PowerShell Practical Assignment

Kartik Velede

8672807

Collaborative Environment Processes 1 – INFO8930

Doug Ferrier

June-16-2020

June-16-2020

**Table of Contents**

[1. Connecting to an Office 365 instance with PowerShell 3](#_Toc43213776)

[2. Connecting to Exchange Online and SharePoint Online with PowerShell 7](#_Toc43213777)

[3. Getting a list of available Office 365 PowerShell cmdlet 10](#_Toc43213778)

[4.Getting a list of all Office 365 users with PowerShell 12](#_Toc43213779)

[5.Creating a new user in Office 365 with PowerShell 14](#_Toc43213780)

[6. Changing a password in Office 365 with PowerShell 14](#_Toc43213781)

[7.Managing group membership in Office 365 with PowerShell 15](#_Toc43213782)

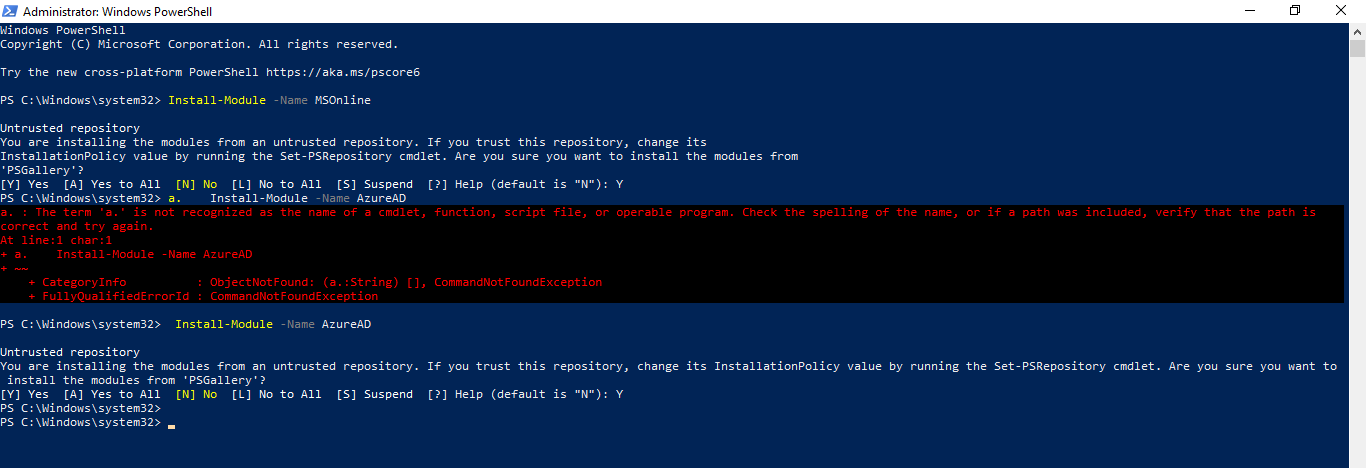
[8.Creating a SharePoint site collection with PowerShell 16](#_Toc43213783)

[9.Creating reports in Office 365 with PowerShell 17](#_Toc43213784)

# 1. Connecting to an Office 365 instance with PowerShell

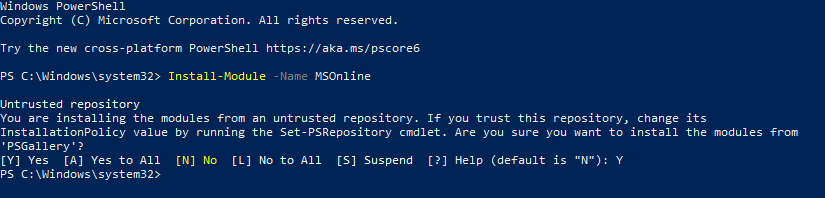
1. Install the Microsoft Online Services Sign-In Assistant for IT Professionals RTW
2. Install the SharePoint Online Management Shell
3. Install the Online Services PowerShell module for Microsoft Azure Active Directory and O365:
4. Install-Module -Name AzureAD

Here we are installing the AzureAd with the above Powershell command and the execution were successful.



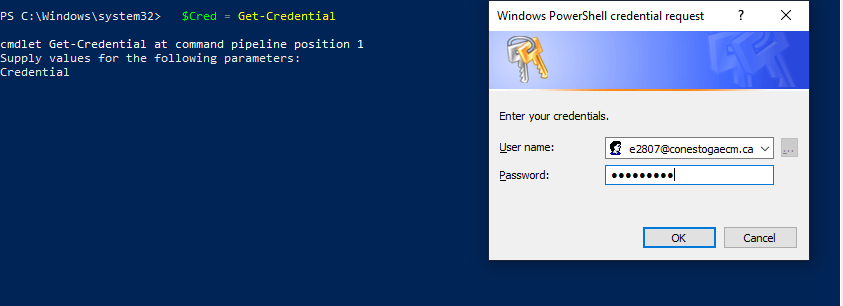
1. Install-Module -Name MSOnline

Here we are installing the MSOnline with the above Powershell Command the execution were successful.



