



AZ-040

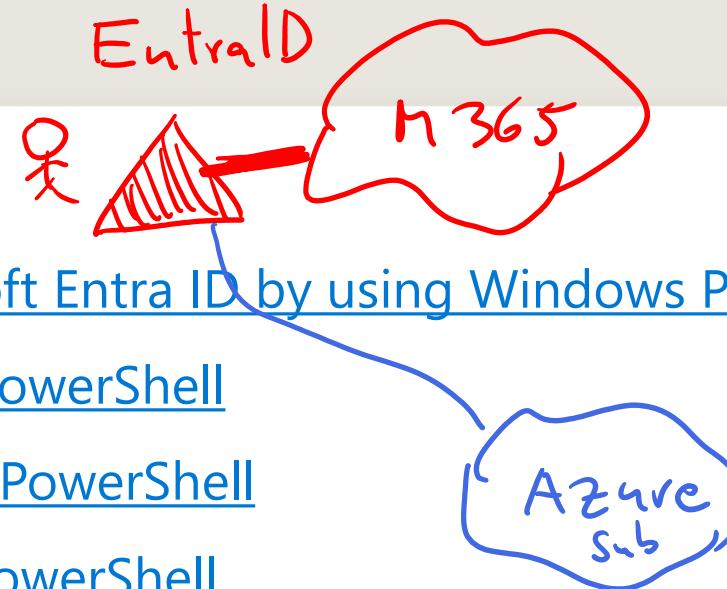
Automate Administration with PowerShell



Learning Path 10: Manage Microsoft 365 services by using Windows PowerShell

Learning objectives

- Manage users, groups, and licenses in Microsoft Entra ID by using Windows PowerShell
- Manage Exchange Online by using Windows PowerShell
- Manage SharePoint Online by using Windows PowerShell
- Manage Microsoft Teams by using Windows PowerShell



Agenda

- LP 1 Get started with Windows PowerShell
- LP 2 Maintain system administration tasks in Windows PowerShell
- LP 3 Work with the Windows PowerShell pipeline
- LP 4 Work with PowerShell providers and PowerShell drives in Windows PowerShell
- LP 5 Querying management information by using CIM and WMI
- LP 6 Use variables, arrays, and hash tables in Windows PowerShell scripts
- LP 7 Create and modify scripts by using Windows PowerShell
- LP 8 Administer remote computers by using Windows PowerShell
- LP 9 Manage cloud resources by using Windows PowerShell
- LP 10 Manage Microsoft 365 services by using Windows PowerShell ←
- LP 11 Create and manage background jobs and scheduled jobs in Windows PowerShell

Overview

Microsoft 365 is a cloud service that includes many components. The most commonly used services in Microsoft 365 are Microsoft Entra ID, Exchange Online, SharePoint Online, and Microsoft Teams. To efficiently manage Microsoft 365 and gain access to all configuration options, you should know how to use PowerShell to manage those services.

Modules:

