PowerShell with SharePoint Server and Office 365

Shane Young

- 13 Year SharePoint MVP
- Shane@powerapps911.com
- @ShanesCows
- http://www.youtube.com/ShaneYoungCloud
- Cincinnati, Ohio
- https://www.PowerApps911.com
- https://www.BoldZebras.com

PowerApps911
PowerApps Help... Now!

BOLD ZEBRAS

MVP
Microsoft®
Most Valuable Professional

SPTechCon
The SharePoint and Office 365 Conference

Todd Klindt

12 Year SharePoint MVP
Writer, speaker, consultant, podcaster, SPDocKit marketer

todd@toddklindt.com
@toddklindt
www.toddklindt.com

Microsoft Most Valuable Professional

SPTechCon
The SharePoint and Office 365 Conference

Agenda

- Official Cmdlets
- Developery Option
- The Goods

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

Official Cmdlets

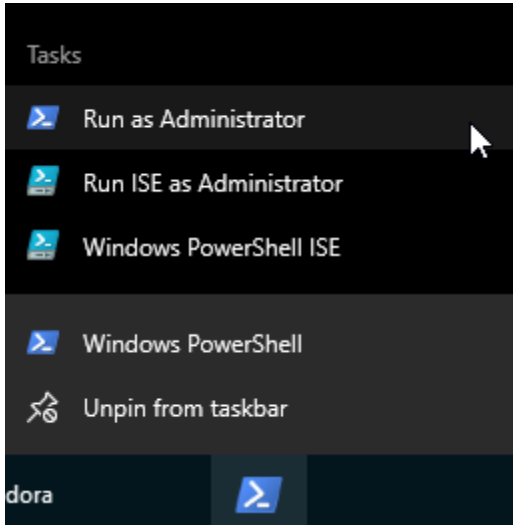
There are 4 things to install

- 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
```

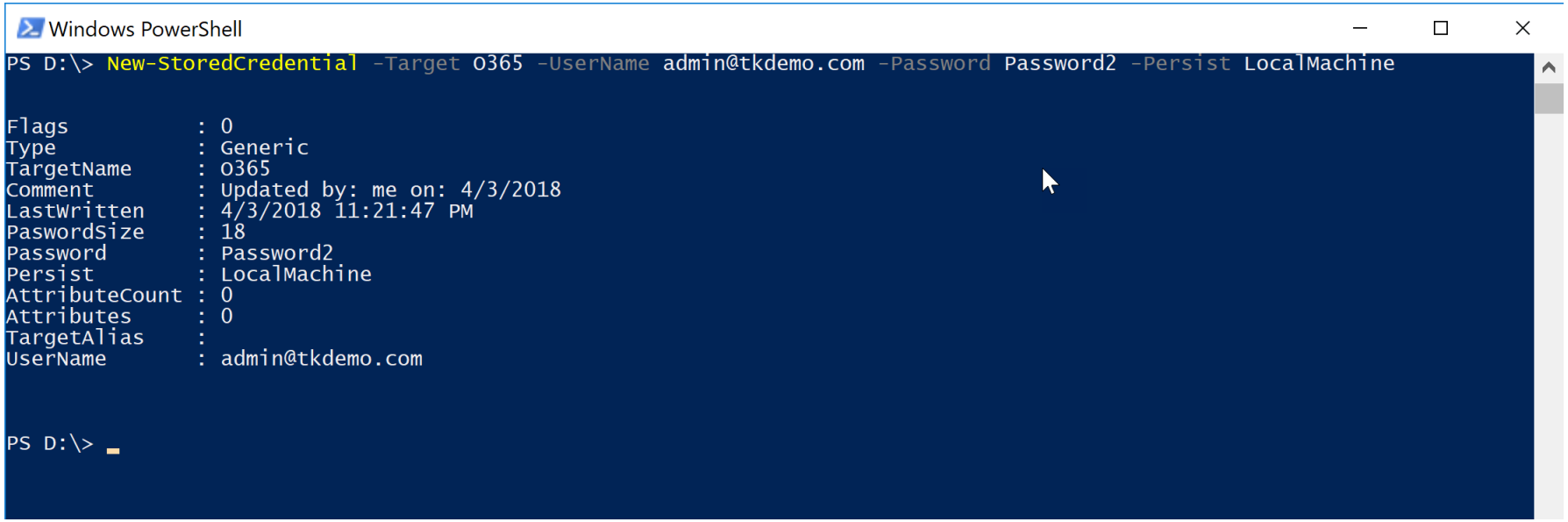
- Less annoying and way, way less secure

```
$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 O365 -UserName admin@tkdemo.com -Password Password2 -Persist LocalMachine`



Connect to your Azure AD Tenant

- MSOnline (v1)