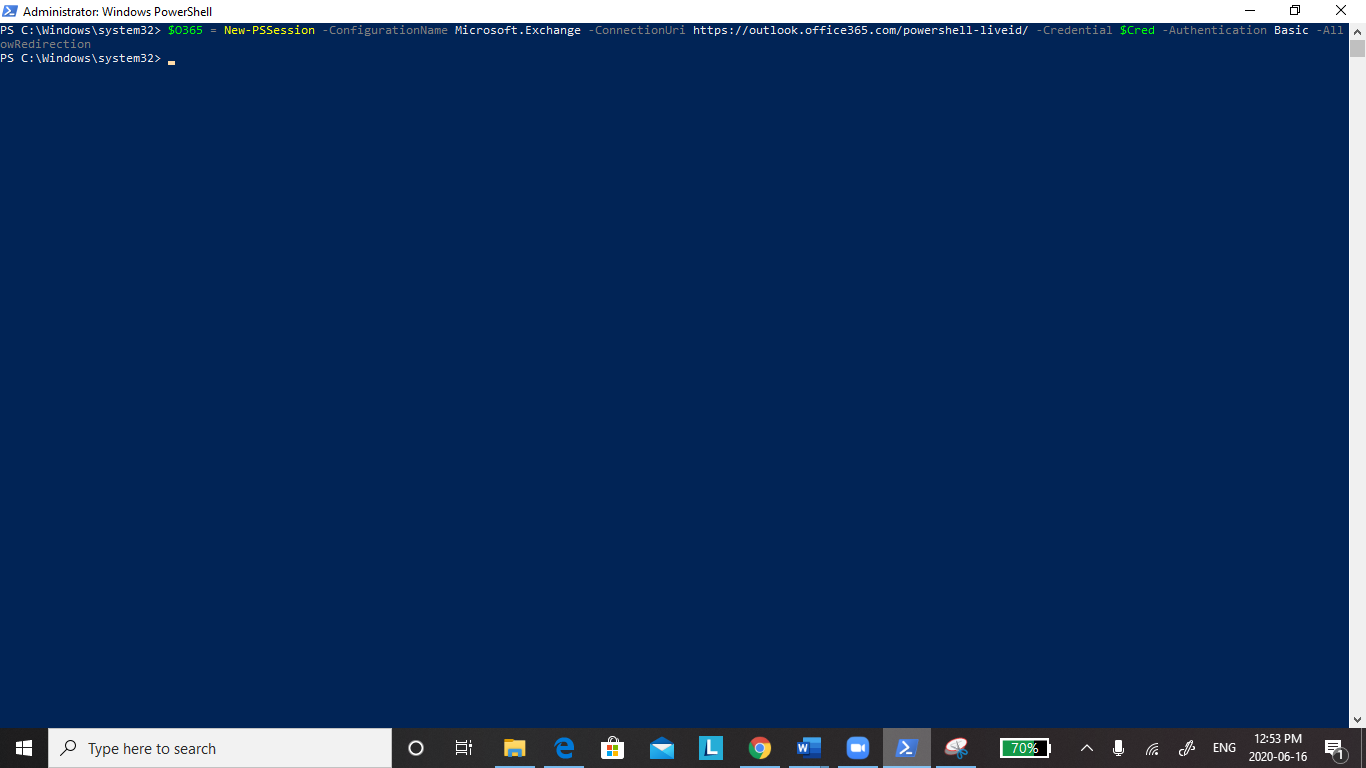
1. Enter your Office 365 admin credentials:
   1. $Cred = Get-Credential

Th above command is used to enter the username and password of the Office 365 admin and the execution of the command were successful.

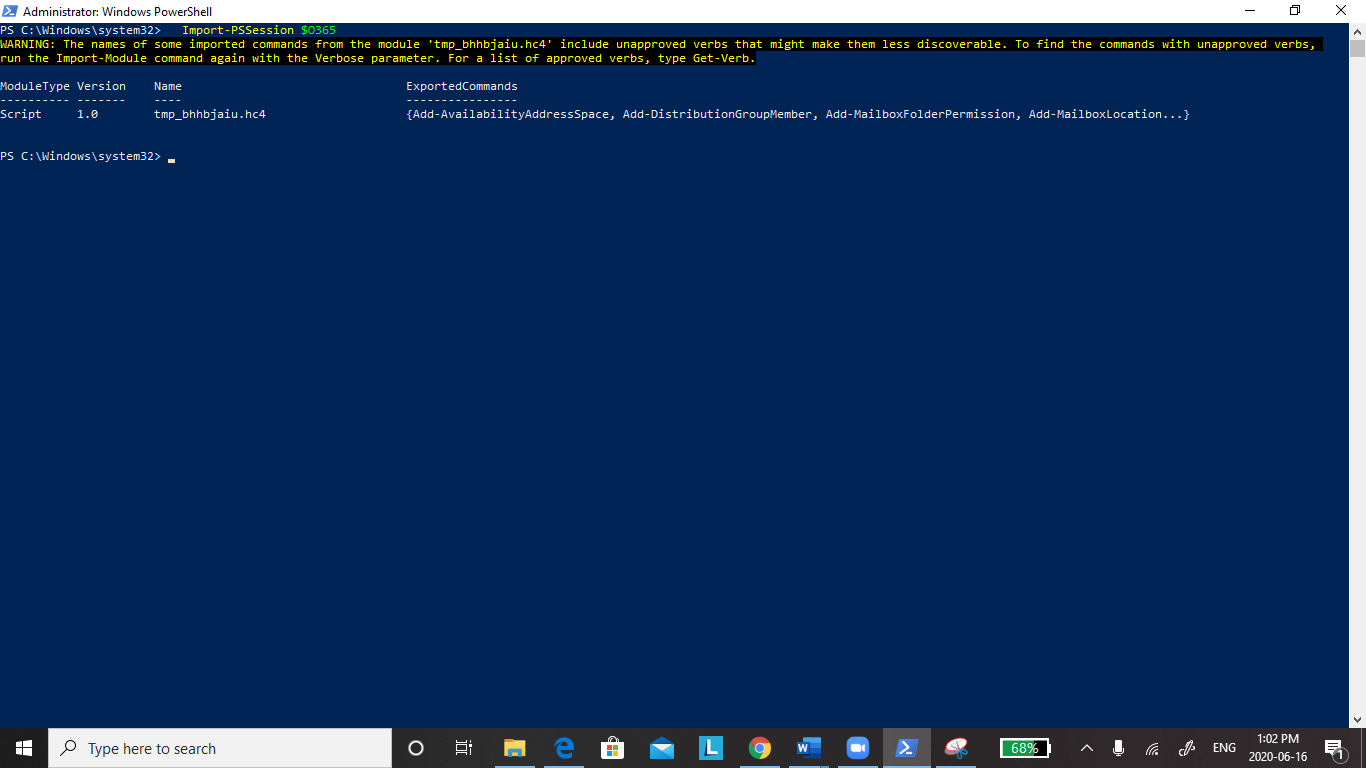


1. Create a remote PowerShell session:
   1. $O365 = New-PSSession-ConfigurationName Microsoft.Exchange -ConnectionUri https://outlook.office365.com/powershell-liveid/ -Credential $Cred -Authentication Basic -AllowRedirection

