



# PowerShell for Azure și Microsoft 365

Presented by BOGDAN BADEA



## Lessons:

- Lesson 1: Azure PowerShell
- Lesson 2: Introduce Azure Cloud Shell
- Lesson 3: Manage Azure VMs with PowerShell
- Lesson 4: Manage storage and subscriptions
- Lesson 5: Manage Microsoft 365 user accounts, licenses, and groups with PowerShell
- Lesson 6: Manage Exchange Online with PowerShell
- Lesson 7: Manage SharePoint Online with PowerShell
- Lesson 8: Manage Microsoft Teams with PowerShell



**Center for Career Development**

*by LINKgroup*

# Lesson 1: Azure PowerShell

Topics include:

- Azure PowerShell overview
- What is the Azure Az PowerShell module?
- Installing the Azure Az PowerShell module
- Migrate Azure PowerShell from AzureRM to Az
- What are the Microsoft Azure Active Directory Module for Windows PowerShell and Azure Active Directory PowerShell for Graph modules?

## Azure PowerShell overview

- Azure PowerShell is a set of cmdlets for managing Azure resources directly from the PowerShell command line.
- It's a module that you add to Windows PowerShell or PowerShell Core.
- Azure PowerShell works with PowerShell 5.1 on Windows, PowerShell 7.0.6 LTS, and PowerShell 7.1.3 or newer on all platforms.
- You can use Azure PowerShell in your browser with Azure Cloud Shell, or install it on your local machine on Windows, MacOS, or Linux.

## What is the Azure Az PowerShell module?

- The Az PowerShell module is a set of cmdlets for managing Azure resources directly from PowerShell.
- The Az PowerShell module is the replacement for AzureRM and is the recommended version to use for interacting with Azure.
- The Az PowerShell module features:
  - Security and stability.
  - Support for all Azure services.
  - New capabilities.
- You can log issues or feature requests for the Az module directly on the GitHub repository, or through Microsoft support if you have a support contract.

## Installing the Azure Az PowerShell module

- You can install the Azure Az PowerShell module by using one of the following methods:
  - The **Install-Module** cmdlet
  - Azure PowerShell MSI
  - Az PowerShell Docker container
- To install the Az module, run the following command:  
**Install-Module -Name Az -Scope CurrentUser -Repository PSGallery -Force**
- To sign in to Azure from Azure PowerShell, run the following command:  
**Connect-AzAccount**



## What are the Microsoft Azure Active Directory Module for Windows PowerShell and Azure Active Directory PowerShell for Graph modules?

- You can use the Azure Active Directory Module for Windows PowerShell cmdlets for Azure AD administrative tasks such as user management, domain management, and configuring single sign-on.
- Microsoft is replacing the Azure Active Directory Module for Windows PowerShell with Active Directory PowerShell for Graph.
- To install Azure Active Directory Module for Windows PowerShell, use the following command:  
**Install-Module MSOnline**
- To install Azure Active Directory PowerShell for Graph, use the following command:  
**Install-Module AzureAD or Install-module AzureADPreview**



## Lesson 2: Introduce Azure Cloud Shell



# **Center for Career Development**

*by LINKgroup*

Topics include:

- Cloud Shell overview
- Features and tools for Azure Cloud Shell
- Demonstration: Use Cloud Shell



## Cloud Shell overview

- Azure Cloud Shell is an interactive, browser-accessible shell for managing Azure resources.
- You can use Cloud Shell to work untethered from a local machine in a way only the cloud can provide.
- You can access the Cloud Shell in three ways:
  - Direct link. Open a browser and refer to **<https://shell.azure.com>**.
  - Azure portal. Select the Cloud Shell icon on the Azure portal.
  - Code snippets.
- After first launch, you can use the shell type drop-down control to switch between **Bash** and **PowerShell**.

## Features and tools for Azure Cloud Shell

- The main features of Azure Cloud Shell are:
  - Secure automatic authentication.
  - \$HOME persistence across sessions.
  - Azure drive (Azure:).
  - Ability to manage Exchange Online.
  - Deep integration with open-source tooling.