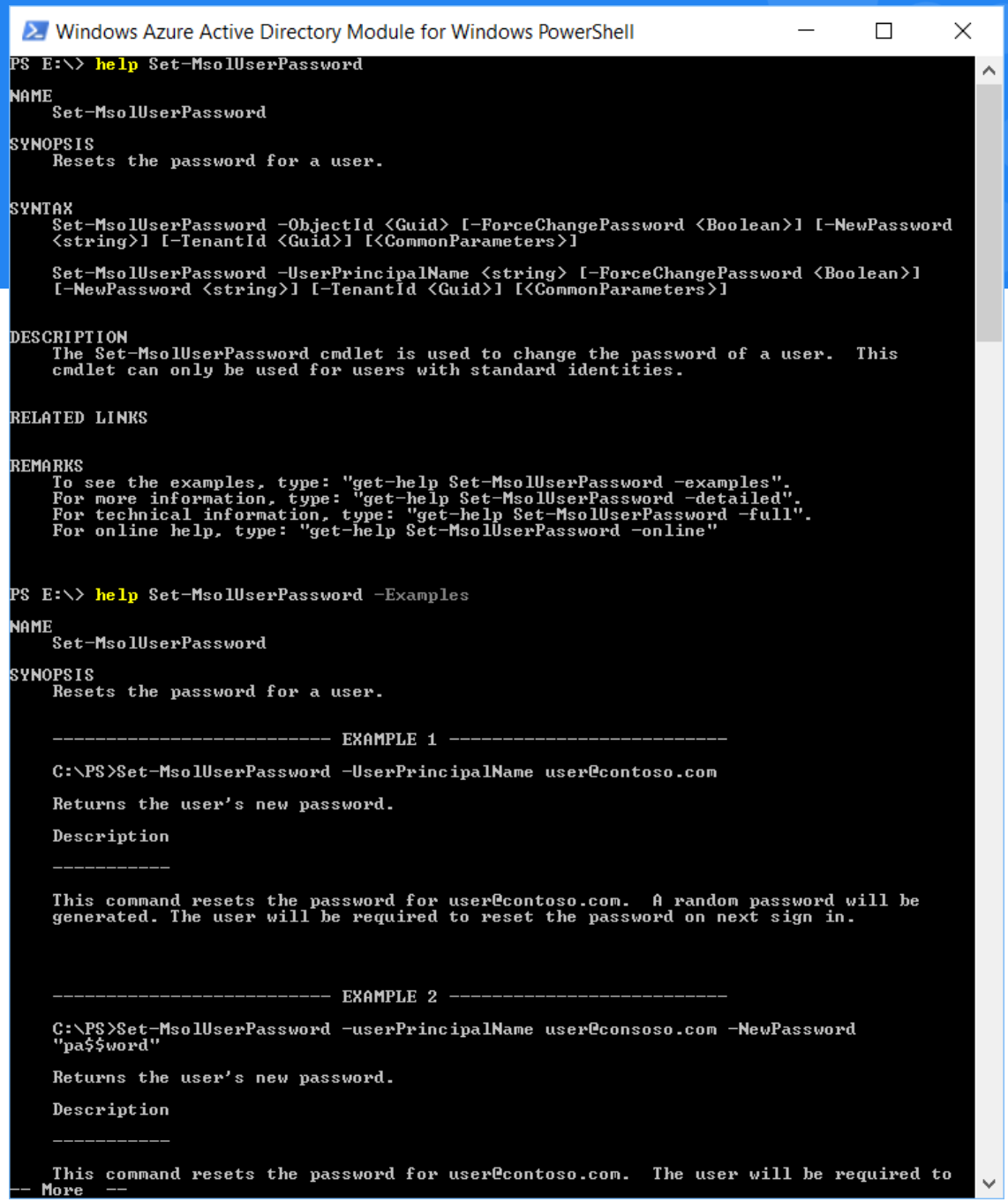
```
$MyAccount = Get-Credential
Connect-MsolService -Credential $MyAccount
Get-MsolUSer
Get-Command -Module msonline
```

- AzureAD (v2)

