Office 365 from PowerShell Project

Imran Hakim Roslan
imrh@protonmail.com

January 30, 2021

1 Introduction

This project utilizes AzureAD PowerShell Module to provide the functionality of managing Office 365 tenants from PowerShell. By using cmdlets provided by the module, managing Office 365 can just be done through PowerShell command line interface and gives better approach for IT administrator for managing tenants in large scale.

2 Connecting to Office 365 from PowerShell

2.1 Install AzureAD Module

Connection to Office 365 can be done via integrating AzureAD PowerShell Module in the PowerShell. Steps required for the implementation are discussed below:

1. Install the AzureAD PowerShell Module.

```
PS C:\Users\w102> Install-Module -Name AzureAD
```

Figure 1: Installing AzureAD PowerShell Module

2. Import the AzureAD module.

```
PS C:\Users\w102> Import-Module -Name AzureAD
```

Figure 2: Import AzureAD module

2.2 Connecting to Office 365

After AzureAD Powershell Module has been loaded in PowerShell, the connection now can be made with the following steps:

```
PS C:\Users\w102> # 1. ask for credential
PS C:\Users\w102> $cred= Get-Credential

cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:
Credential
PS C:\Users\w102> # 2. Connect to AzureAD
PS C:\Users\w102> Connect-AzureAD -Credential $cred

Account           Environment TenantId                             TenantDomain AccountType
-------           ----------- --------                             ------------ -----------
m4120@vip365.link AzureCloud  8cdccc1c-8ead-4d4c-a425-10310af89bf2 loffice.ink  User
```

Figure 3: Create Connection to AzureAD

2.3 Tips for Better Approach

A better and faster approach connecting to Office 365 is by saving credential inside an encrypted xml document. This can be done via:

```
PS C:\Users\w102> Get-Credential | Export-Clixml C:\Scripts\azuread_cred.xml

cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:
Credential
```

Figure 4: Save AzureAD credential in an XML file

By saving password in an XML file, there's no need to re-input credentials when running a PowerShell script. Since the password is encrypted, hard-coded password inside a script is no longer needed which is a big plus.

3 Managing Office 365 Users

3.1 Managing Users

Several useful cmdlets can be scripted to view and manage users in tenant (Figure 5).

3.2 Create New Office 365 User

Creation of new Office 365 user can be done via the following (Figure 6):

```powershell
# View all users in tenant
Get-AzureADUser

# View specific user with its properties
Get-AzureADUser -ObjectId m4120@vip365.link | Format-List

# Modify User Properties
Set-AzureADUser `
-ObjectId m4120@vip365.link `
-City "Kuala Lumpur" `
-Country "Malaysia" `
-JobTitle "IT Manager" `
-Department "IT"
```

Figure 5: Example managing tenant users

```powershell
# 1. Create password object
$PasswordProfile= New-Object `
-TypeName Microsoft.Open.AzureAD.Model.PasswordProfile
$PasswordProfile.Password= "testPasswOrd"
$PasswordProfile.ForceChangePasswordNextLogin= $true

# 2. Create User
New-AzureADUser `
-GivenName "John" `
-Surname "Terry" `
-DisplayName "John TERRY" `
-UserPrincipalName "johnTERRY@vip365.link" `
-MailNickname "johnTERRY" `
-AccountEnabled $true `
-PasswordProfile $PasswordProfile `
-JobTitle "IT Help Desk" `
-Department "IT"
```

Figure 6: Create new user in Office 365 tenant

3.3 Managing User Licenses

Examples of managing Office 365 user licenses in PowerShell are presented in Figure 7.

4 Office 365 SharePoint

4.1 View SharePoint sites in Tenant

The following PowerShell code snippet in Figure 8 illustrate method of view all available SharePoint page in the tenant.

```powershell
<# 1. View all subscribed licenses in tenant with their
current availability #>
Get-AzureADSubscribedSku |
Select-Object `
-Property ObjectId, SkuPartNumber, ConsumedUnits `
-ExpandProperty PrepaidUnits

# 2. Assigning license to a user

# 2.1 Defining parameters
$user= Get-AzureADUser -ObjectId john@link123
$sku= New-Object -TypeName Microsoft.Open.AzureAD.Model.AssignedLicense
$license= New-Object -TypeName Microsoft.Open.AzureAD.Model.AssignedLicense

# 2.2 Get license SkuID value
Get-AzureADSubscribedSku | Select-Object -Property SkuPartNumber, SkuID

# 2.3 Attach the license attribute to parameters
$sku.SkuId= "c7df2760-2c81-4ef7-b578-5b5392b571df" # example SkuID
$licenses.AddLicenses= $sku

# 2.4 Assigned user the license
Set-AzureADUserLicense -ObjectId $user.ObjectId -AssignedLicense $licenses
```

Figure 7: Example assigning user a license

```powershell
Get-SPOSite | Select Url, Title, Template
```

Figure 8: View available SharePoint site

4.2 Create a SharePoint Site with PowerShell

A new SharePoint site can be made easily in PowerShell with preceding code example in Figure 9.

```powershell
New-SPOSite `
-Url https://office365powershell.sharepoint.com/teams/IT `
-Owner vlink-admin365@office365powershell.com `
-Storagequota 1024 `
-LocaleID 1033 `
-Template "STS#0" `
-Title "IT Team Site"
```

Figure 9: Create SharePoint site with PowerShell

4.3 Viewing Groups and Users of SharePoint Site

Viewing all groups and users of a specific SharePoint site can be done as follows:

```
$site= Get-SPOSite `
-Identity https://office365powershell.sharepoint.com/teams/IT
$groups= Get-SPOSiteGroup -Site $site
foreach ($group in $groups) {
    Write-Host $group.Title -ForegroundColor "Blue"
    Get-SPOSiteGroup -Site $site -Group $group.Title |
    Select -ExpandProperty Users
    Write-Host  # output empty line '\n'
}
```

Figure 10: List groups and users of a SharePoint site