**Center for Career Development**

*by LINKgroup*

# Lesson 3: Manage Azure VMs with Powershell



# **Center for Career Development**

*by LINKgroup*

## Topics:

- Creating Azure VMs with Windows PowerShell
- Managing Azure VMs with Windows PowerShell



## Creating Azure VMs with Windows Powershell

- To create a new Azure VM with PowerShell commands, you can use your locally installed Windows PowerShell with Az module or the Cloud Shell environment available in the Azure portal.
- When using locally installed PowerShell, you first need to use the **Connect-AzAccount** command to authenticate.
- To create and use a new Azure VM, you should:
  - Create a resource group.
  - Run the **New-AzVM** command with the configuration parameters.
  - Find the public IP address.
  - Connect to the Azure VM.

## Managing Azure VMs with Windows Powershell

- The most common management tasks for Azure VMs are:
  - Modifying VM size profile with the **Get-AzVMSize** and **Update-AzVM** commands.
  - Starting and stopping the VM with the **Start-AzVM** and **Stop-AzVM** commands.
  - Adding or modifying disks to the Azure VM.





**Center for Career Development**

*by LINKgroup*

## Lesson 4: Manage storage and subscriptions



# **Center for Career Development**

*by LINKgroup*

## Topics:

- Managing storage with Azure PowerShell
- Managing Azure subscriptions with Azure PowerShell



## Managing storage with Windows Powershell

- Azure PowerShell allows you to manage Azure-related storage.
- Before you start managing your storage, you should first create a storage account by using **New-AzStorageAccount**.
- You can use the **Set-AzStorageAccount** cmdlet to modify an Azure Storage account.



## Managing Azure subscriptions with Windows Powershell

- In Azure PowerShell, accessing the resources for a subscription requires changing the subscription associated with your current Azure session.
- To change subscriptions, you need to first retrieve an Azure PowerShell Context object with **Get-AzSubscription** and then change the current context with **Set-AzContext**.



# Lesson 5: Manage Microsoft 365 user accounts, licenses and groups with Powershell



## Topics:

- Benefits of using PowerShell for Microsoft 365
- Connecting to the Microsoft 365 tenant with PowerShell
- Managing users in Microsoft 365 with PowerShell
- Managing groups in Microsoft 365 with PowerShell
- Managing roles in Microsoft 365 with PowerShell
- Managing licenses in Microsoft 365 with PowerShell

## Benefits of using Powershell for Microsoft 365

- PowerShell is an alternative to web-based consoles
- PowerShell has the following benefits over the web-based consoles:
  - Access to additional configuration options
  - Query for objects matching criteria and generate reports
  - Use of the pipeline to perform complex operations
  - Automation of bulk processes
  - Management of multiple services simultaneously

## Connecting to the Microsoft 365 tenant with Powershell

- There are multiple modules for connecting to Azure AD:
  - AzureAD
  - MSOnline
  - Microsoft.Graph
- Use the cmdlet **Install-Module** to install these modules from the PowerShell Gallery
- AzureAD and MSOnline have similar functionality but aren't exactly the same
- Microsoft.Graph is the focus of future development
- Azure Cloud Shell is a web-based prompt with modules automatically loaded





## Managing users in Microsoft 365 with PowerShell (Slide 1)

- To create a user with AzureAD:
  - **`$UserPassword=New-Object -TypeName Microsoft.Open.AzureAD.Model.PasswordProfile`**
  - **`$UserPassword.Password="Pa55w.rd"`**
  - **`New-AzureADUser -DisplayName "Abbie Parsons" -GivenName "Abbie" -SurName "Parsons" -UserPrincipalName AbbieP@adatum.com -UsageLocation US -PasswordProfile $UserPassword -AccountEnabled $true`**
- Other AzureAD cmdlets for managing users:
  - **`Get-AzureADUser`**
  - **`Set-AzureADUser`**
  - **`Remove-AzureADUser`**
  - **`Set-AzureADUserPassword`**
  - **`Get-AzureADRMSTeletedDirectoryObject`**



