

PowerShell and Microsoft 365

Woo Hoo!

Todd Klindt



15 Year SharePoint MVP

Writer, speaker, podcaster, consultant
at Sympraxis Consulting

todd.klindt@sympraxisconsulting.com
[@toddklindt](#)
www.toddklindt.com



Agenda

- PowerShell Love
- Talk about Microsoft's Modules
- Talk about PnP
- Look at some fun scripts



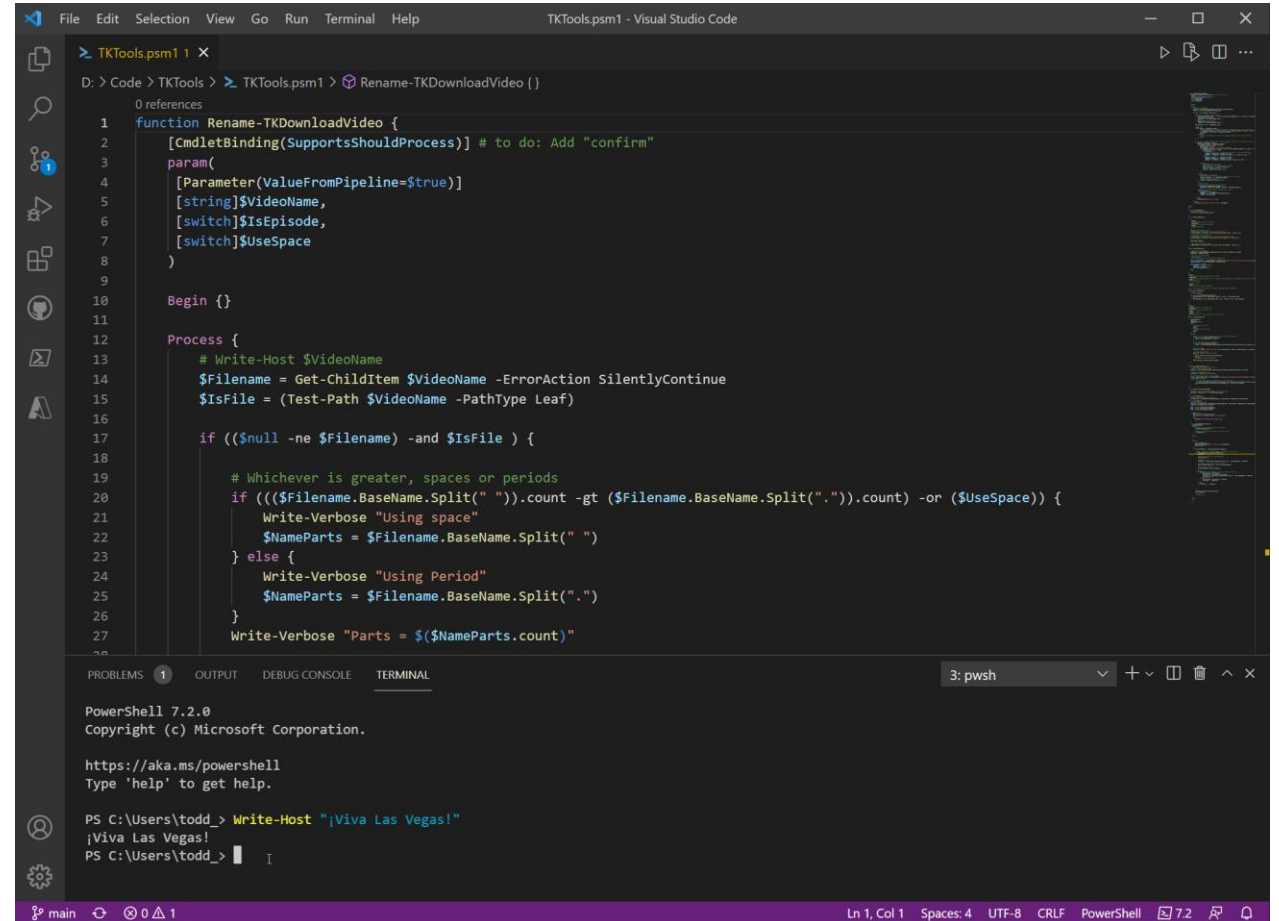
Why PowerShell Anyway?

- Repeatability
- Consistency
- Free Documentation
- Be the toolmaker, not the tool user



VS Code

- Not just for developers
- [Free download](#)
- [PowerShell Extension](#)
- Not a bad place
- Source Control
- Github
- Azure Dev Ops
- In the browser
 - <https://vscode.dev>
 - <https://github1s.com/pnp/powershell>



The screenshot shows the Visual Studio Code interface. The main editor window displays a PowerShell script named `TKTools.psm1` with the following content:

```
function Rename-TKDownloadVideo {  
    [CmdletBinding(SupportsShouldProcess)] # to do: Add "confirm"  
    param(  
        [Parameter(ValueFromPipeline=$true)]  
        [string]$VideoName,  
        [switch]$IsEpisode,  
        [switch]$UseSpace  
    )  
  
    Begin {}  
  
    Process {  
        # Write-Host $VideoName  
        $Filename = Get-ChildItem $VideoName -ErrorAction SilentlyContinue  
        $IsFile = (Test-Path $VideoName -PathType Leaf)  
  
        if (($null -ne $Filename) -and $IsFile) {  
            # Whichever is greater, spaces or periods  
            if (((($Filename.BaseName.Split(" ")).count -gt ($Filename.BaseName.Split(".")).count) -or ($UseSpace)) {  
                Write-Verbose "Using space"  
                $NameParts = $Filename.BaseName.Split(" ")  
            } else {  
                Write-Verbose "Using Period"  
                $NameParts = $Filename.BaseName.Split(".")  
            }  
            Write-Verbose "Parts = $($NameParts.count)"  
        }  
    }  
}
```