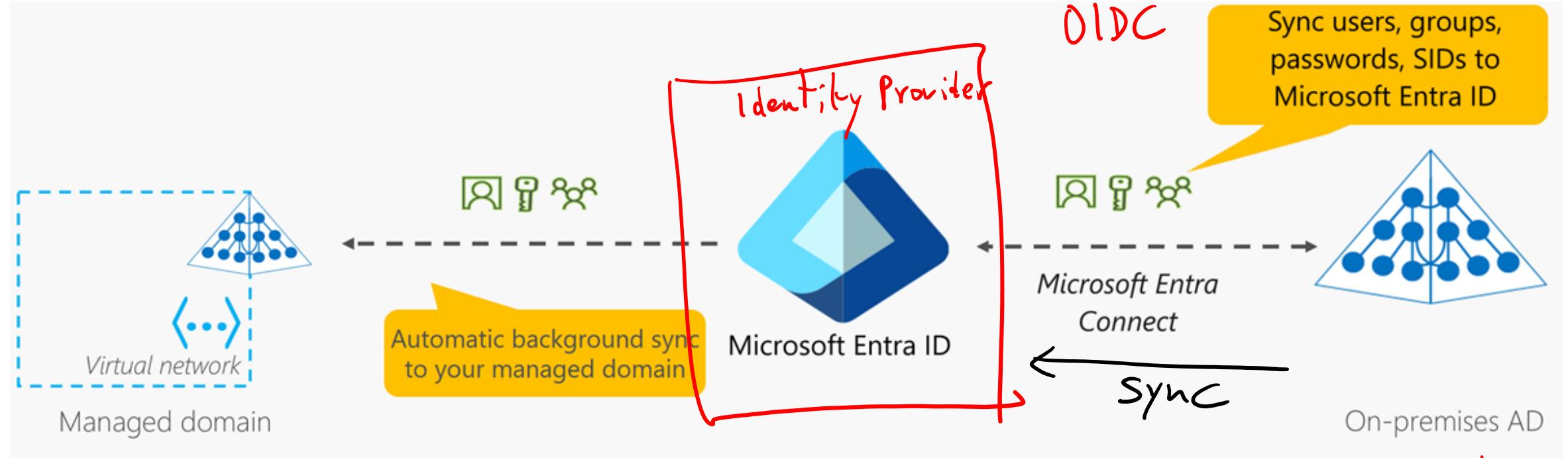
- Manage users, groups, and licenses in Microsoft Entra ID by using Windows PowerShell
- Manage Exchange Online by using Windows PowerShell
- Manage SharePoint Online by using Windows PowerShell
- Manage Microsoft Teams by using Windows PowerShell

ISE → VS Code
+ PowerShell Ext.