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

- Install-Module azuread

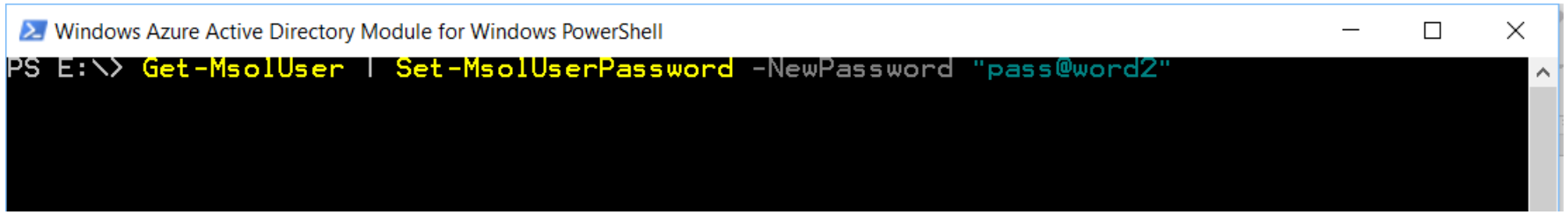
Fun Gotchas


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



Connect to Skype for Business

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

Import-PSSession $Skype

Get-CsOnlineUser

Remove-PSSession $Skype
```

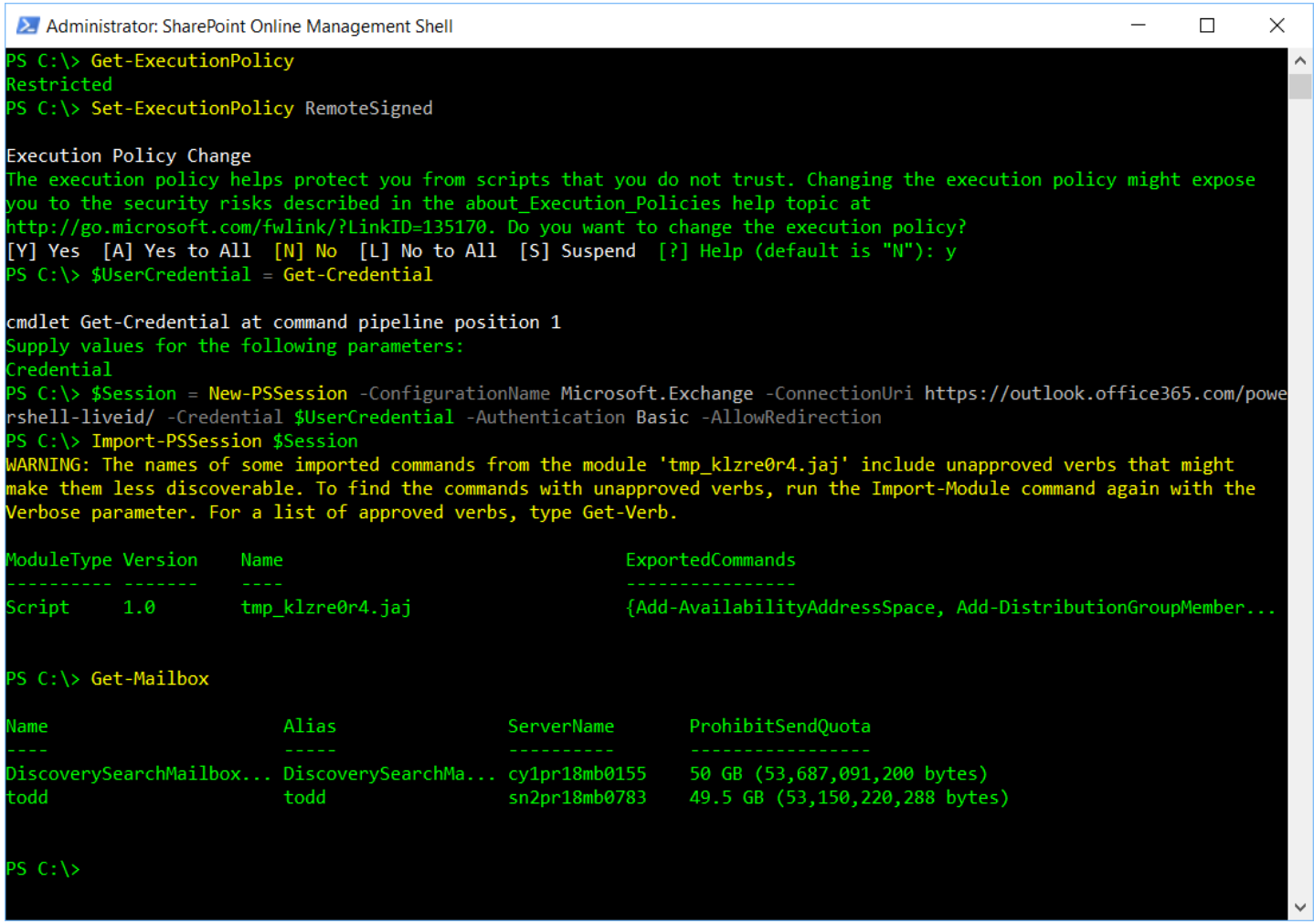
- This one can be confusing. Remember that Skype for Business, Lync, and Communication Server are all the same thing. The cmdlets and documentation tend to use them interchangeably. ☹

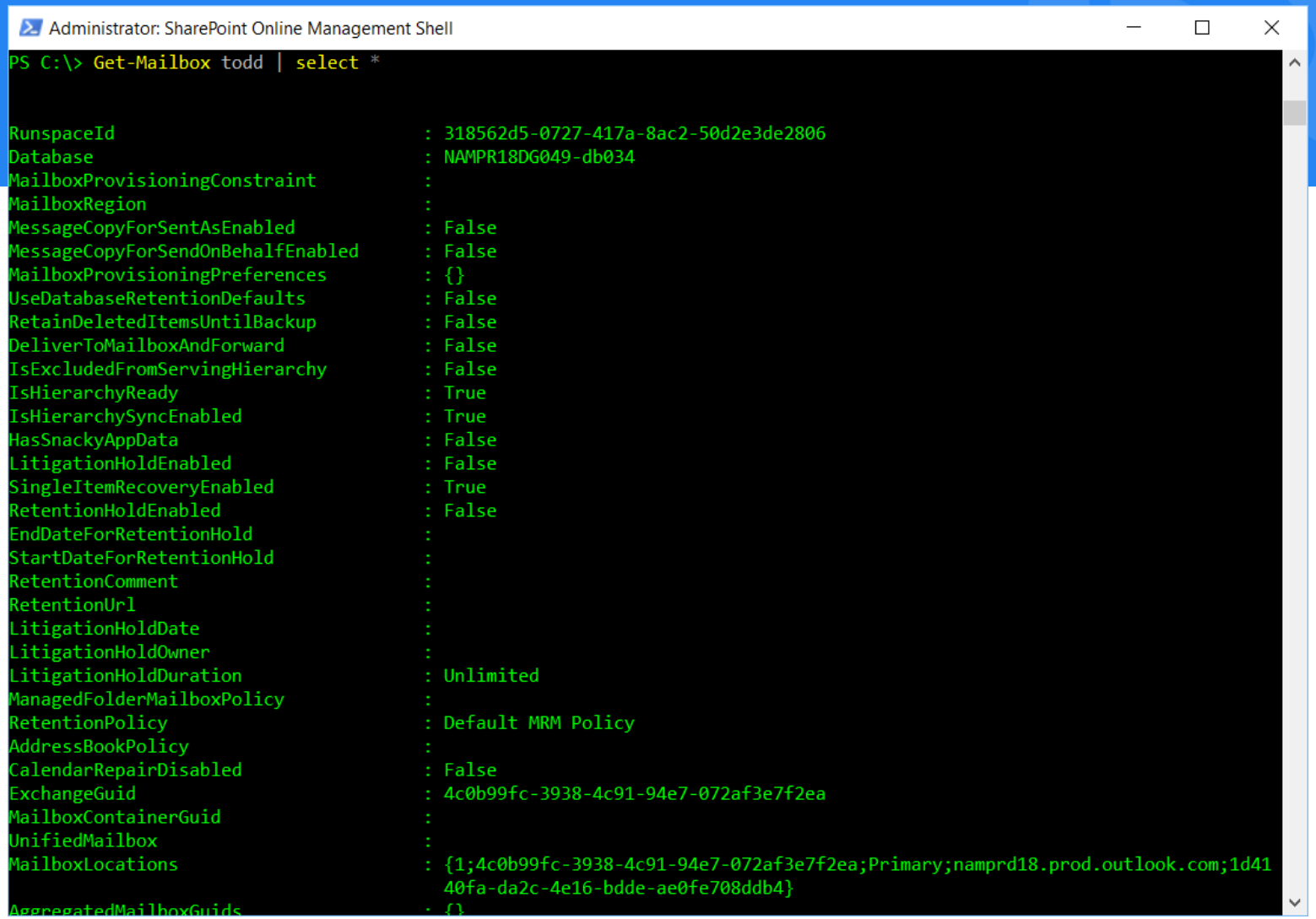
Connect to Exchange