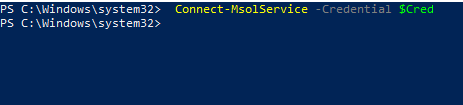
Once the Credential are successfully implemented, here we are creating remote PowerShell session with the above command and the execution were successful.



1. Import the session commands into the local Windows PowerShell session:
   1. **Import**-PSSession $O365

After creating the remote session, the above command is used to import the commands needed in the session into the local window PowerShell Session and the execution of the command were successful.

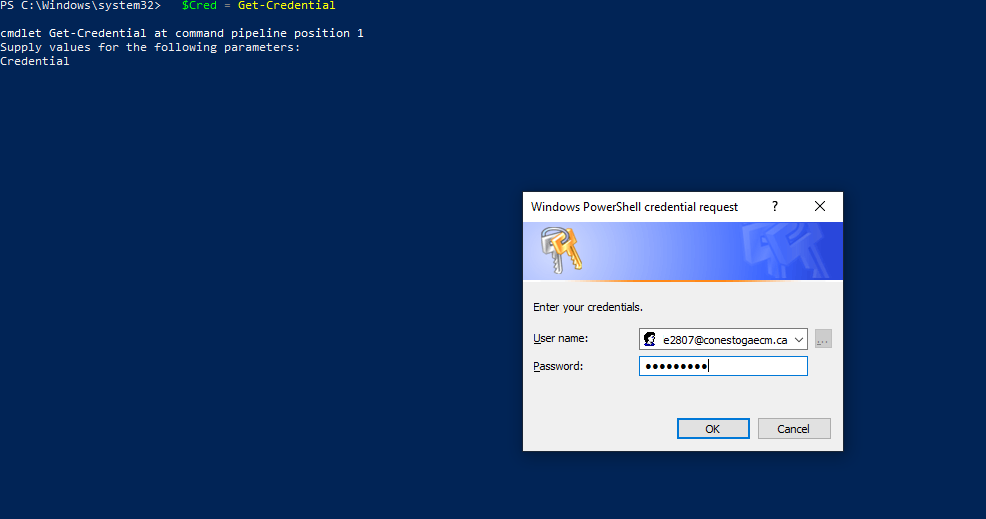
1. Connect to all Office 365 services:
   1. Connect-MsolService –Credential $O365



# 2. Connecting to Exchange Online and SharePoint Online with PowerShell

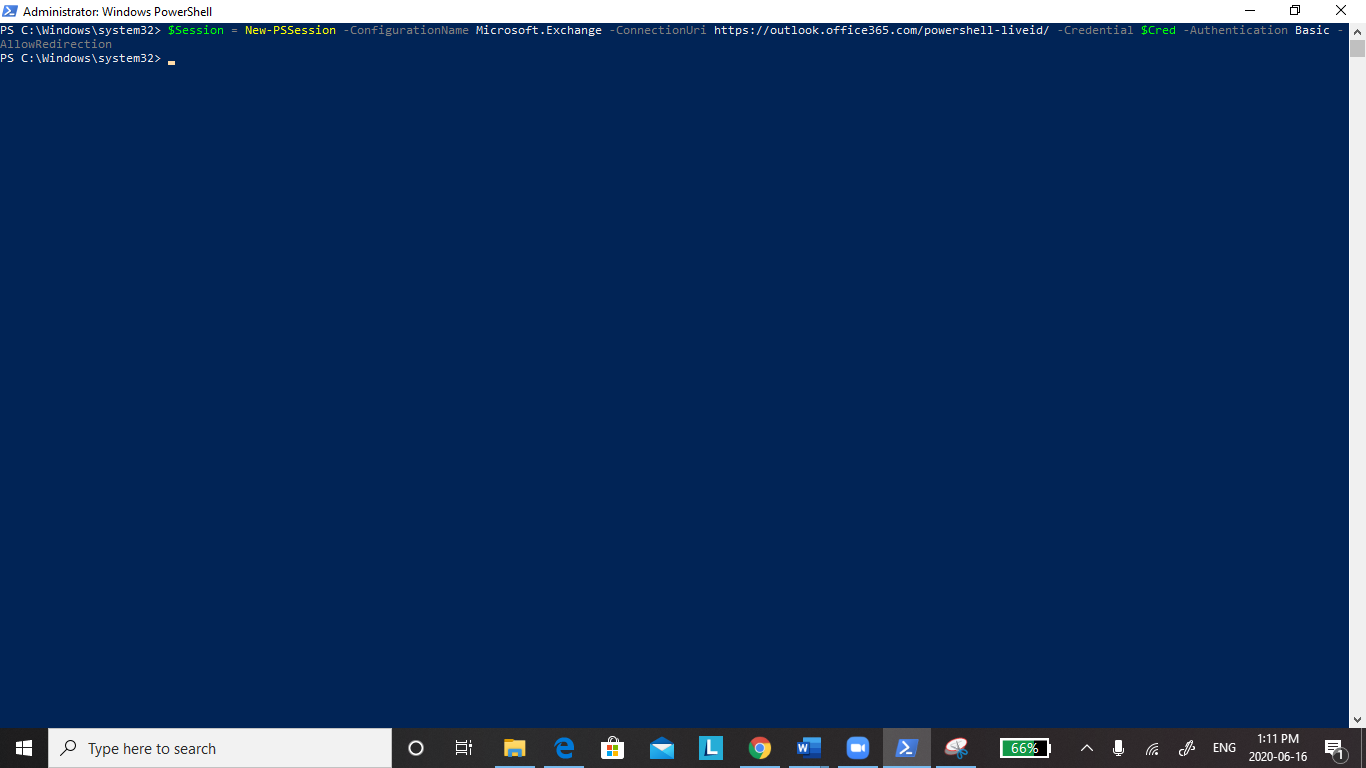
1. Connect to Exchange Online
   1. $Cred = Get-Credential