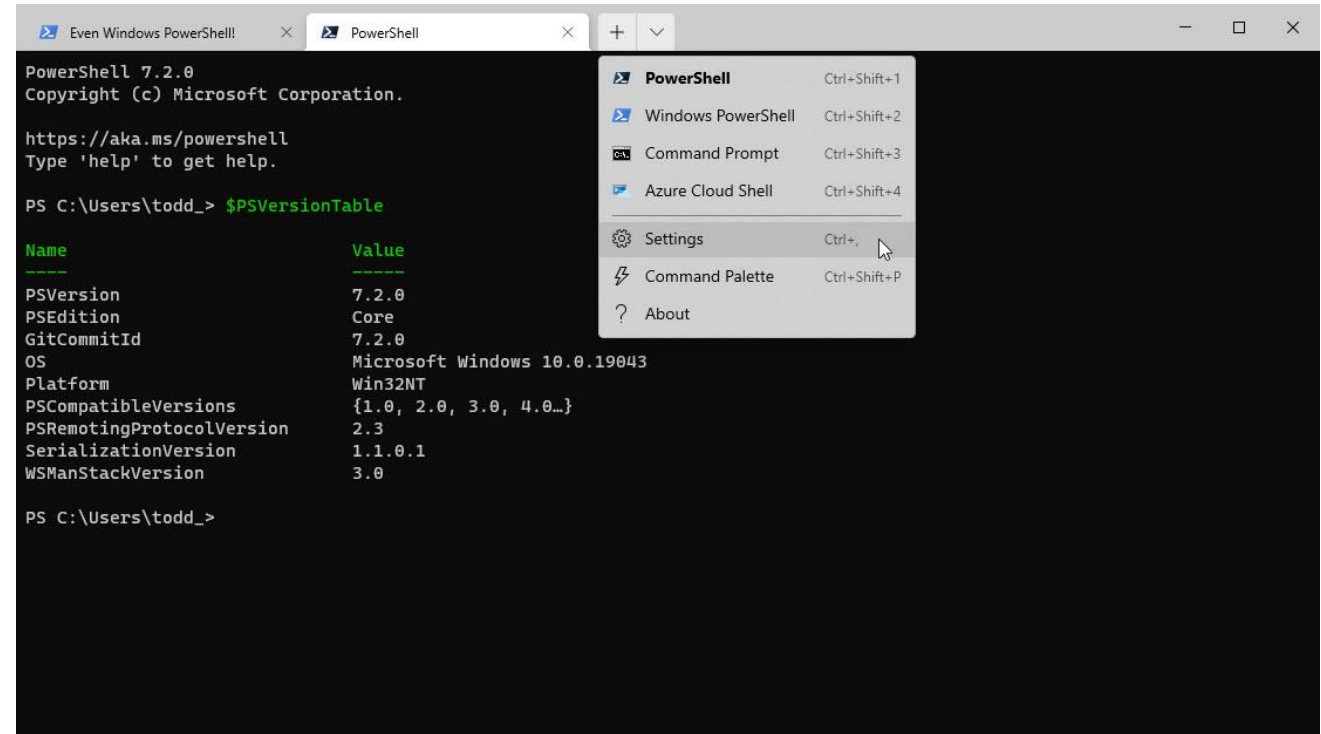
The bottom panel shows the `TERMINAL` window with the following output:

```
PowerShell 7.2.0  
Copyright (c) Microsoft Corporation.  
  
https://aka.ms/powershell  
Type 'help' to get help.  
  
PS C:\Users\todd> Write-Host "¡Viva Las Vegas!"  
¡Viva Las Vegas!  
PS C:\Users\todd>
```

The status bar at the bottom indicates the current file is `main`, the version is `0.0.1`, and the encoding is `UTF-8`.

Windows Terminal

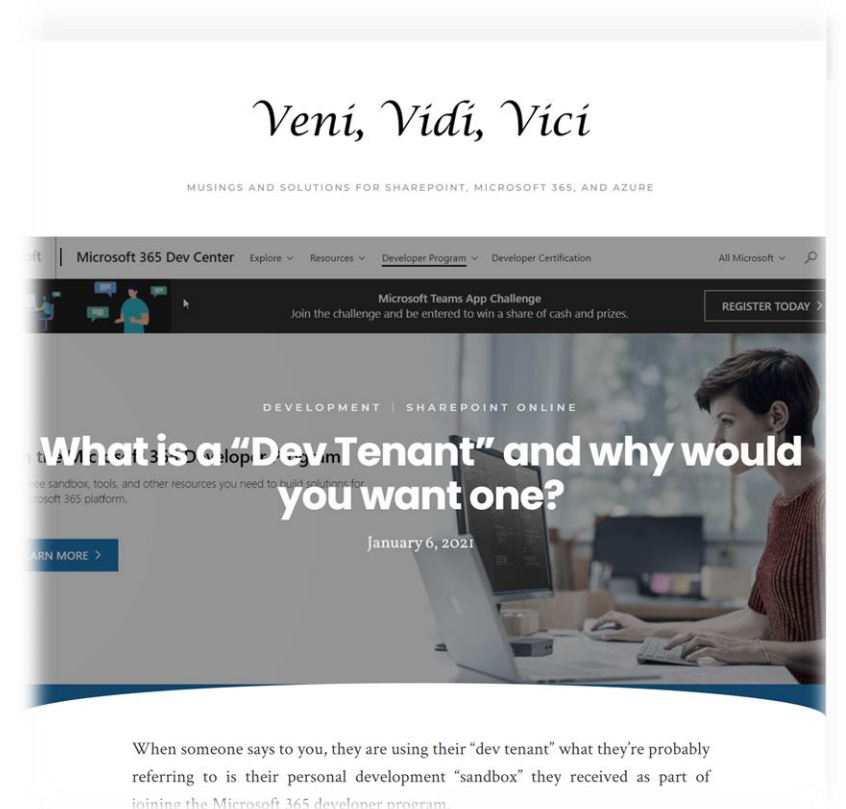
- The Future
- Windows Store
- Tabs



- <https://github.com/Microsoft/Terminal>

Get a Dev Tenant

- Great place to noodle around
- Free
- 100% less chance you'll be fired
- <https://www.toddklindt.com/GetDevTenant>