```
$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
```

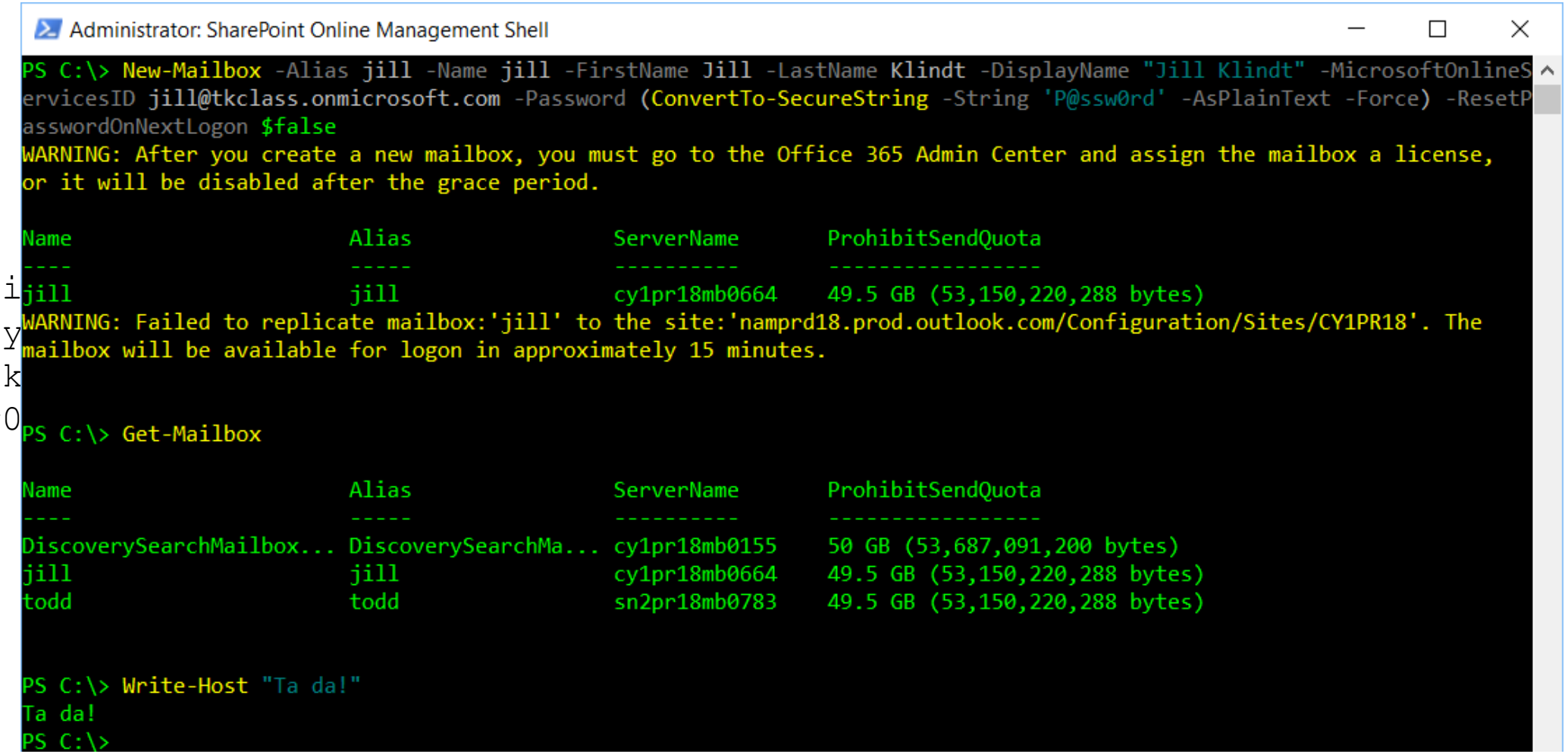
- Skype and Exchange are limited to 3 sessions so always end your session.

- Just a little different
  - No cmdlets, uses Remoting
  - Limited to three sessions
  - Requires port 80
  - Close out gracefully
    - Remove-PSSession $Session
  - Supports MFA



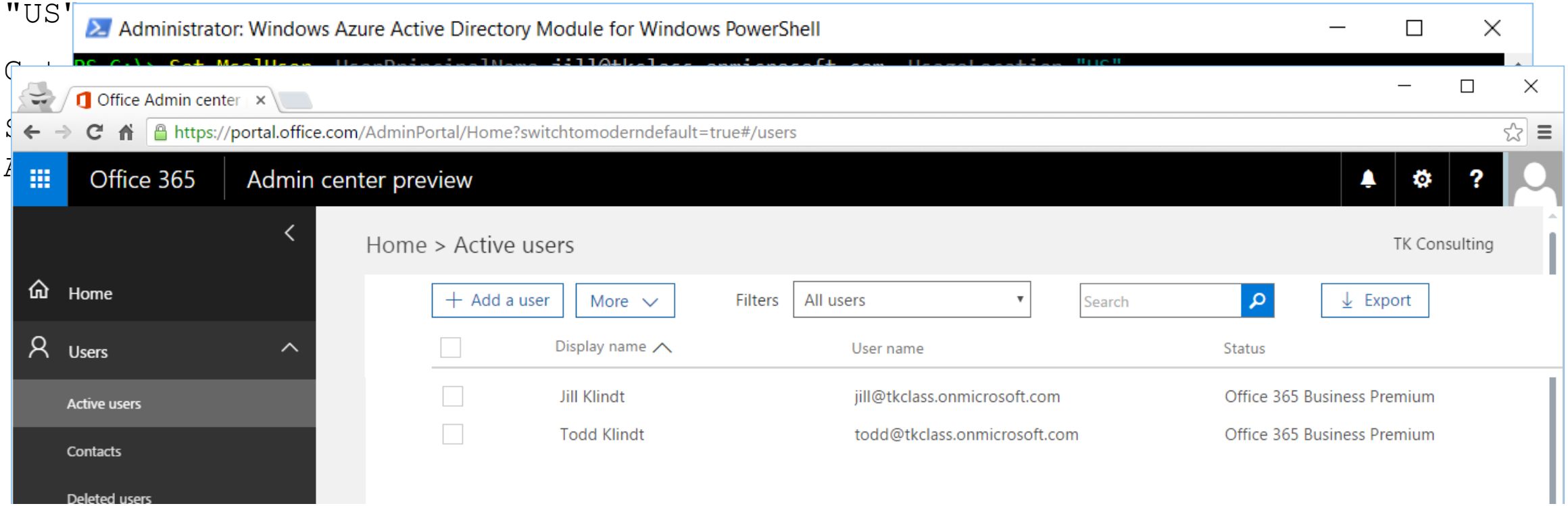

Administrator: SharePoint Online Management Shell
PS C:\> Get-Mailbox todd | select *
RunspaceId : 318562d5-0727-417a-8ac2-50d2e3de2806
Database : NAMPR18DG049-db034
MailboxProvisioningConstraint :
MailboxRegion :
MessageCopyForSentAsEnabled : False
MessageCopyForSendOnBehalfEnabled : False
MailboxProvisioningPreferences : {}
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;1d41
40fa-da2c-4e16-bdde-ae0fe708ddb4}
AggregatedMailboxGuids : {}

