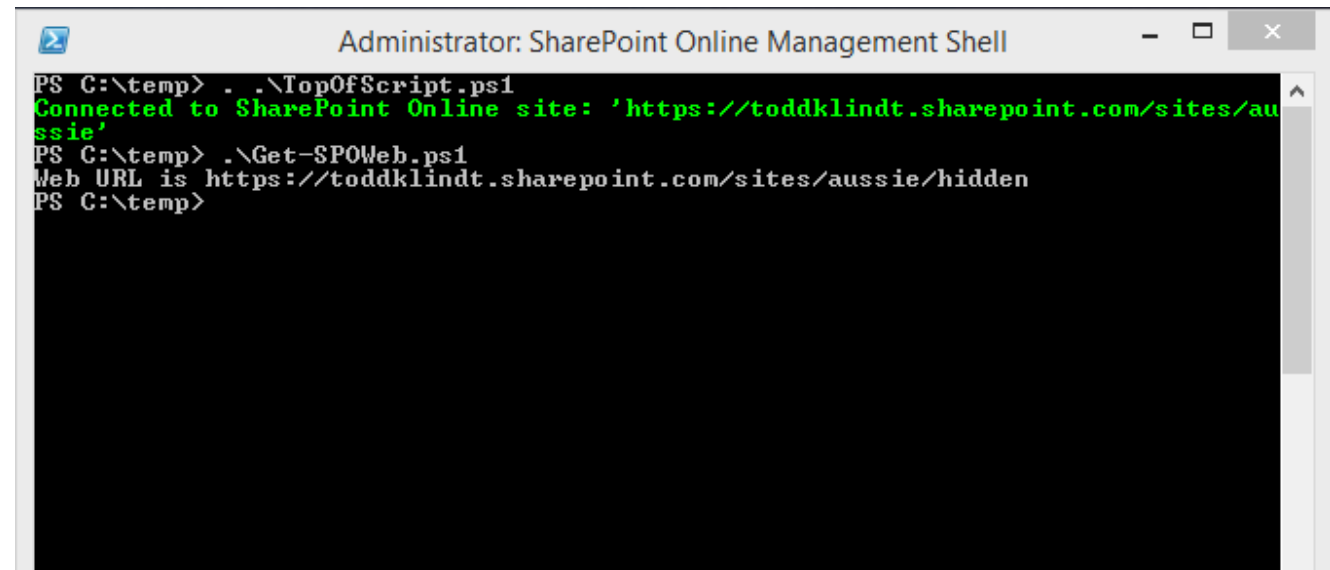
# Teams, Flow, and PowerApps

- Teams
  - For automating all those Teams Admin Tasks
  - Install-Module MicrosoftTeams
  - [Read all about it](#)
- Flow and PowerApps
  - For both creators and Admins
  - Get list of all Flows and PowerApps
  - Kind of a [janky install](#)