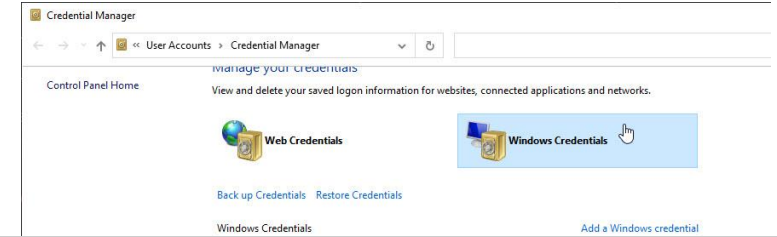
Let's talk about Passwords

- You will need your M365 username and password a lot
- Good and bad options
- Annoying but secure
 - \$Credentials = Get-Credential
- Less annoying and way, way less secure
 - \$username = admin@company.onmicrosoft.com
 - \$password = "RightHereInPlainText"
 - \$secure = \$password | ConvertTo-SecureString -AsPlainText -Force
 - \$Credentials = New-Object System.Management.Automation.PSCredential (\$username, \$secure)



Windows Credential Manager

- Built into Windows
- Let Microsoft worry about them securely
- Add-PnPStoredCredential
- Get-PnPStoredCredential

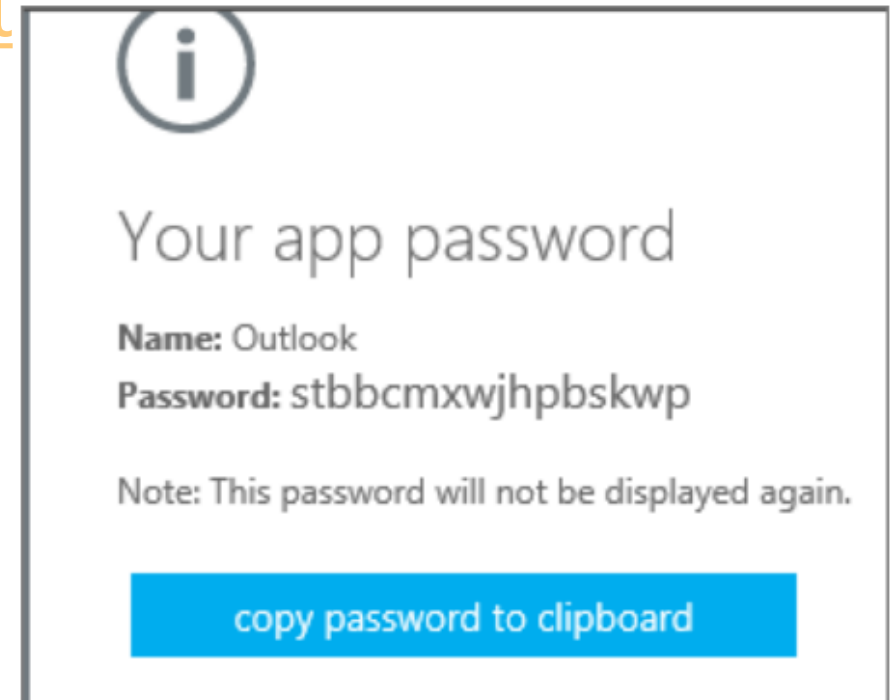


```
PS C:\Users\todd> $Creds = Get-PnPStoredCredential -Name https://m365x995492-admin.sharepoint.com
PS C:\Users\todd> Connect-SPOService -Url https://m365x995492-admin.sharepoint.com -Credential $Creds
PS C:\Users\todd> Get-SPOSite
```

Url	Owner	Storage Quota
https://m365x995492.sharepoint.com/search		26214400
https://m365x995492.sharepoint.com/sites/STS3	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/RetailOperations	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/SalesAndMarketing		26214400
https://m365x995492.sharepoint.com/sites/Petri1		26214400
https://m365x995492.sharepoint.com/sites/ContosoNews	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/livelycreditor8		26214400
https://m365x995492-my.sharepoint.com/		26214400
https://m365x995492.sharepoint.com/sites/ContosoBrand	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/USSales		26214400
https://m365x995492.sharepoint.com/sites/Communications		26214400
https://m365x995492.sharepoint.com/sites/Remoteliving		26214400
https://m365x995492.sharepoint.com/sites/GlobalMarketing	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/operations		26214400
https://m365x995492.sharepoint.com/sites/Leadership-connection	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/benefits	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/delete21		26214400
https://m365x995492.sharepoint.com/sites/TestCommSite	adelev@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/DigitalInitiativePublicRelations		26214400
https://m365x995492.sharepoint.com/sites/TestSite9	adelev@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/contosoteam		26214400
https://m365x995492.sharepoint.com/sites/massivedepartment39		26214400
https://m365x995492.sharepoint.com/sites/SOCTeam		26214400
https://m365x995492.sharepoint.com/sites/office365adoption		26214400

A Quick Word about MFA

- Depends on your authentication provider
- Enable MFA at the tenant (AzureAD) level
- Create an app password for your account