```
New-Mai
Display
jill@tk
'P@ssw0
```



License Up That New Mailbox

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



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"
```

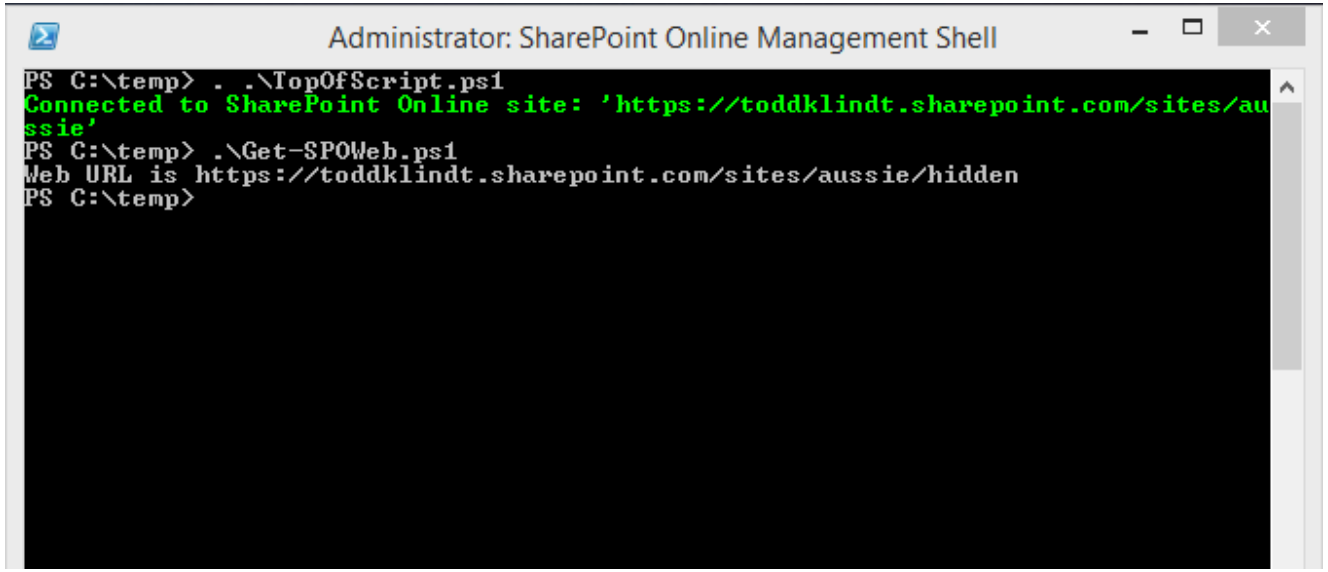
```
# Set the user's location. Without that the license will fail

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

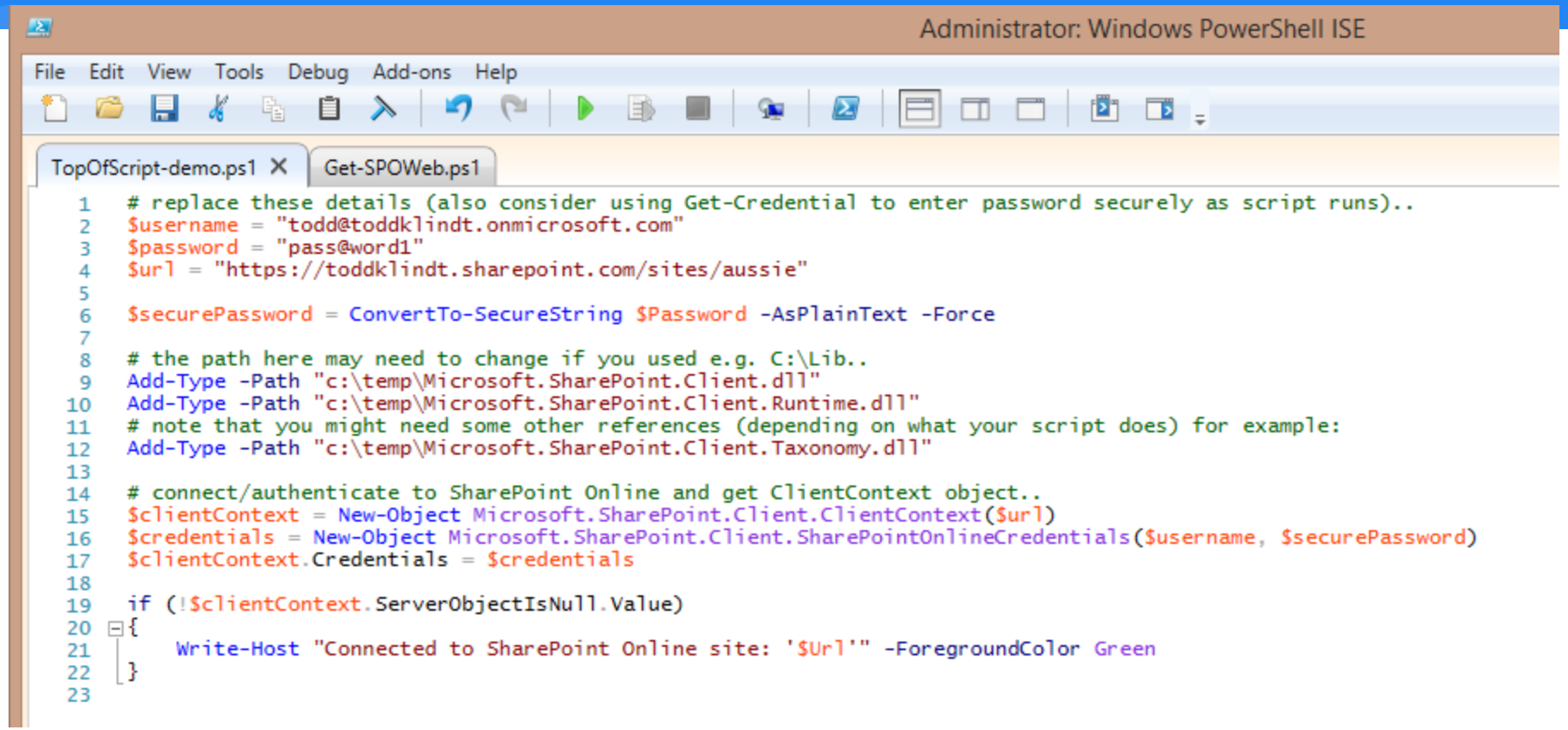
- Teams
  - For automating all those Teams Admin Tasks
  - Install-Module MicrosoftTeams
  - Read all about it
- Flow and PowerApps
  - For both creators and Admins
  - Get list of all Flows and PowerApps
  - Kind of a janky install

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

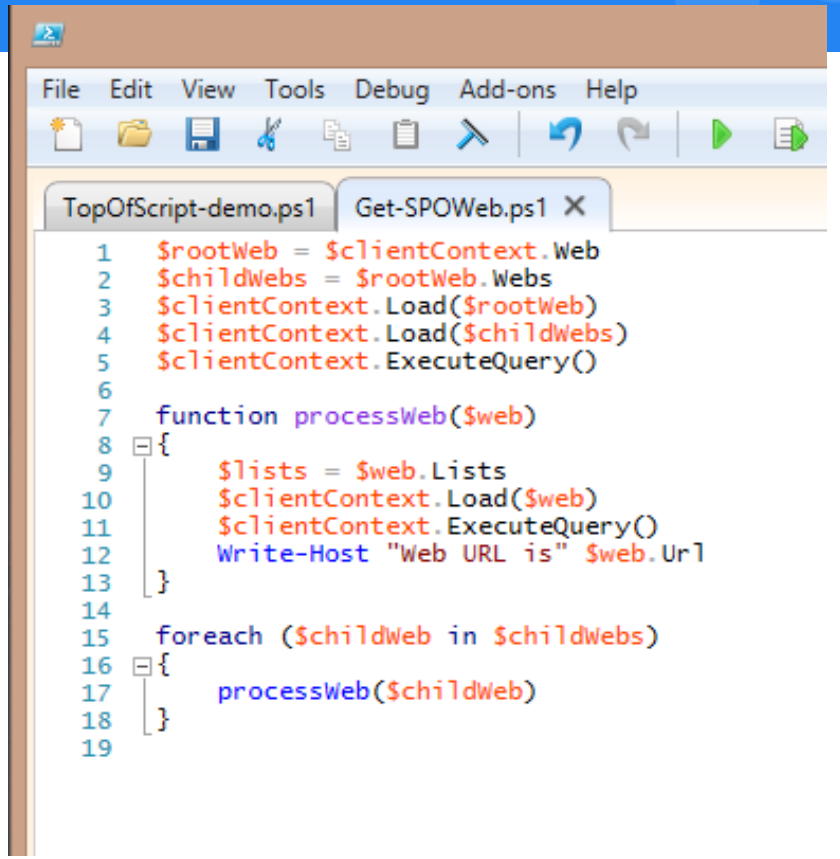


Top Of Script



Get-SPOweb

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



More Examples

TopOfScript-OnPrem.ps1*