Here we are connecting to exchange online and Share Online by giving the Office 365 Credentials and the execution were successful.



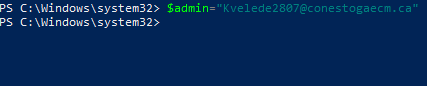
* 1. $Session = New-PSSession-ConfigurationName Microsoft.Exchange -ConnectionUri https://outlook.office365.com/powershell-liveid/ -Credential $Cred -Authentication Basic –AllowRedirection

Here we are creating a new session to execute the PowerShell commands and the above execution were successful.



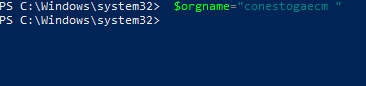
1. Connect to SharePoint Online
   1. $admin=[Kvelede2807@conestogaecm.ca](mailto:Kvelede2807@conestogaecm.ca)

The above command is used to enter the admin name and the execution of the command were successful.



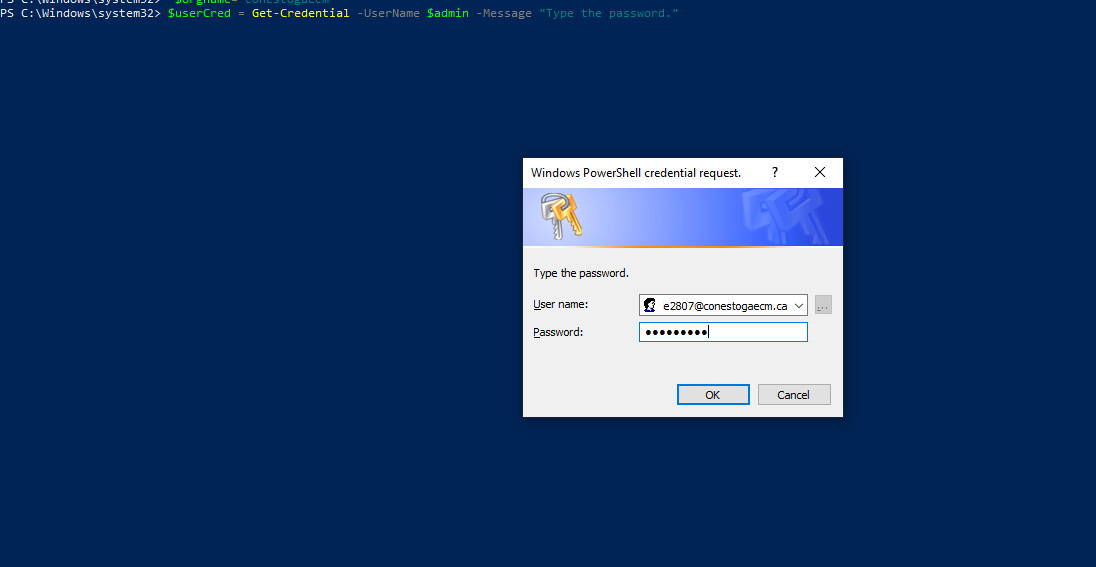
* 1. $orgname="conestogaecm”

The above command is used to enter the organization name and the execution of the command were successful.



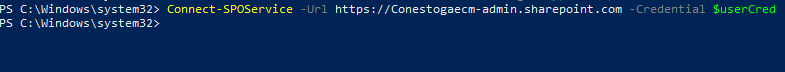
* 1. $userCred = Get-Credential -UserName $admin -Message "Type the password."

Here the above command is used to connect to the organization by entering the username and password and the execution were successful.



* 1. Connect-SPOService -Url https://Conestogaecm-admin.sharepoint.com -Credential $userCred

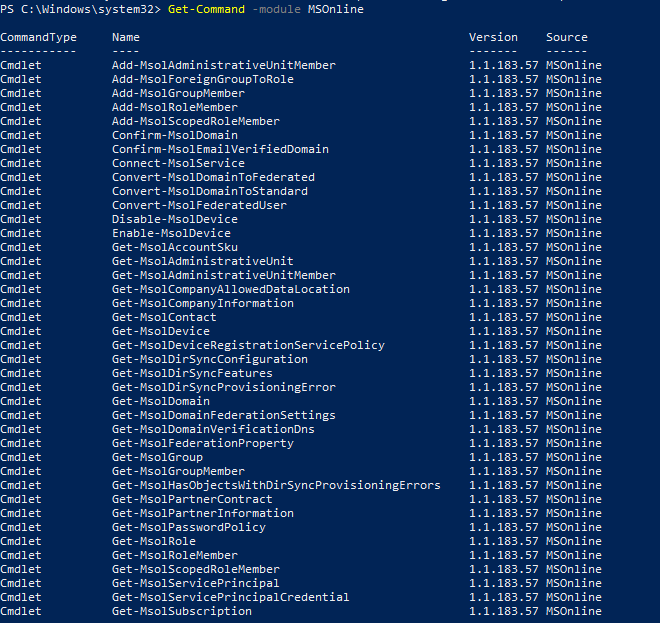
The above command is used to connect to the server and the execution of the code were successful.