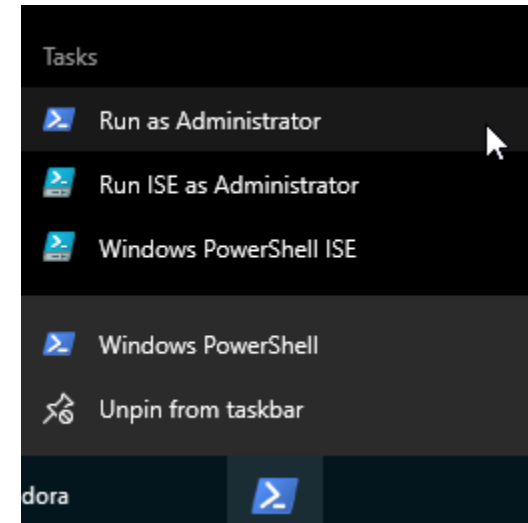
Certificates

- Good for server to server authentication
- Better than storing username and password or secret
- Self signed cert for testing
- More on this later

Microsoft Modules

Before you Connect

- Default positions
 - Have to be able to Run PowerShell as an Administrator
 - Have to be an Microsoft 365 Global Administrator
 - Except Exchange
 - Recommend 5.1 on your Windows desktop
 - No PowerShell 7 support yet ☹️
 - Execution policy needs to be RemoteSigned
 - Install Sign-in Assistant – 64bit



Connect to your Azure AD Tenant

- MSOnline (Deprecated)

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

- AzureAD (Net Deprecated Yet)

```
$Credentials = Get-PnPStoredCredential -Name https://m365x995492-admin.sharepoint.com  
Connect-AzureAD -Credential $Credentials  
Get-AzureADUser  
Get-Command -Module AzureAD
```



Exchange Online

- Just a little different
 - Install-Module -Name ExchangeOnlineManagement
 - Part install, part download
 - Supports PowerShell 7
 - Supports Modern Auth including MFA



PowerShell with SharePoint Online

- Be prepared for disappointment
 - Getting better with age
- Allows basic manipulation of SharePoint Online
 - Users and groups
 - Tenants
 - Site Collections
 - Hub Sites
 - Multi-Geo
- `Install-Module -Name Microsoft.Online.SharePoint.PowerShell`



Connect to SharePoint Online



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

Microsoft Teams!!

- Woo Hoo!
- For automating all those Teams Tasks

```
Install-Module MicrosoftTeams
```

```
$Credentials = Get-Credential
```

```
Connect-MicrosoftTeams -Credential $Credentials
```

```
Get-Command -Module MicrosoftTeams
```



Connect to Skype for Business



```
$Skype = New-CsOnlineSession -Credential $Credentials  
Import-PSSession $Skype  
Get-CsOnlineUser  
Remove-PSSession $Skype
```

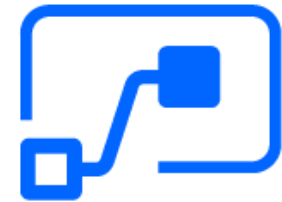
- This one can be confusing. Remember that Skype for Business, Lync, and Communication Server are all the same thing. The cmdlets and documentation tend to use them interchangeably.
☹️

Power Automate and Power Apps



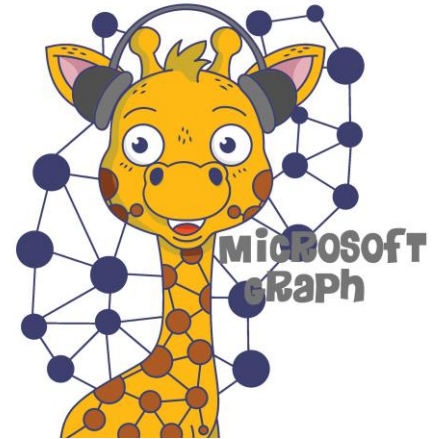
- Flow and Power Apps

- For both creators and Admins
- Get list of all Flows and Power Apps
- `Install-Module -Name Microsoft.PowerApps.Administration.PowerShell`
- `Install-Module -Name Microsoft.PowerApps.PowerShell -AllowClobber`
- `Add-PowerAppsAccount`



Graph SDK

- Got its fingers in everything
- Find project on Github
 - <https://github.com/microsoftgraph/msgraph-sdk-powershell>
- Not for the faint of heart
- Cmdlets mostly generated by AutoREST
- Future of AzureAD module