```powershell
# replace these details
# Can also use Read-Host or technique in http://www.toddklindt.com/PoshSecurePasswords
$username = "todd@toddklindt.com"
$password = "pass@word1"
$url = "http://upgrade.toddklindt.com/blog"

$securePassword = ConvertTo-SecureString $Password -AsPlainText -Force
# $securePassword = Read-Host "Enter password for $username" -AsSecureString

# the path here may need to change if you used e.g. C:\Lib..
Add-Type -Path "c:\temp\Microsoft.SharePoint.Client.dll"
Add-Type -Path "c:\temp\Microsoft.SharePoint.Client.Runtime.dll"
# note that you might need some other references (depending on what your script does) for example:
Add-Type -Path "c:\temp\Microsoft.SharePoint.Client.Taxonomy.dll"

# connect/authenticate to SharePoint Online and get ClientContext object..
$clientContext = New-Object Microsoft.SharePoint.Client.ClientContext($url)
# $credentials = New-Object Microsoft.SharePoint.Client.SharePointOnlineCredentials($username, $securePassword)
$credentials = New-Object System.Management.Automation.PSCredential ($username, $securePassword)
$clientContext.Credentials = $credentials

if (!$clientContext.ServerObjectIsNull.Value)
{
    Write-Host "Connected to SharePoint Online site: '$Url'" -ForegroundColor Green
}
```

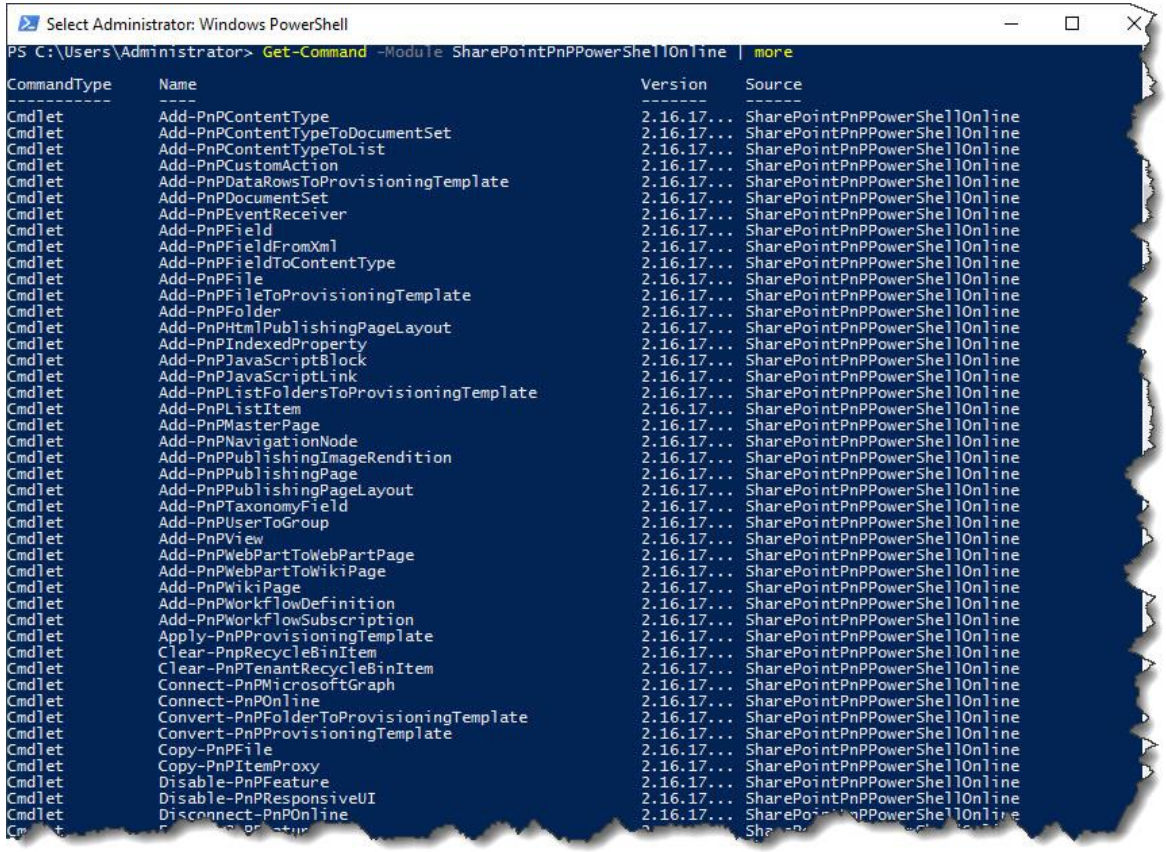
Get-SPOList.ps1