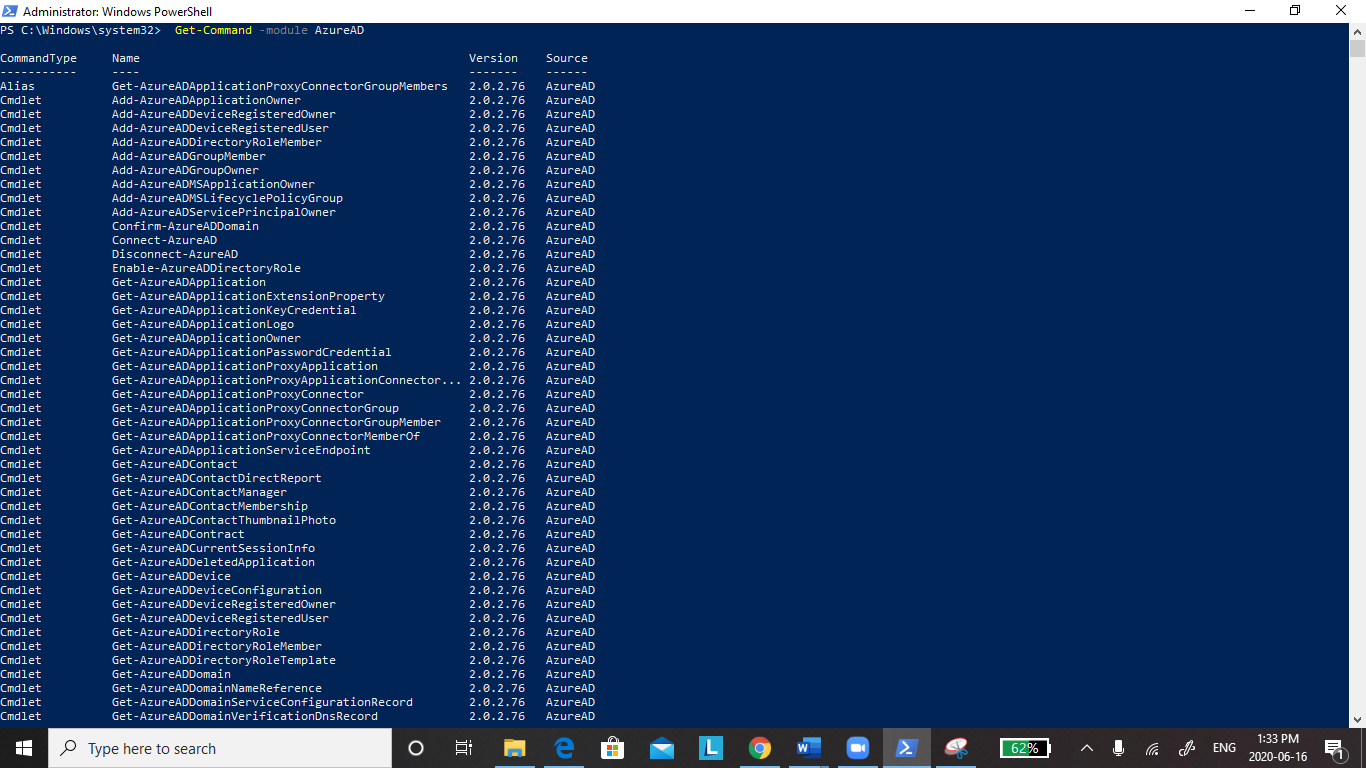
# 3. Getting a list of available Office 365 PowerShell cmdlet

1. Get a list of all available Office 365 PowerShell commands
   1. Get-Command -module MSOnline

The above command is used to get the list of command used for the Module MSOnline in PowerShell and the execution of the code were successful.



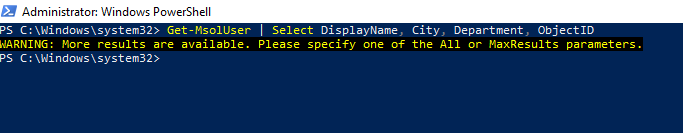
1. Get the list of cmdlets for Azure Active Directory
   1. Get-Command -module AzureAD

The above command is used to get the list of command used for the Module AzureAD in PowerShell and the execution of the code were successful. 

# 4.Getting a list of all Office 365 users with PowerShell

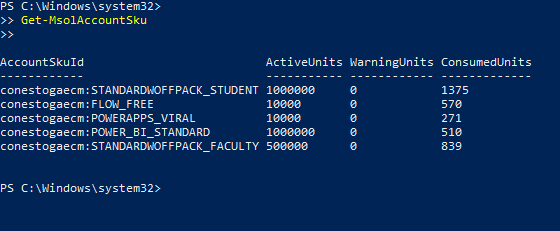
1. If you need to provide a list of Office 365 users and licenses, use the Get-MsolUser cmdlet. It will retrieve all users with a valid license in the Office 365 tenant, along with the DisplayName, City, Department and ObjectID parameters.
   1. Get-MsolUser | Select DisplayName, City, Department, ObjectID

The above command is used to get data about the users and the execution were unsuccessful.



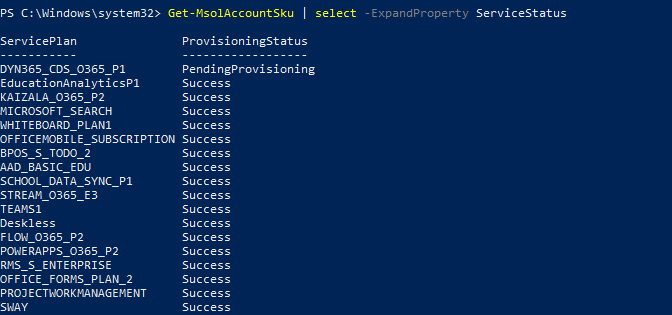
1. To Get the number of account licenses, you need to run the following cmdlet:
   1. Get-MsolAccountSku

Here we used the above command to get the licensed accounts and the execution were successful.