Examine Microsoft Entra ID

Access Token
JWT
OAuth 2.0

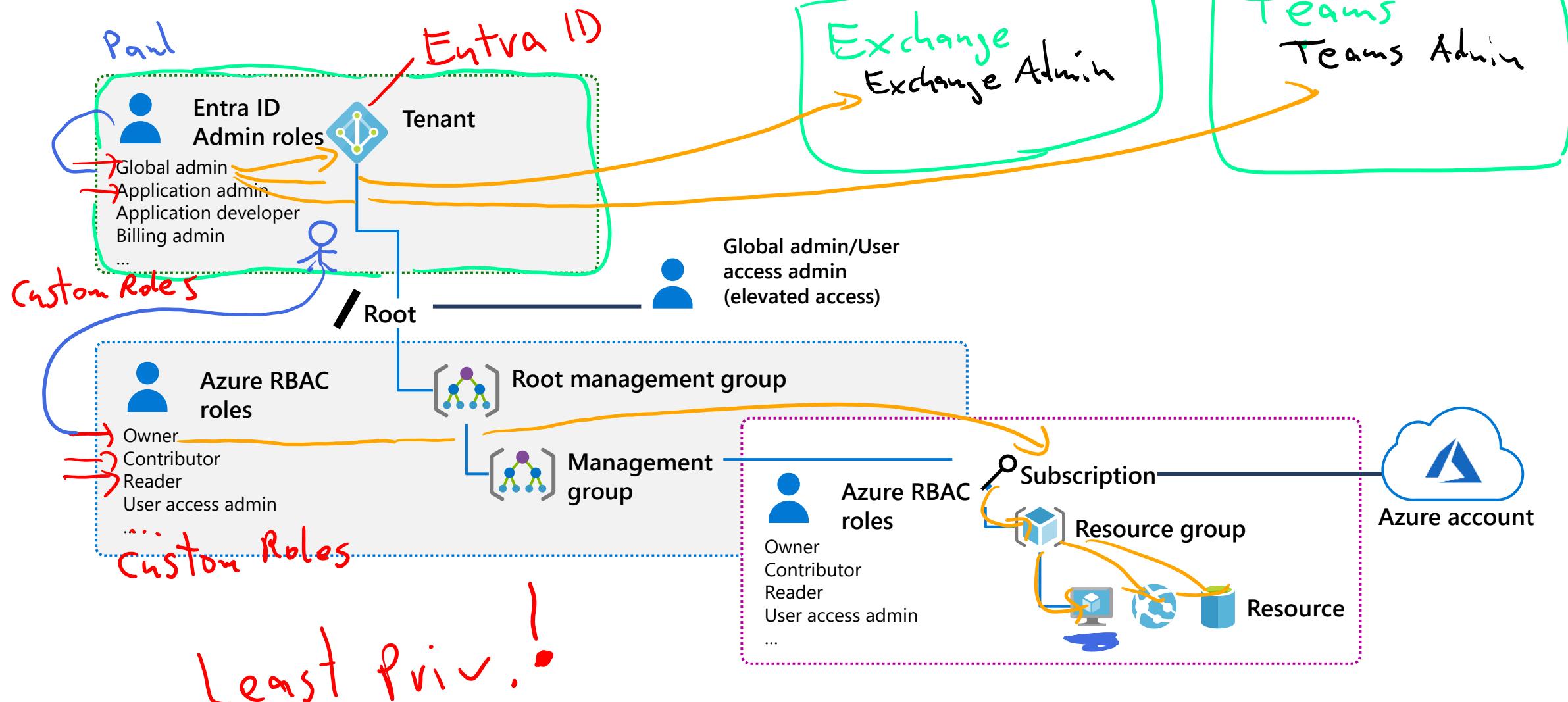
OIDC

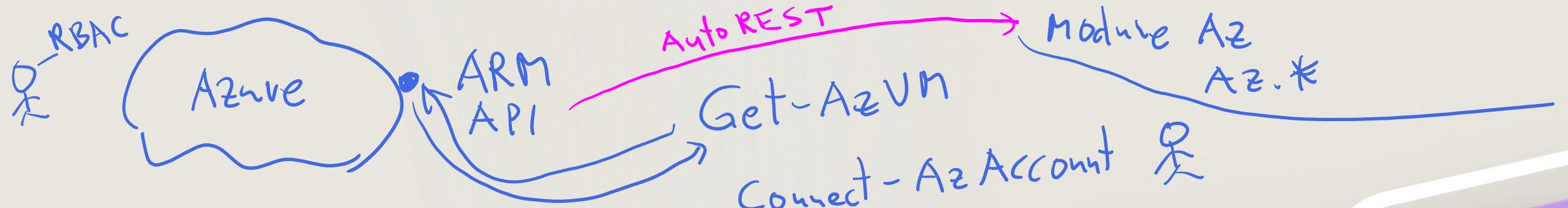


- Configure access to applications, including single sign-on
- Manage and provision users and groups
- Providing an identity management solution, including federation
- Implement security features like multi-factor authentication and conditional access