## Managing users in Microsoft 365 with PowerShell (Slide 2)

- To create a user with MSOnline:
  - **New-MsolUser -DisplayName "Abbie Parsons" -FirstName "Abbie" -LastName "Parsons" -UserPrincipalName AbbieP@adatum.com -Password "Pa55w.rd"**
- Other MSOnline cmdlets for managing users:
  - **Get-MsolUser**
  - **Set-MsolUser**
  - **Remove-MsolUser**
  - **Set-MsolUserPassword**
  - **Set-MsolUserPrincipalName**
  - **Restore-MsolUser**



## Managing groups in Microsoft 365 with PowerShell

- There are multiple group types in Microsoft 365:
  - Microsoft 365 group
  - Distribution group
  - Security group
  - Mail-enabled security group
- Microsoft 365 groups are managed in Exchange Online as unified groups



## Managing groups in Microsoft 365 with PowerShell (Slide 2)

- To create a group by using AzureAD:
  - **New-AzureADGroup -DisplayName "Marketing Group" -MailEnabled \$true -SecurityEnabled \$true -MailNickname MarketingGrpDistribution group**
- To manage groups, you need to provide the ObjectID as a unique identifier
- Other AzureAD cmdlets for managing groups:
  - **Get-AzureADGroup**
  - **Set-AzureADGroup**
  - **Remove-AzureADGroup**
  - **Get-AzureADGroupMember**
  - **Add-AzureADGroupMember**
  - **Remove-AzureADGroupMember**
  - **Get-AzureADGroupOwner**



## Managing groups in Microsoft 365 with PowerShell (Slide 3)

- To create a group by using MSOnline:
  - **New-MsolGroup -DisplayName "Marketing Group"**
- Other AzureAD cmdlets for managing groups:
  - **Get-MsolGroup**
  - **Set-MsolGroup**
  - **Remove-MsolGroup**
  - **Get-MsolADGroupMember**
  - **Add-MsolADGroupMember**
  - **Remove-MsolADGroupMember**



## Managing roles in Microsoft 365 with PowerShell

- To enable a role by using AzureAD:
  - **`$roleTemplate = Get-AzureADDirectoryRoleTemplate | Where {$_.displayName -eq 'User Administrator'}`**
  - **`Enable-AzureADDirectoryRole -RoleTemplateId $roleTemplate.ObjectId`**
- To add a user to a role by using AzureAD:
  - **`$user = Get-AzureADUser -ObjectID AbbieP@adatum.com`**
  - **`$role = Get-AzureADDirectoryRole | Where {$_.displayName -eq 'User Administrator'}`**
  - **`Add-AzureADDirectoryRoleMember -ObjectId $role.ObjectId -RefObjectId $user.ObjectId`**
- To add a user to a role by using MSOnline:
  - **`Add-MsolRoleMember -RoleMemberEmailAddress AbbieP@adatum.com -RoleName 'User Administrator'`**



## Managing licenses in Microsoft 365 with PowerShell

- To review available licenses by using AzureAD:
  - **Get-AzureADSubscribedSku | Select-Object -Property Sku\*,ConsumedUnits -ExpandProperty PrepaidUnits**
- To assign licenses to a user by using AzureAD:
  - **\$License = New-Object -TypeName Microsoft.Open.AzureAD.Model.AssignedLicense**
  - **\$License.SkuId = '05e9a617-0261-4cee-bb44-138d3ef5d965'**
  - **\$LicensesToAssign = New-Object -TypeName Microsoft.Open.AzureAD.Model.AssignedLicenses**
  - **\$LicensesToAssign.AddLicenses = \$License**
  - **Set-AzureADUserLicense -ObjectId AbbieP@adatum.com -AssignedLicenses \$LicensesToAssign**
- To disable specific service plans by using AzureAD:
  - **\$License.DisabledPlans = '7547a3fe-08ee-4ccb-b430-5077c5041653'**
  - **\$License.DisabledPlans.Add('a23b959c-7ce8-4e57-9140-b90eb88a9e97')**