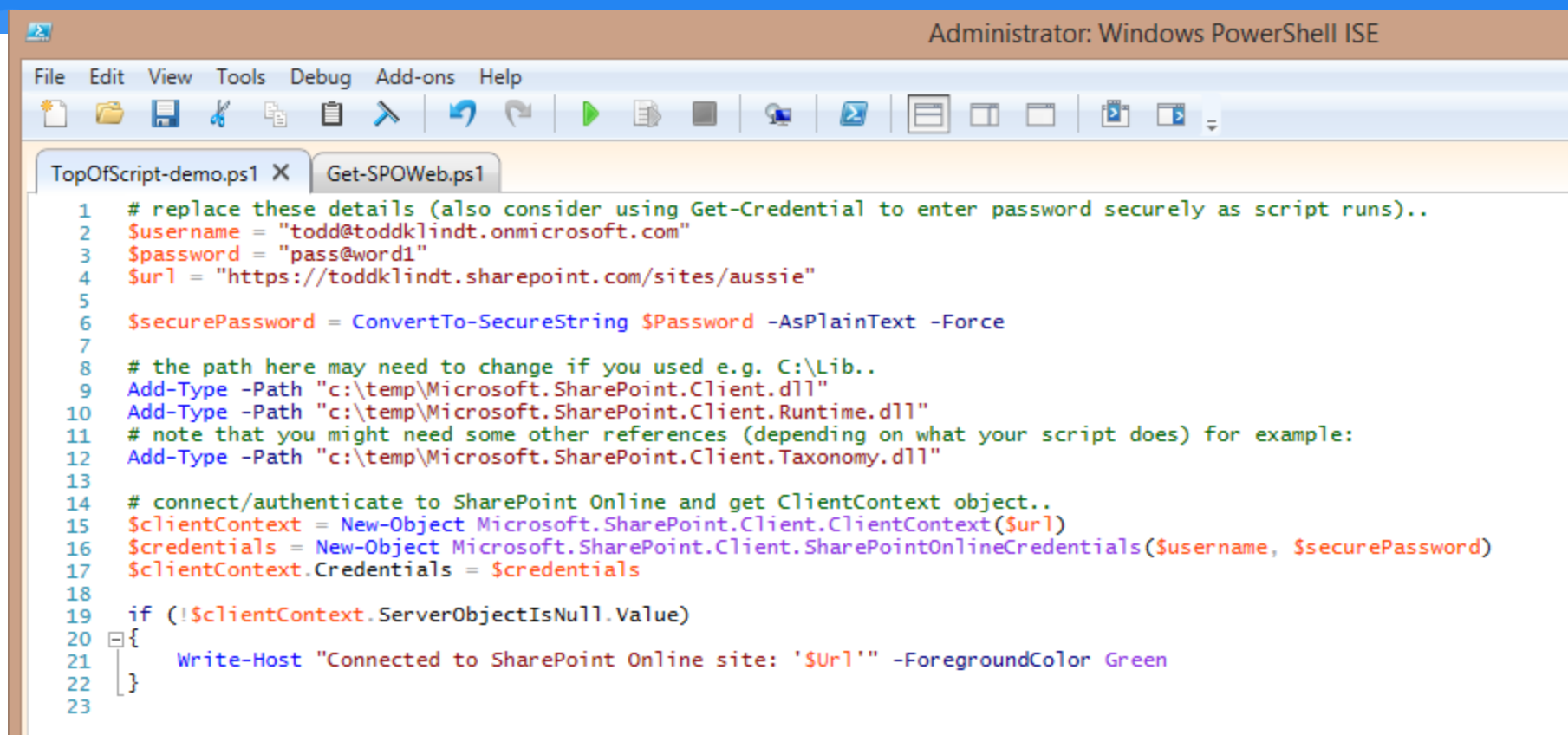
# Alternative #1

- Can use the Client Side Object Model with PowerShell to do more
- Developery, be afraid
- Copy DLLs from server
- Or download [SharePoint 2016 Client SDK](#)



```
Administrator: SharePoint Online Management Shell
PS C:\temp> . .\TopOfScript.ps1
Connected to SharePoint Online site: 'https://toddklindt.sharepoint.com/sites/aussie'
PS C:\temp> .\Get-SPOWeb.ps1
Web URL is https://toddklindt.sharepoint.com/sites/aussie/hidden
PS C:\temp>
```