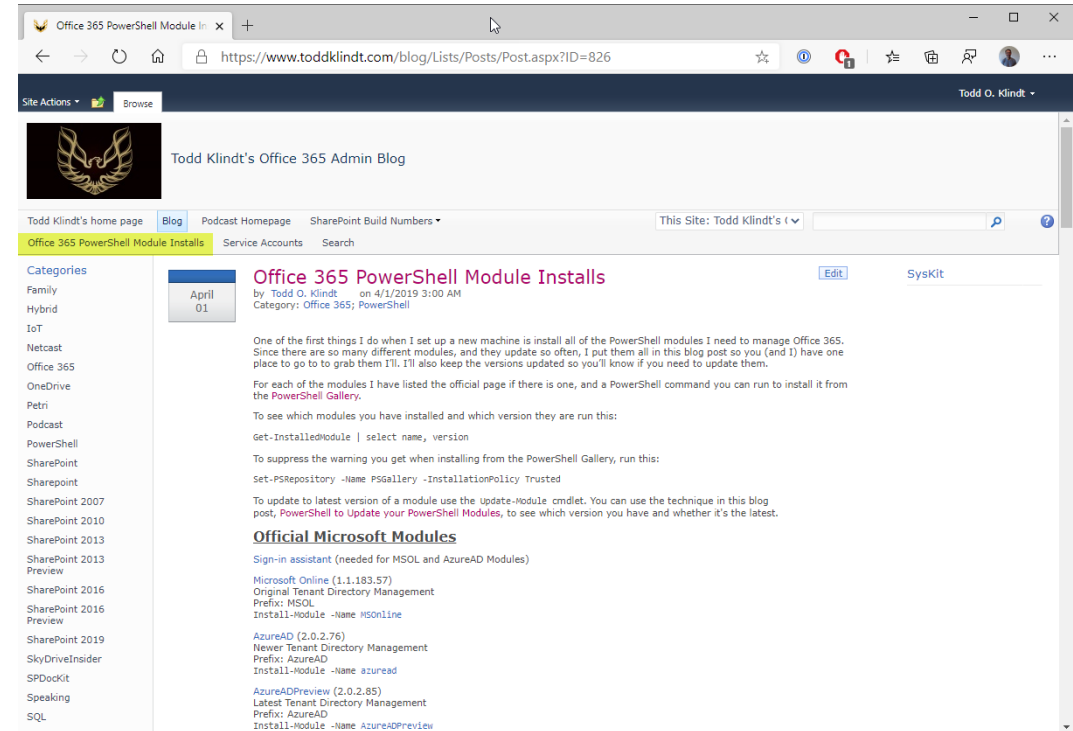
Office 365 / Microsoft 365 / Unified Groups

- No Groups module
- MSOL / Azure
- Exchange
- Teams
- PnP PowerShell
- Graph SDK



Things to Install

- [Sign-in Assistant](#) – 64bit
- Install-Module MSOnline
- Install-Module AzureAD
- Install-Module PnP.PowerShell
- Install-Module Microsoft.Online.SharePoint.PowerShell
- Install-Module MicrosoftTeams
- Install-Module Microsoft.PowerApps.Administration.PowerShell
- Install-Module Microsoft.PowerApps.PowerShell -AllowClobber
- Do all in Windows PowerShell Admin Console
- -Force -Scope AllUsers
- Get-InstalledModule
- Get-Module -ListAvailable
- Update-Module Microsoft.Online.SharePoint.PowerShell



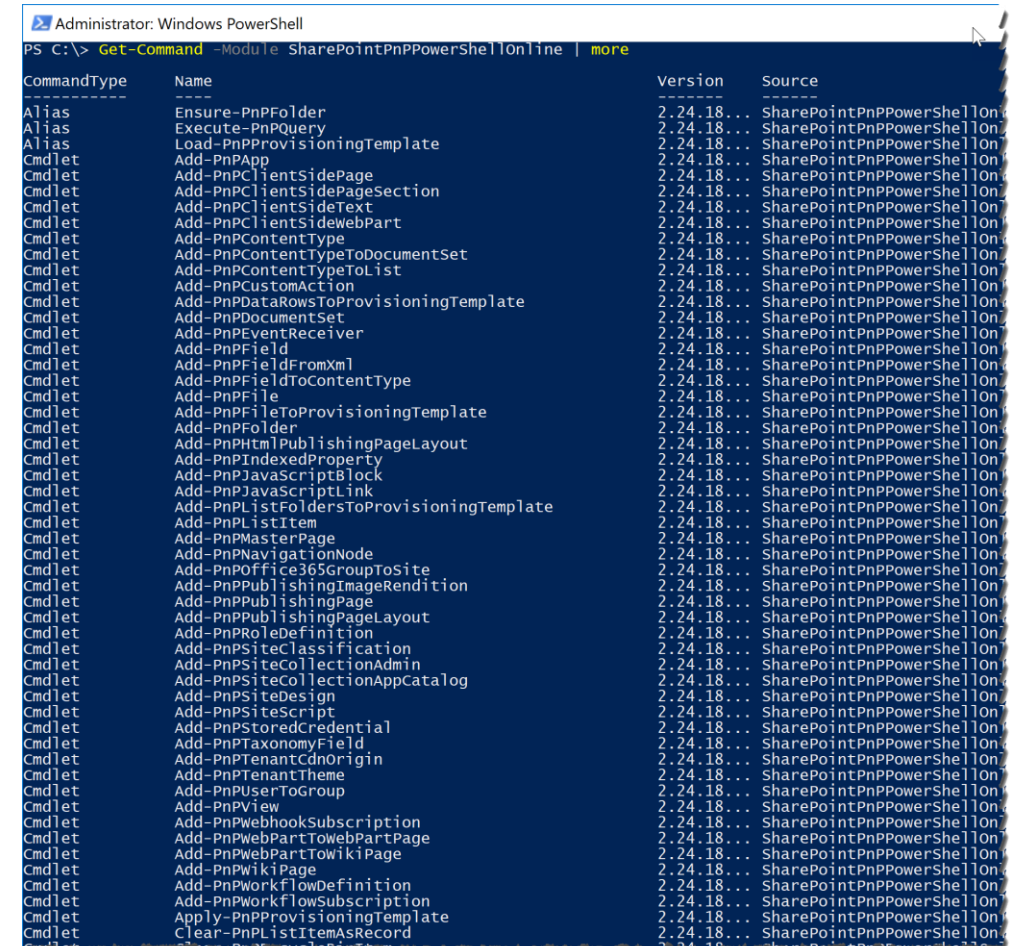
<https://www.toddklindt.com/POSHOffice365Modules>

<https://www.toddklindt.com/PoshUpdateModules>

PnP PowerShell!!!

Patterns and Practices PowerShell (Phew!)