## Managing licenses in Microsoft 365 with PowerShell (Slide 2)

- To review available licenses by using MSOnline:
  - **Get-MsolAccountSku**
- To assign licenses to a user by using MSOnline:
  - **Set-MsolUserLicense -UserPrincipalName "AbbieP@adatum.com" -AddLicenses "Adatum:SPE\_E3"**
- To disable specific service plans by using MSOnline:
  - **\$planList = "YAMMER\_ENTERPRISE","SWAY"**
  - **\$licenseOptions=New-MsolLicenseOptions -AccountSkuld \$accountSkuld -DisabledPlans \$planList**
  - **Set-MsolUserLicense -UserPrincipalName "AbbieP@adatum.com" -LicenseOptions \$licenseOptions**
- To remove a license from a user by using MSOnline:
  - **Set-MsolUserLicense -UserPrincipalName "AbbieP@adatum.com" -RemoveLicenses "Adatum:SPE\_E3"**





**Center for Career Development**

*by LINKgroup*

# Lesson 6: Manage Exchange Online with Powershell



# **Center for Career Development**

*by LINKgroup*

## Topics:

- Connecting to Exchange Online PowerShell
- Managing mailboxes in Exchange Online
- Managing resources in Exchange Online
- Managing admin roles in Exchange Online

## Connecting to Exchange Online Powershell

- To install the Exchange Online module:
  - **Install-Module -Name ExchangeOnlineManagement**
  - Scripts must be enabled
  - Basic authentication must be allowed for the WinRM client
- To connect to Exchange Online:
  - **\$ProxyOptions = New-PSSessionOption -ProxyAccessType IEConfig**
  - **Connect-ExchangeOnline -PsSessionOption \$ProxyOptions**

## Managing mailboxes in Exchange Online

- Mailboxes are created automatically for users when an Exchange Online license is applied
- Mailboxes created with New-Mailbox are typically:
  - Room
  - Equipment
  - Shared
- To modify a mailbox:
  - **Set-Mailbox AbbieP@adatum.com -ForwardingSmtpAddress DoraM@adatum.com -DeliverToMailboxAndForward \$true**
- **Get-EXOMailbox** returns only a partial details by default
- Cmdlets to manage:
  - Mailbox permissions
  - Mailbox folder permissions

## Managing resources in Exchange Online

- Resource mailboxes are used to book equipment and rooms
- Delegates can approve meeting requests for resource mailboxes by running:
  - **Set-CalendarProcessing -Identity BoardRoom -ResourceDelegates AbbieP@adatum.com**
- Some parameters for the Set-CalendarProcessing cmdlet are listed in the following table:

<i>-AllBookInPolicy</i>	<i>-AllowConflicts</i>
<i>-AllRequestInPolicy</i>	<i>-AllRequestOutOfPolicy</i>
<i>-AutomateProcessing</i>	<i>-BookInPolicy</i>
<i>-EnforceCapacity</i>	<i>-MaximumDurationInLimits</i>
<i>-RequestInPolicy</i>	<i>-RequestOutOfPolicy</i>

## Managing admin roles in Exchange Online

- Exchange Online has admin roles that are separate from the Microsoft 365 admin roles, including:
  - Organization Management
  - Recipient Management
  - View-only Management
  - Records Management
  - Discovery Management
- To add a user to a role, you use:
  - **Add-RoleGroupMember -Identity "Recipient Management" -Member AbbieP@adatum.com**
- It's possible to create custom role groups as well



**Center for Career Development**

*by LINKgroup*

# Lesson 7: Manage SharePoint Online with Powershell



# **Center for Career Development**

*by LINKgroup*

## Topics:

- SharePoint Online Management Shell overview
- Managing SharePoint Online users and groups with PowerShell
- Managing sites with Windows PowerShell
- Managing external user sharing with Windows PowerShell





## SharePoint Online Management Shell overview

- To install SharePoint Online Management Shell:
  - **Install-Module -Name Microsoft.Online.SharePoint.PowerShell**
- To update SharePoint Online Management Shell:
  - **Update-Module -Name Microsoft.Online.SharePoint.PowerShell**
- To connect to SharePoint Online:
  - **Connect-SPOService -Url <https://yourtenant-admin.sharepoint.com>**