# Top Of Script



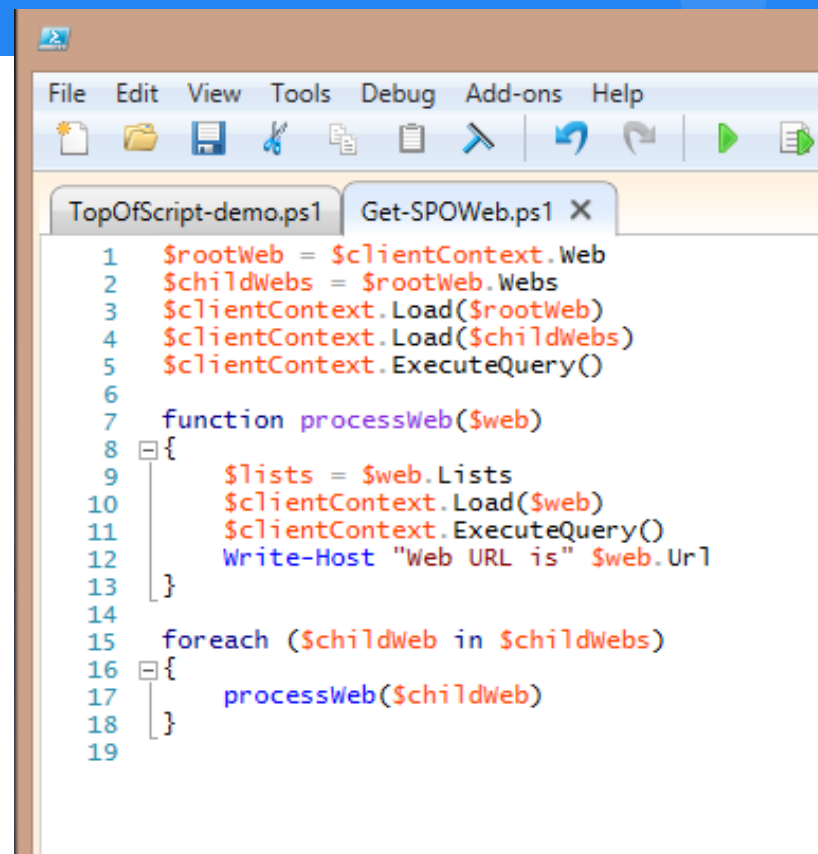
The image shows a screenshot of the Windows PowerShell ISE (Integrated Scripting Environment) running as Administrator. The window title is "Administrator: Windows PowerShell ISE". The menu bar includes File, Edit, View, Tools, Debug, Add-ons, and Help. The toolbar contains various icons for file operations and execution. Two tabs are open: "TopOfScript-demo.ps1" and "Get-SPOWeb.ps1". The active tab shows a PowerShell script with the following content:

```
1 # replace these details (also consider using Get-Credential to enter password securely as script runs)..
2 $username = "todd@toddklindt.onmicrosoft.com"
3 $password = "pass@word1"
4 $url = "https://toddklindt.sharepoint.com/sites/aussie"
5
6 $securePassword = ConvertTo-SecureString $Password -AsPlainText -Force
7
8 # the path here may need to change if you used e.g. C:\Lib..
9 Add-Type -Path "c:\temp\Microsoft.SharePoint.Client.dll"
10 Add-Type -Path "c:\temp\Microsoft.SharePoint.Client.Runtime.dll"
11 # note that you might need some other references (depending on what your script does) for example:
12 Add-Type -Path "c:\temp\Microsoft.SharePoint.Client.Taxonomy.dll"
13
14 # connect/authenticate to SharePoint Online and get ClientContext object..
15 $clientContext = New-Object Microsoft.SharePoint.Client.ClientContext($url)
16 $credentials = New-Object Microsoft.SharePoint.Client.SharePointOnlineCredentials($username, $securePassword)
17 $clientContext.Credentials = $credentials
18
19 if (!$clientContext.ServerObjectIsNull.Value)
20 {
21     Write-Host "Connected to SharePoint Online site: '$url'" -ForegroundColor Green
22 }
23
```



# Get-SPOweb

- Examples from:
- <http://www.sharepointnutsandbolts.com/2013/12/Using-CSOM-in-PowerShell-scripts-with-Office365.html>



```
File Edit View Tools Debug Add-ons Help
TopOfScript-demo.ps1 Get-SPOWeb.ps1 X
1 $rootWeb = $clientContext.Web
2 $childWebs = $rootWeb.Webs
3 $clientContext.Load($rootWeb)
4 $clientContext.Load($childWebs)
5 $clientContext.ExecuteQuery()
6
7 function processWeb($web)
8 {
9     $lists = $web.Lists
10    $clientContext.Load($web)
11    $clientContext.ExecuteQuery()
12    Write-Host "Web URL is" $web.Url
13 }
14
15 foreach ($childWeb in $childWebs)
16 {
17     processWeb($childWeb)
18 }
19
```