- GitHub
 - <https://github.com/pnp/powershell>
- Adds 570+ more cmdlets
- Scoped at Site Collection
- Install-Module PnP.PowerShell
- Get-Command -Module PnP.PowerShell
- Use SharePointPnPPowerShellOnline for on-prem SharePoint



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

CommandType      Name                                     Version      Source
-----
Alias             Ensure-PnPFolder                       2.24.18...   SharePointPnPPowerShellOnline
Alias             Execute-PnPQuery                       2.24.18...   SharePointPnPPowerShellOnline
Alias             Load-PnPProvisioningTemplate           2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPApp                             2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPClientSidePage                  2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPClientSidePageSection           2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPClientSideText                  2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPClientSideWebPart               2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPContentType                     2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPContentTypeToDocumentSet         2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPContentTypeToList               2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPCustomAction                    2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPDataRowsToProvisioningTemplate   2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPDocumentSet                     2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPEventReceiver                   2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPField                           2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPFieldFromXml                    2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPFieldToContentType               2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPFile                             2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPFileToProvisioningTemplate       2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PnPFolder                           2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPHTMLPublishingPageLayout         2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPIndexedProperty                  2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPJavaScriptBlock                  2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPJavaScriptLink                   2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPListFolderToProvisioningTemplate 2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPListItem                         2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPMasterPage                       2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPNavigationNode                   2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPOffice365GroupToSite             2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPublishingImageRendition          2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPublishingPage                    2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPublishingPageLayout              2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPRoleDefinition                   2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPSiteClassification               2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPSiteCollectionAdmin              2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPSiteCollectionAppCatalog         2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPSiteDesign                       2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPSiteScript                       2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPStoredCredential                  2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPtaxonomyField                     2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPtenantCdnOrigin                   2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPtenantTheme                       2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPUserToGroup                      2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPView                             2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPwebhookSubscription               2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPwebPartToWebPartPage             2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPwebPartToWikiPage                2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPWikiPage                         2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPWorkflowDefinition                2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Add-PNPWorkflowSubscription              2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Apply-PNPProvisioningTemplate            2.24.18...   SharePointPnPPowerShellOnline
Cmdlet            Clear-PNPListItemAsRecord                2.24.18...   SharePointPnPPowerShellOnline
```

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-PnPListItem
- Get-PnPProperty
- Set-PnPGroupPermissions
- Add-PnPView
- Get-PnPField
- Restore-PnPRecycleBinItem
- Office 365 Groups
 - New-PnPMicrosoft365Group, Remove-PnPMicrosoft365Group
 - Get-PnPMicrosoft365Group, Set-PnPMicrosoft365Group
 - Get-PnPMicrosoft365GroupMembers
 - Get-PnPMicrosoft365GroupOwners

PnP Authentication

```
Add-PnPStoredCredential -Name "https://tenant.sharepoint.com" -Username  
yourname@tenant.onmicrosoft.com
```

```
Connect-PnPOnline -Url "https://tenant.sharepoint.com"
```

```
Connect-PnPOnline -Url "https://tenant.sharepoint.com/sites/hr"
```

```
$Credentials = Get-PnPStoredCredential -Name  
"https://tenant.sharepoint.com"
```

```
Connect-SPOService -Url https://tenant-admin.sharepoint.com -  
Credentials $ Credentials
```

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 used to work 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

PnP Vs. SPO

- SPO

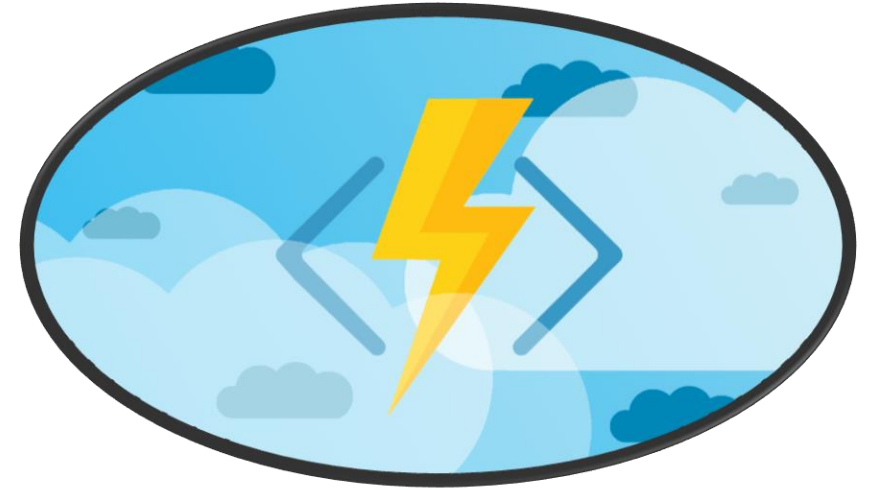
- Tenant Wide tasks
- Basic tasks
- New functionality that hasn't made it into SharePoint Admin Center
- New functionality that hasn't made it into PnP yet

- PNP

- Everything SPO can do
- Tasks by site owners
- Tasks by users
- Site specific settings
- Working on an object smaller than a site
- More specific tasks
- Want to be cooler

Azure Function

- Great serverless automation
- Can be trigger from Flow
 - Http
 - Azure Queue
- PowerShell 7 environment (PnP)
- Use Certificate authentication
- Can publish with VS Code
- Use Console to troubleshoot



Examples

Upload a File

```
$web = https://tenant.sharepoint.com/sites/hr
```

```
$folder = "Shared Documents"
```

```
Connect-PnPOnline -Url $web
```

```
Add-PnPFile -Path '.\Boot fairs with Graphic design.docx' -  
Folder $folder
```


Add a Folder

```
Add-PnPFolder -Name "Folder 1" -Folder $folder
```

```
Add-PnPFile -Path '.\Building materials licences to budget  
for Storytelling.docx' -Folder "$folder\Folder 1"
```

Get Internal Shared Files