1. List the available services
   1. Get-MsolAccountSku | select -ExpandProperty ServiceStatus

The above command is used to list all the available services along with their parameter and the execution were successful.

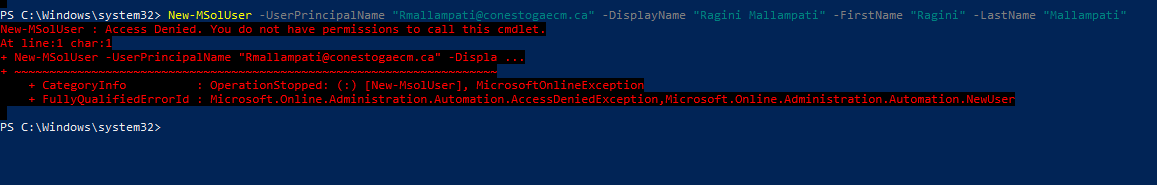


# 5.Creating a new user in Office 365 with PowerShell

1. Create a new user

a. New-MSolUser -UserPrincipalName "Rmallampati@conestogaecm.ca" -DisplayName "Ragini Mallampati" -FirstName “Ragini" -LastName "Mallampati"

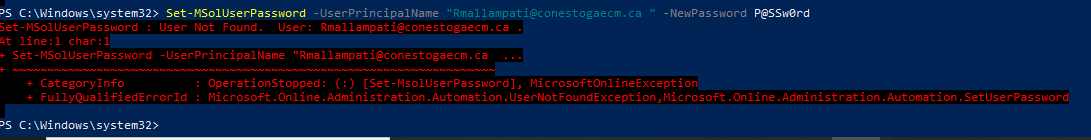
To add the new user into the service we use the above command in which we give user name, display name, first name, last name and the command execution were unsuccessful.



# 6. Changing a password in Office 365 with PowerShell

A. The below command is used to set the password for the user added in the above step and the command execution were unsuccessful.

* 1. Set-**MSolUserPassword** -UserPrincipalName "Rmallampati@conestogaecm.ca " -NewPassword P@SSw0rd

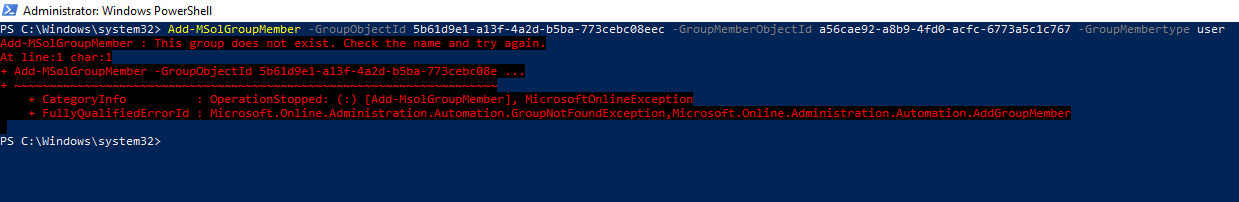


# 7.Managing group membership in Office 365 with PowerShell

We can also manage Office 365 groups using PowerShell cmdlets. To retrieve a list of all groups in Office 365, simply use the command Get-MsolGroup

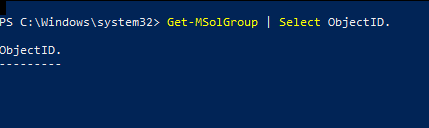
1. To add users to a group, use the Add-MsolGroupMember command
   1. Add-MsolGroupMember -GroupObjectId 5b61d9e1-a13f-4a2d-b5ba-773cebc08eec -GroupMemberObjectId a56cae92-a8b9-4fd0-acfc-6773a5c1c767 -GroupMembertype user

The above command is used to add the users to groups and the execution of the command were unsuccessful (Error Message: Group already exists)



1. GroupObjectId is the hexadecimal ID of the group, which you can get from the Get-MsolGroup command. GroupMemberObejctId is the user object ID, which you can find by running this command
   1. **Get**-MSolGroup | Select ObjectID.

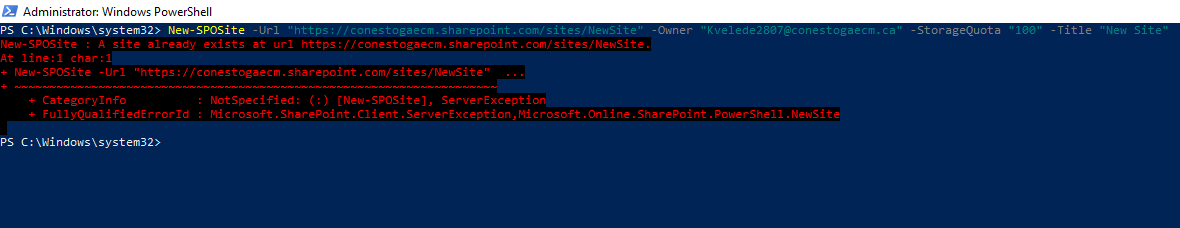
Every user will have a unique object id and the above command is used to get the objected of the user and the command execution were successful.