# More Examples

```
TopOfScript-OnPrem.ps1* X
1 # replace these details
2 # Can also use Read-Host or technique in http://www.toddklindt.com/PoshSecurePasswords
3 $username = "todd@toddklindt.com"
4 $password = "pass@word1"
5 $url = "http://upgrade.toddklindt.com/blog"
6
7 $securePassword = ConvertTo-SecureString $Password -AsPlainText -Force
8 # $securePassword = Read-Host "Enter password for $username" -AsSecureString
9
10 # the path here may need to change if you used e.g. C:\Lib..
11 Add-Type -Path "c:\temp\Microsoft.SharePoint.Client.dll"
12 Add-Type -Path "c:\temp\Microsoft.SharePoint.Client.Runtime.dll"
13 # note that you might need some other references (depending on what your script does) for example:
14 Add-Type -Path "c:\temp\Microsoft.SharePoint.Client.Taxonomy.dll"
15
16 # connect/authenticate to SharePoint Online and get ClientContext object..
17 $clientContext = New-Object Microsoft.SharePoint.Client.ClientContext($url)
18 # $credentials = New-Object Microsoft.SharePoint.Client.SharePointOnlineCredentials($username, $securePassword)
19 $credentials = New-Object System.Management.Automation.PSCredential ($username, $securePassword)
20 $clientContext.Credentials = $credentials
21
22 if (!$clientContext.ServerObjectIsNull.Value)
23 {
24     Write-Host "Connected to SharePoint Online site: '$url'" -ForegroundColor Green
25 }
--
```

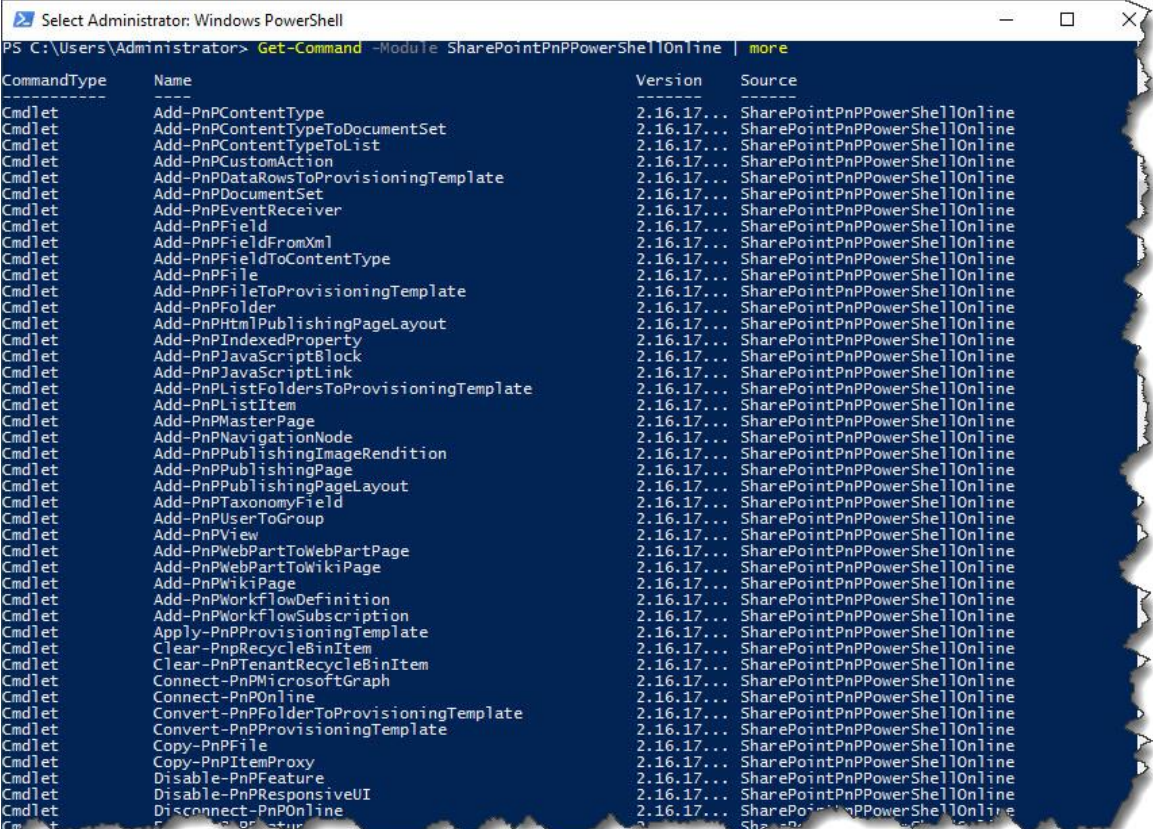
```
Get-SPOList.ps1 X
1 $rootWeb = $clientContext.Web
2 $childWebs = $rootWeb.Webs
3 $clientContext.Load($rootWeb)
4 $clientContext.Load($childWebs)
5 $clientContext.ExecuteQuery()
6
7 function getLists($web)
8 {
9     $lists = $web.Lists
10     # $clientContext.Load($web)
11     $clientContext.Load($lists)
12     $clientContext.ExecuteQuery()
13     $lists | select title
14 }
15
16 getLists($rootWeb)
17
18
```