```
$doclibs = Get-PnPList -Includes DefaultViewUrl,IsSystemList | Where-Object -Property IsSystemList -EQ -Value $false | Where-Object -Property BaseType -EQ -Value "DocumentLibrary"
Foreach ($doclib in $doclibs)
{
    $docs = Get-PnPListItem -List $DocLib
    foreach ($doc in $docs) {
        if (($doc.FieldValues).SharedWithUsers -ne $null) {
            foreach ($user in (($doc.FieldValues).SharedWithUsers)) {
                Write-Output "$(($doc.FieldValues).FileRef) - $($user.email)"
            }
        }
    }
}
```

Get Extended File Info

```
$doclibs = Get-PnPList -Includes DefaultViewUrl,IsSystemList | Where-Object -Property IsSystemList -EQ -Value $false | Where-Object -Property BaseType -EQ -Value "DocumentLibrary"
```

```
    Foreach ($doclib in $doclibs)
    {
        $doclibTitle = $doclib.Title
        $docs = Get-PnPListItem -List $DocLib
        $docs | ForEach-Object { Get-PnPProperty -ClientObject $_ -Property File, ContentType, ComplianceInfo}
        foreach ($doc in $docs) {
            [pscustomobject]@{Library= $doclibTitle;Filename = ($doc.File).Name;ContentType = ($doc.ContentType).Name;Label = ($doc.ComplianceInfo).ComplianceTag}
        }
    }
```

Bulk Undelete Files

```
Connect-PnPOnline -Url https://sadtenant.sharepoint.com/ -Credentials  
SadTenantAdmin
```

```
$bin = Get-PnPRecycleBinItem | Where-Object -Property Leafname -Like -Value  
"*.*jpg" | Where-Object -Property Dirname -Like -Value "Important  
Photos/Shared Documents/*" | Where-Object -Property DeletedByEmail -EQ -  
Value baduser@sadtenant.phooey
```

```
$bin.count
```

```
$bin | foreach -begin { $a = 0 } -Process {Write-Host "$a - $($_.LeafName)"  
; $_ | Restore-PnPRecycleBinItem -Force ; $a++ } -End { Get-Date }
```

```
($bin[20001..30000]) | foreach -begin { $a = 0 } -Process {Write-Host "$a -  
$($_.LeafName)" ; $_ | Restore-PnPRecycleBinItem -Force ; $a++ } -End { Get-  
Date }
```

<https://www.toddklindt.com/PoshRestoreSPOFiles>

New Site and Group

- New-SPOSite
 - No Group, No Team, No Bueno
 - Can be Groupified later
- New-PnPSite -Type TeamSite -Title "Modern Team Site"
-Alias ModernTeamSite -IsPublic
 - Group, No Team
 - Can be Teamified later

The Whole Enchilada

- New-Team -DisplayName "Fancy Group" -Description "Fancy Group made by PowerShell?" -Alias FancyGroup -AccessType Public
 - There is no later!

"Save Site as Template"

- `Get-PnPSiteTemplate -Out customer.xml`
- `Add-PnPListFoldersToSiteTemplate -Path customer.xml -List 'Data Storage' -Recursive`
- `Invoke-PnPSiteTemplate -Path customer.xml -Handlers Lists, SiteSecurity`
- `Get-Command -Module PnP.PowerShell -Name *pnp*template*`

Copy Office 365 Group Permissions

```
# Set some values
# use Get-PnPUnifiedGroup to get Unified Group names
# Name of Unified Group whose owners and membership we want to copy
$source = "Regulations"
# Name of Unified Group whose owners and membership we want to populate
$destination = "Empty"
# Whether to overwrite Destination membership or merge them
$mergeusers = $false
# Check to see if PnP Module is loaded
$pnploaded = Get-Module PnP.PowerShell
if ($pnploaded -eq $false) {
    Write-Host "Please load the PnP PowerShell and run again"
    Write-Host "install-module PnP.PowerShell"
    break
}
```



```
# PnP Module is loaded
# Check to see if user is connected to Microsoft Graph
try
{
    $owners = Get-PnPMicrosoft365GroupOwners -Identity $source
}
catch [System.InvalidOperationException]
{
    Write-Host "No connection to Microsoft Graph found" -BackgroundColor Black -ForegroundColor Red
    Write-Host "No Azure AD connection, please connect first with Connect-PnPOnline -Graph" -BackgroundColor Black -ForegroundColor Red
    break
}
catch [System.ArgumentNullException]
{
    Write-Host "Group not found" -BackgroundColor Black -ForegroundColor Red
    Write-Host "Verify connection to Azure AD with Connect-PnPOnline -Graph" -BackgroundColor Black -ForegroundColor Red
    Write-Host "Use Get-PnPUnifiedGroup to get Unified Group names" -BackgroundColor Black -ForegroundColor Red
    break
}
catch
{
    Write-Host "Some other error" -BackgroundColor Black -ForegroundColor Red
    break
}
```

```
$members = Get-PnPUnifiedGroupMembers -Identity $source
if ($mergeusers -eq $true) {
    # Get existing owners and members of Destination so that we can combine them
    $ownersDest = Get-PnPMicrosoft365GroupOwners -Identity $destination
    $membersDest = Get-PnPUnifiedGroupMembers -Identity $destination
    # Add the two lists together so we don't overwrite any existing owners or members in Destination
    $owners = $owners + $ownersDest
    $members = $members + $membersDest
}
# Set the owners and members of Destination
$owners | ForEach-Object -begin {$ownerlist = @() } -process {$ownerlist += $($_.UserPrincipalName)}
$members | ForEach-Object -begin {$memberlist = @() } -process {$memberlist += $($_.UserPrincipalName)}
Set-PnPMicrosoft365Group -Identity $destination -Members $memberlist -Owners $ownerlist
# https://www.toddklindt.com/PoshCopyO365GroupMembers
```

Get All the Flows

Add-PowerAppsAccount