# 8.Creating a SharePoint site collection with PowerShell

1. Create a SharePoint site collection
   1. New-SPOSite -Url "https://conestogaecm.sharepoint.com/sites/NewSite" -Owner "Kvelede2807@conestogaecm.ca" -StorageQuota "100" -Title "New Site"

The above command is used to create a new SharePoint site collection and the command execution were unsuccessful (Error message: Site already exists)

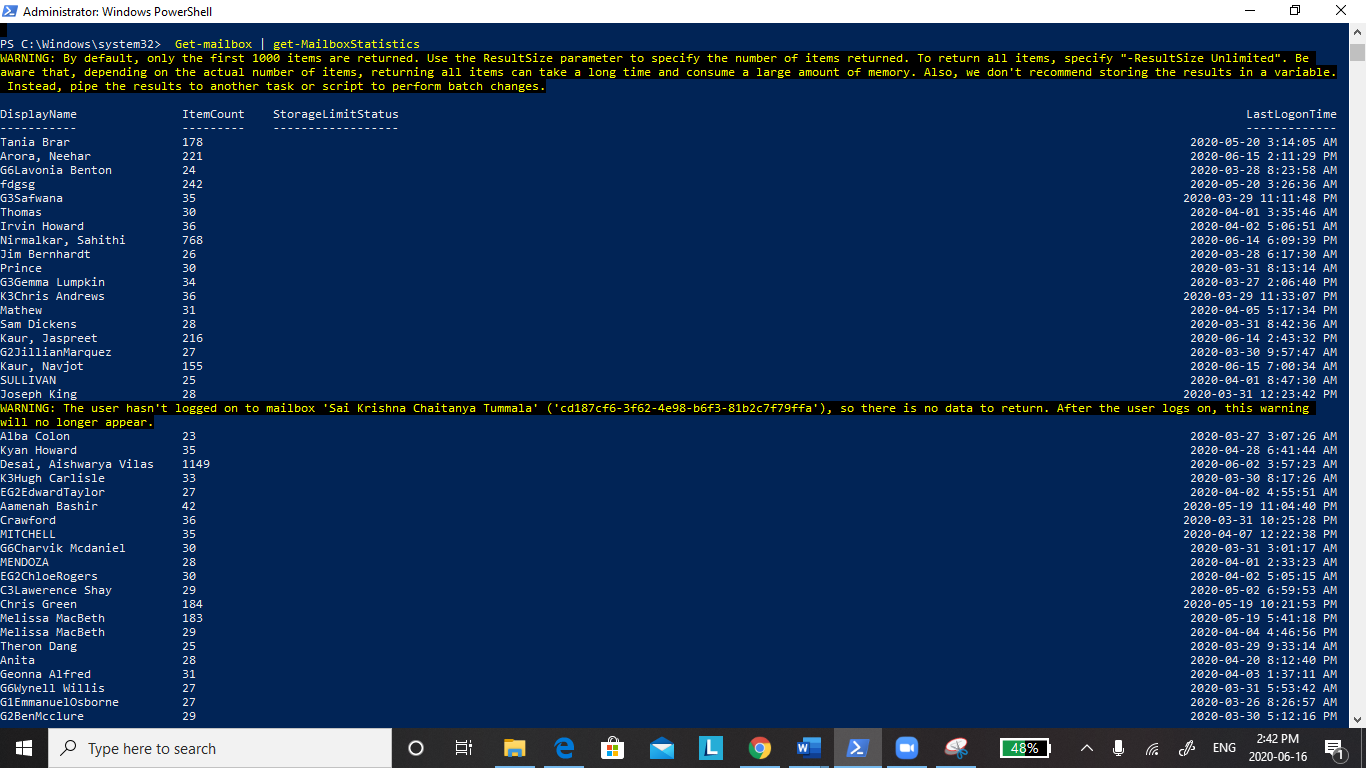


# 9.Creating reports in Office 365 with PowerShell

PowerShell is a great tool for making different reports. Here are some useful Office 365 reports done via PowerShell:

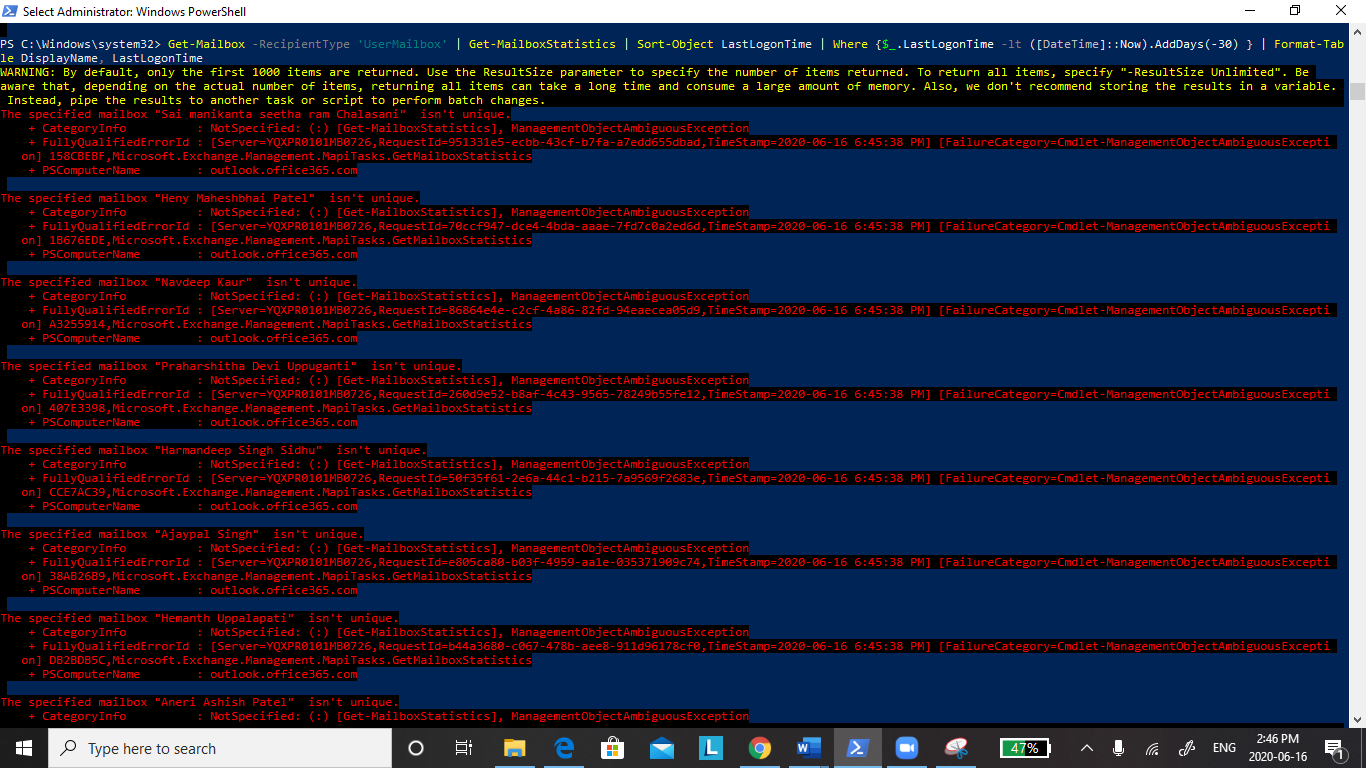
1. Get details about all mailboxes
   1. Get-mailbox | get-MailboxStatistics