# Alternative #2

# Patterns and Practices PowerShell (Phew!)

- More scary developer stuff
- Hidden in [Github](#)
  - <https://github.com/SharePoint/PnP-PowerShell>
- Adds 250 more cmdlets
- Install-Module SharePointPnPPowerShellOnline
- Get-Command -Module SharePointPnPPowerShellOnline
- Works with all the SharePoints
- Scoped at Site Collection



```
Select Administrator: Windows PowerShell
PS C:\Users\Administrator> Get-Command -Module SharePointPnPPowerShellOnline | more

CommandType      Name                                     Version      Source
-----
Cmdlet            Add-PnPContentType                     2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPContentTypeToDocumentSet        2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPContentTypeToList               2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPCustomAction                   2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPDataRowsToProvisioningTemplate  2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPDocumentSet                    2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPEventReceiver                  2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPField                           2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPFieldFromXml                   2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPFieldToContentType              2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPFile                             2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPFileToProvisioningTemplate      2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPFolder                          2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPHTMLPublishingPageLayout       2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPIndexedProperty                 2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPJavaScriptBlock                 2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPJavaScriptLink                  2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPListFoldersToProvisioningTemp... 2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPListItem                        2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPMasterPage                      2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPNavigationNode                  2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPPublishingImageRendition        2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPPublishingPage                  2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPPublishingPageLayout           2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPTaxonomyField                   2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPUserToGroup                     2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPView                             2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPWebPartToWebPartPage            2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPWebPartToWikiPage               2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPWikiPage                         2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPWorkflowDefinition               2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Add-PnPWorkflowSubscription             2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Apply-PnPProvisioningTemplate           2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Clear-PnPRecycleBinItem                 2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Clear-PnPTenantRecycleBinItem           2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Connect-PnPMicrosoftGraph              2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Connect-PnPOnline                       2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Convert-PnPFolderToProvisioningTemp...  2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Convert-PnPProvisioningTemplate         2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Copy-PnPFile                             2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Copy-PnPItemProxy                       2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Disable-PnPFeature                      2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Disable-PnPResponsiveUI                 2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            Disconnect-PnPOnline                    2.16.17...  SharePointPnPPowerShellOnline
Cmdlet            ...
```