```
Get-AdminFlow | ForEach-Object { $ownername = (Get-MsolUser -ObjectId  
$_.CreatedBy.userId).DisplayName ; $owneremail = (Get-MsolUser -ObjectId  
$_.CreatedBy.userId).UserPrincipalName ; Write-Host $_.DisplayName, $ownername,  
$owneremail }
```

- Requires MSOL module and connection

Get Your Flows

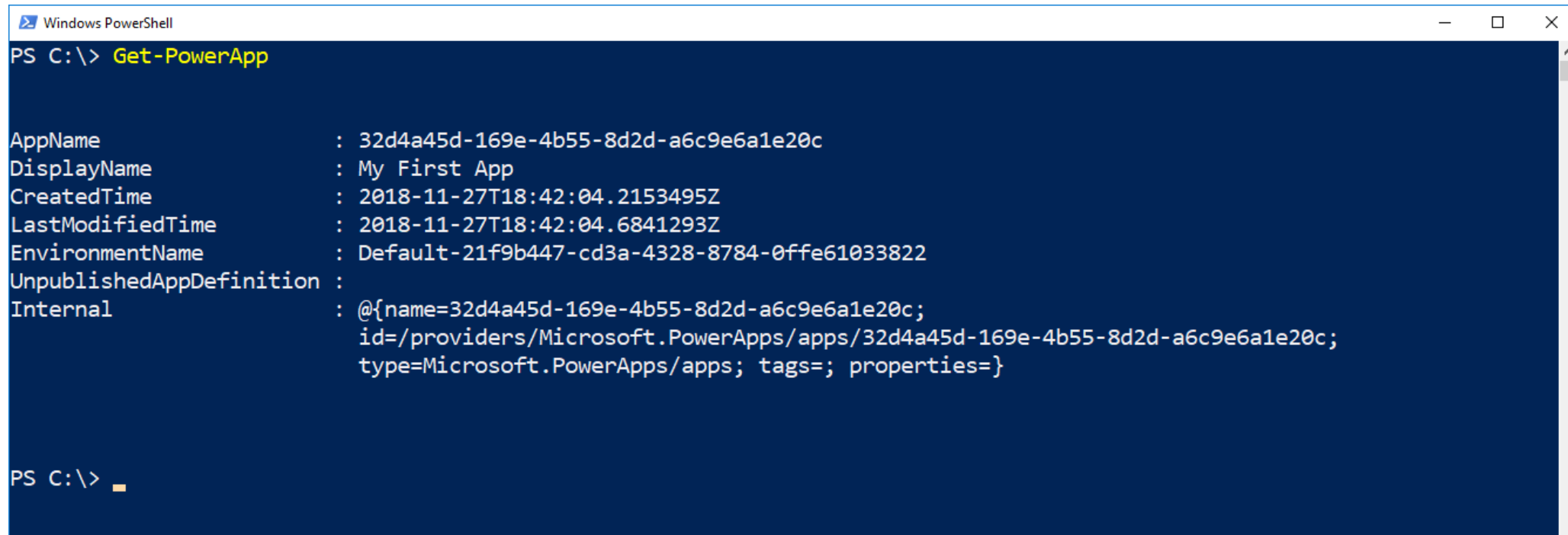
```
Windows PowerShell
PS C:\> Get-Flow

FlowName      : ddf63e93-3f0a-4a78-951e-46d2fa7eec63
Enabled       : True
DisplayName    : Send an email when a new item is created in SharePoint.
UserType      : Owner
CreatedTime   : 2018-11-27T18:25:32.6267962Z
LastModifiedTime : 2018-11-27T18:25:32.9141198Z
EnvironmentName : Default-21f9b447-cd3a-4328-8784-0ffe61033822
Internal      : @{name=ddf63e93-3f0a-4a78-951e-46d2fa7eec63; id=/providers/Microsoft.ProcessSimple/environments/Default-21f9b447-cd3a-4328-8784-0ffe61033822/flows/ddf63e93-3f0a-4a78-951e-46d2fa7eec63; type=Microsoft.ProcessSimple/environments/flows; properties=}

PS C:\>
```

And Power Apps Apps

- Get-PowerApp



```
Windows PowerShell
PS C:\> Get-PowerApp

AppName           : 32d4a45d-169e-4b55-8d2d-a6c9e6a1e20c
DisplayName        : My First App
CreatedTime        : 2018-11-27T18:42:04.2153495Z
LastModifiedTime   : 2018-11-27T18:42:04.6841293Z
EnvironmentName    : Default-21f9b447-cd3a-4328-8784-0ffe61033822
UnpublishedAppDefinition :
Internal           : @{name=32d4a45d-169e-4b55-8d2d-a6c9e6a1e20c;
                      id=/providers/Microsoft.PowerApps/apps/32d4a45d-169e-4b55-8d2d-a6c9e6a1e20c;
                      type=Microsoft.PowerApps/apps; tags=; properties=}

PS C:\>
```

Disable Flow Button

- SPO Method

- Connect-SPOService -Url <https://flowhater-admin.sharepoint.com>
- \$val = [Microsoft.Online.SharePoint.TenantAdministration.FlowsPolicy]::Disabled
- Set-SPOSite -Identity <https://flowhater.sharepoint.com/sites/SadSite> -DisableFlows \$val

- PnP Method

- Connect-PnPOnline -Url <https://flowhater.sharepoint.com/sites/SadSite> -Credentials 'SiteAdmin'
- Set-PnPSite -DisableFlows:\$true

Questions?



todd.klindt@sympraxisconsulting.com
[@toddklindt](#)
www.toddklindt.com