The above command is used to get the details of mailboxes and the command execution were successful.



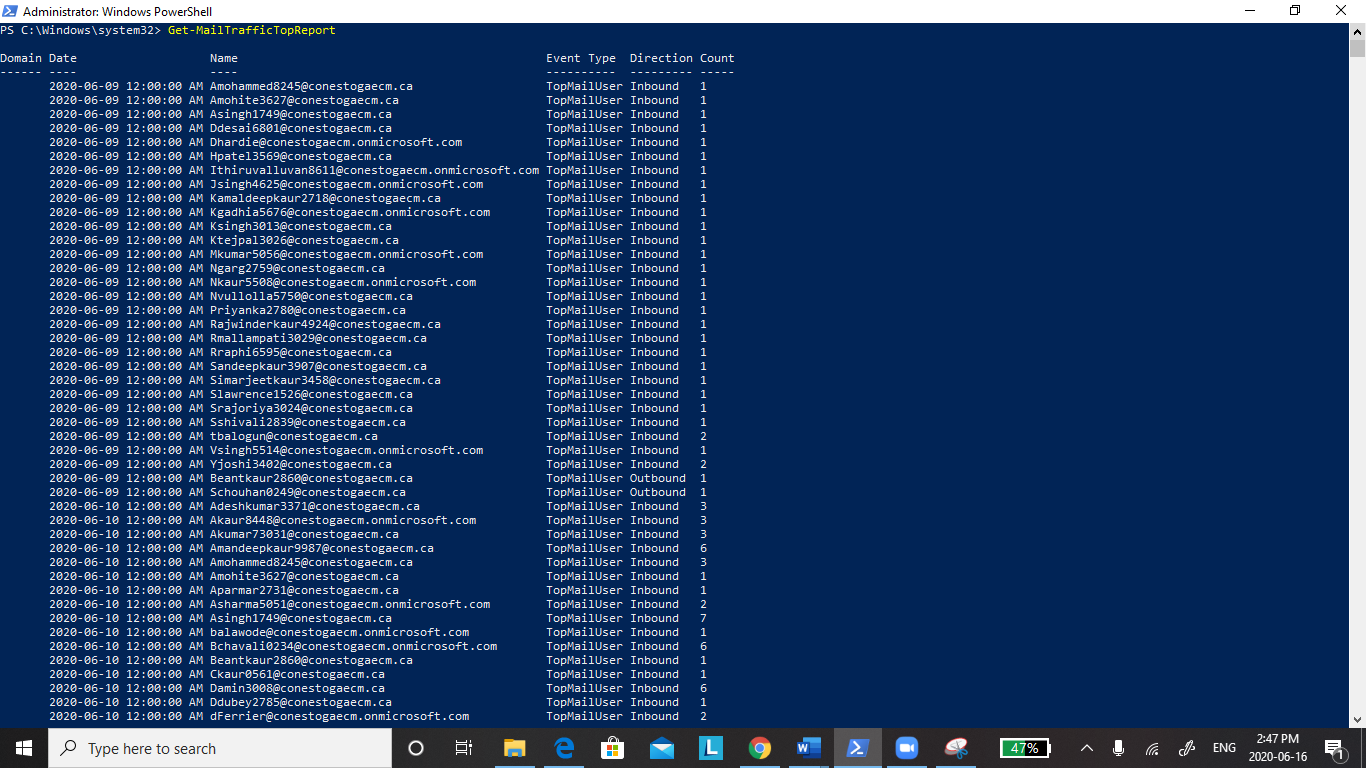
1. A list of all mailboxes that have not been logged into during the last 30 days
   1. Get-Mailbox –RecipientType 'UserMailbox' | Get-MailboxStatistics | Sort-Object LastLogonTime | Where {$\_.LastLogonTime –lt ([DateTime]::Now).AddDays(-30) } | Format-Table DisplayName, LastLogonTime

The above command is used to display the user who are inactive from last 30 days and the command execution were successful.



1. A report on the highest volume senders and recipient
   1. Get-MailTrafficTopReport

The above command execution was successful and provided data about the user who has highest volume of mail traffic.



1. A report on all groups and their members
   1. function Get-AllO365Members

{

Try

{

$O365Groups=Get-UnifiedGroup

foreach ($O365Group in $O365Groups)

{

Write-Host "Group Membership: " $O365Group.DisplayName -ForegroundColor Green

Get-UnifiedGroupLinks –Identity $O365Group.Identity –LinkType Members

Write-Host

}

}

catch [System.Exception]

{

Write-Host -ForegroundColor Red $\_.Exception.ToString()

}

}

Get-AllO365Members

The above command gives information about all the users ,groups and the command execution were successful.