```powershell
$rootWeb = $clientContext.Web
$childWebs = $rootWeb.Webs
$clientContext.Load($rootWeb)
$clientContext.Load($childWebs)
$clientContext.ExecuteQuery()

function getLists($web)
{
    $lists = $web.Lists
    # $clientContext.Load($web)
    $clientContext.Load($lists)
    $clientContext.ExecuteQuery()
    $lists | select title
}

getLists($rootWeb)
```

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



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

Sign in to PnP Office 365

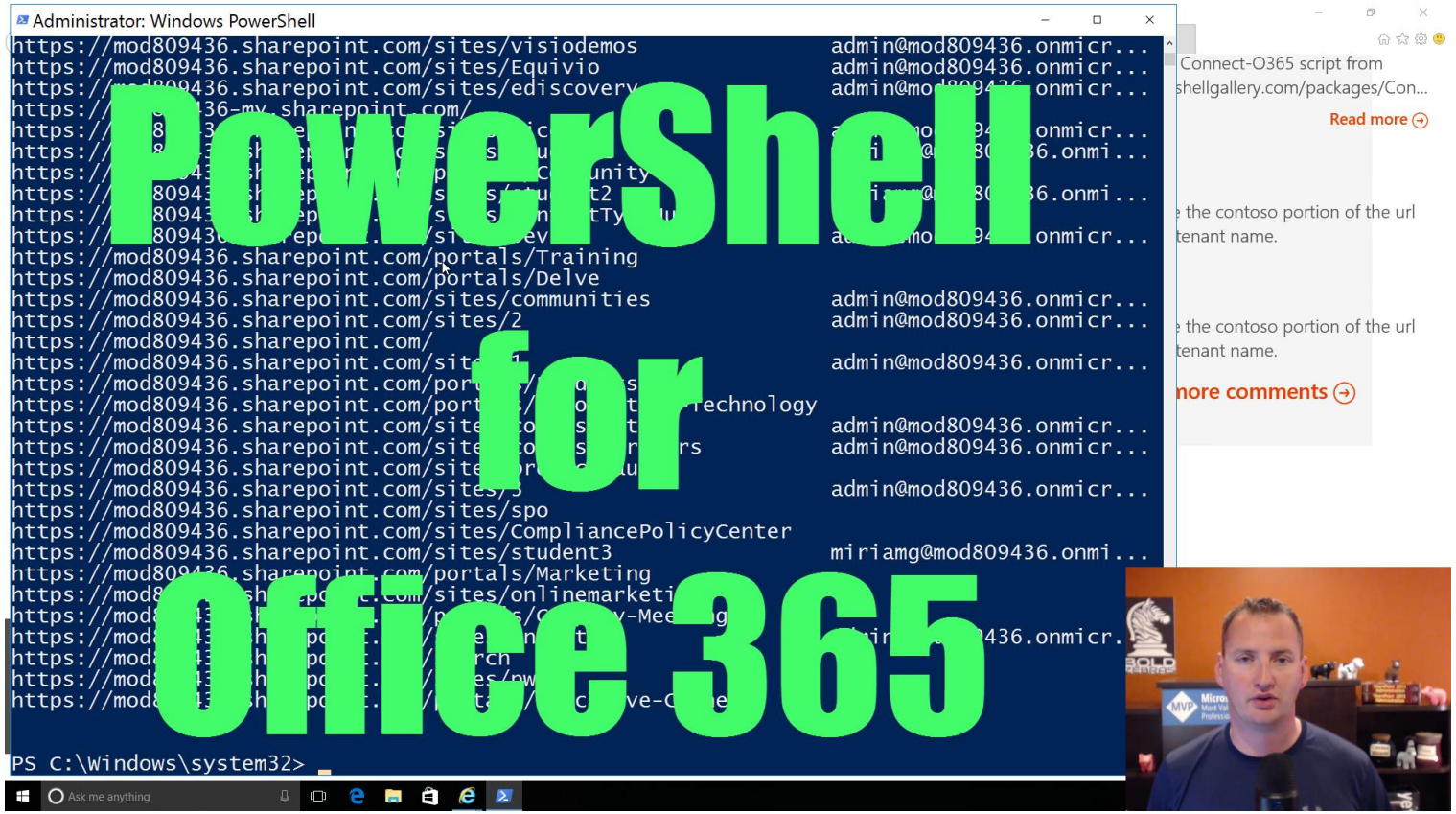
TKC Admin

Windows PowerShell

```
PS C:\> help Get-PnPUnifiedGroupOwners -Examples

NAME
    Get-PnPUnifiedGroupOwners

SYNOPSIS
    * Supported in: SharePoint Online.

    Gets owners of a paricular 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:\>
```

Video that shows you how to do all of that and a bag of chips

- https://youtu.be/rEy2mlFVWa4



- Script to connect to Office 365/Exchange
  - https://eightwone.com/2015/08/31/connecting-to-office-365exchange/

Contact Us

Shane.Young@BoldZebras.com

@ShanesCows

www.PowerApps911.com

todd@toddklindt.com

@toddklindt

www.toddklindt.com

Download the SPTechCon Mobile App

Search for SPTechCon in your App Store and download the 2018 Mobile App to stay connected throughout the entire event.

- **Conference and Session Feedback**
- Get up-to-date show details
- Reference speaker profiles
- Take notes and download presentations
- Connect with other attendees