## Managing SharePoint Online users and groups with Powershell

- To assign permissions to a site, create a SharePoint group, then add Azure AD users to the group
- SharePoint groups are created per site
- To create a SharePoint group:
  - **New-SPOSiteGroup -Group MarketingUsers -PermissionLevels Read -Site <https://adatum.sharepoint.com/sites/Marketing>**
- To add a user to a SharePoint group:
  - **Add-SPOUser -Site <https://adatum.sharepoint.com/sites/Marketing> -Group MarketingUsers -LoginName AbbieP@adatum.com**
- Security groups can also be added to SharePoint groups



## Managing SharePoint sites with Windows Powershell

- SharePoint online contains sites for Collaboration, Microsoft 365 groups, Microsoft Teams, and OneDrive
- To create a SharePoint site:
  - **New-SPOSite -Url <https://adatum.sharepoint.com/sites/Marketing> -Owner AbbieP@adatum.com -StorageQuota 256**
- To review a list of available templates:
  - **Get-SPOWebTemplate**
- To modify a site:
  - **Set-SPOSite -Identity <https://adatum.sharepoint.com/sites/Marketing> -Title "Marketing Portal"**
- To review site configuration:
  - **Get-SPOSite -Identity <https://adatum.sharepoint.com/sites/Marketing> | Format-List**
- To remove a site:
  - **Remove-SPOSite -Identity <https://adatum.sharepoint.com/sites/Marketing>**



## Managing SharePoint sites with Windows Powershell

- Configure sharing for a site:
  - Set-SPOsite -https://adatum.sharepoint.com/sites/Marketing -SharingCapability Disabled
- Valid values for SharingCapability are:
  - ExternalUserAndGuestSharing
  - ExternalUserSharingOnly
  - ExistingExternalUserSharingOnly
  - Disabled
- Other parameters for sharing include:

<i>-DefaultLinkSharingType</i>	<i>-DefaultLinkPermission</i>
<i>-AnonymousLinkExpirationDays</i>	<i>-OverrideTenantAnonymousLinkExpirationPolicy</i>
<i>-ExternalLinkExpirationDays</i>	<i>-OverrideTenantExternalLinkExpirationPolicy</i>



**Center for Career Development**

*by LINKgroup*

# Lesson 7: Manage Microsoft Teams with Powershell



# **Center for Career Development**

*by LINKgroup*

## Topics:

- Overview of the Microsoft Teams PowerShell module
- Installing the Microsoft Teams PowerShell module
- Managing Teams with the Microsoft Teams PowerShell module

## Overview of the Microsoft Teams Powershell module

- Cmdlets for team management in the Microsoft Teams PowerShell module use Team as part of the noun:
  - **Get-Team**
  - **Add-TeamUser**
  - **New-TeamsApp**
- Functions that manage communication policies include:
  - **Set-CsTeamsMeetingPolicy**
  - **Remove-CsTeamTemplate**
  - **New-CsTeamsEmergencyCallingPolicy**
  - **Get-CsTeamsMessagingPolicy**

## Installing the Microsoft Teams Powershell module

- To install the module:
  - **Install-Module -Name MicrosoftTeams**
- To update the module:
  - **Update-Module -Name MicrosoftTeams**
- To connect to Microsoft Teams:
  - **Connect-MicrosoftTeams**



## Managing Teams with the Microsoft Teams Powershell module

- To create a team:
  - **New-Team -DisplayName "Marketing Team"**
- Template use is limited within PowerShell
- To review a list of teams:
  - **Get-Team**
- To configure a team:
  - **Set-Team -GroupId 26be526d-201a-4af6-9918-2fdbf6306916 -MailNickName "MarketingTeam".**
- To add a team member:
  - **Add-TeamUser -GroupId 26be526d-201a-4af6-9918-2fdbf6306916 -User AbbieP@adatum.com -Role Member**



## Managing Teams with the Microsoft Teams PowerShell module (Slide 2)

- Other cmdlets for managing teams:
  - **New-TeamChannel**
  - **Add-TeamChannelUser**
  - **Get-TeamChannel**
  - **Remove-TeamChannelUser**
  - **Set-TeamChannel**