# Favorites

- PnPFile
  - Add-PnPFile , Copy-PnPFile, Find-PnPFile, Get-PnPFile, Move-PnPFile, Remove-PnPFile, Rename-PnPFile, Set-PnPFileCheckedIn, Set-PnPFileCheckedOut
- PnPList
  - Add, Get, Set, Remove
- Get-PnPListItem
- Set-PnPGroupPermissions
- Add-PnPView
- Get-PnPField
- Provisioning
  - Get-Command -Module SharePointPnPPowerShellOnline -Noun "\*Provisioning\*"

# But my Boss HATES PnP PowerShell!

- Your boss is misinformed 😊
- [Vesa Juvonen](#), Senior Program Manager from SharePoint Engineering and MCM for SharePoint, is one of the main project owners
- Is scanned the same as any [PowerShell Gallery](#) Module (not at all)
- Erwin van Hunen works at [RenCore](#)
  - Exceptions are approved by SharePoint Engineering team
- Signed with Microsoft's key starting November 2017
- Uses the same API as web parts and other SharePoint code
- It's Open Source
- Respects SharePoint security
  - Can be more secure, as it can be more fine grained
- PnP PowerShell hits the Office 365 API a billion times a month



# Demo Time

# Another

```
TKC Admin - □ ×
Sign in to PnP Office 365 ×
Windows PowerShell - □ ×
PS C:\> help Get-PnPUnifiedGroupOwners -Examples
NAME
    Get-PnPUnifiedGroupOwners
SYNOPSIS
    * Supported in: SharePoint Online.
    Gets owners of a particular Office 365 Group (aka Unified Group)
    -----EXAMPLE 1-----
    PS:> Get-PnPUnifiedGroupOwners -Identity $groupId
    Retrieves all the owners of a specific Office 365 Group based on its ID
    -----EXAMPLE 2-----
    PS:> Get-PnPUnifiedGroupOwners -Identity $group
    Retrieves all the owners of a specific Office 365 Group based on the group's object instance

PS C:\> Get-PnPUnifiedGroup | Get-PnPUnifiedGroupOwners
cmdlet Get-PnPUnifiedGroupOwners at command pipeline position 2
Supply values for the following parameters:
(Type !? for Help.)
Identity:
PS C:\> Get-PnPUnifiedGroup | ForEach-Object { $o = (Get-PnPUnifiedGroupOwners -Identity $_.groupId).UserPrincipalName ;
write-host "$($_.displayname) $o" }
Podcast Team todd@toddklindt.com
Podcast todd@toddklindt.com
PublicTest todd@toddklindt.com
Another Test todd@toddklindt.com
ADGroupTest01 todd@toddklindt.onmicrosoft.com todd@toddklindt.com
ADGroupTest02 todd@klindt.org
PS C:\>
```







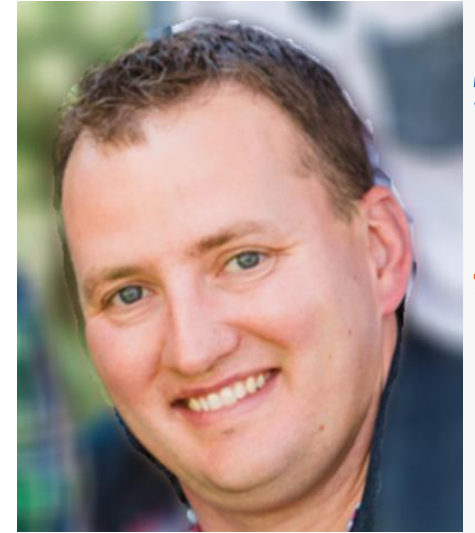
- [Script to connect to Office 365/Exchange](#)
  - <https://eightwone.com/2015/08/31/connecting-to-office-365exchange/>



Questions?

# Contact Us

[Shane.Young@BoldZebras.com](mailto:Shane.Young@BoldZebras.com)  
@ShanesCows  
[www.PowerApps911.com](http://www.PowerApps911.com)



[todd@toddklindt.com](mailto:todd@toddklindt.com)  
@toddklindt  
[www.toddklindt.com](http://www.toddklindt.com)