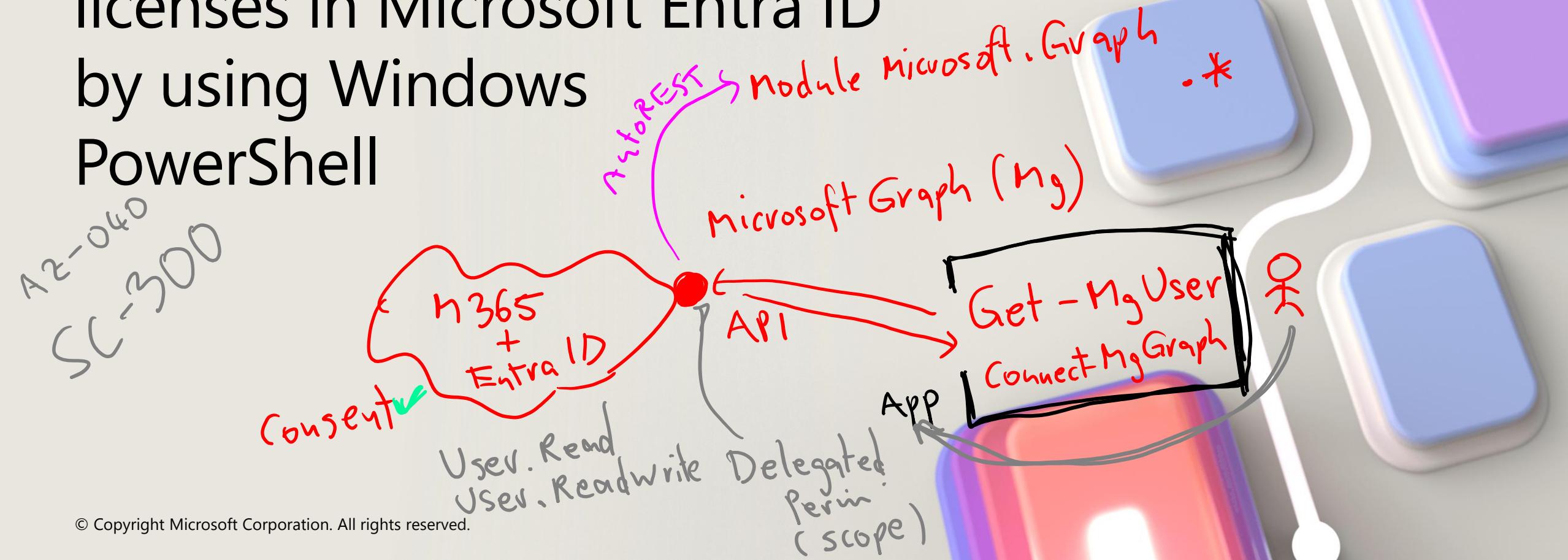
keyberos
klist tgt

Apply RBAC Authentication





Manage users, groups, and licenses in Microsoft Entra ID by using Windows PowerShell



Module overview

Microsoft 365 includes multiple cloud services. However, the core of Microsoft 365 is Microsoft Entra ID, which provides identity management for all the services in Microsoft 365. In this Module, you'll learn how to create users and groups, and manage roles and licenses.

Units:

- 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 Microsoft Entra ID:
 - 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

- 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-AzureADMSDeletedDirectoryObject**

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 -AccountSkuid \$accountSkuid -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"**

Manage Exchange Online by using Windows PowerShell

Module overview

Exchange Online is one of the most commonly used services in Microsoft 365. You can use PowerShell cmdlets to efficiently manage bulk operations and perform tasks that aren't possible through the web-based administrative interface. Being adept at managing Exchange Online with PowerShell will make you a much better administrator.

Units:

- 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 customized role groups as well

Manage SharePoint Online by using Windows PowerShell

Module overview

SharePoint Online is a collaboration service that allows you to store and share information through a web-based interface. Many organizations use SharePoint sites to build web portals for departments. In this Module, you'll learn how to create a site and assign user permissions.

Units:

- 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 Microsoft Entra ID 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 external user sharing 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>

Manage Microsoft Teams by using Windows PowerShell

Manage Microsoft Teams by using Windows PowerShell overview

Microsoft Teams is a collaboration service that combines multiple Microsoft 365 services into a single interface. It's a useful tool for workgroups and projects. You can use the Microsoft Teams PowerShell module to perform tasks such as creating teams and managing user permissions.

Units:

- 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**

Lab – Managing Microsoft 365 with PowerShell

Lab: Managing Microsoft 365 with PowerShell



- **Exercise 1:** Managing users and groups in Microsoft Entra ID
- **Exercise 2:** Managing Exchange Online
- **Exercise 3:** Managing SharePoint Online
- **Exercise 4:** Managing Microsoft Teams

Sign-in information for the exercises:

Virtual machines: **AZ-040T00A-LON-DC1** and **AZ-040T00A-LON-CL1**

Username: **Adatum\Administrator**

Password: **Pa55w.rd**

Lab scenario



You've created a new Microsoft 365 tenant. As a new administrator, you want to try using PowerShell to manage some of the Microsoft 365 services before you start deploying them to users.

Lab-review questions



- Why do you need an **AssignedLicense** and an **AssignedLicenses** object to configure licenses?
- Which cmdlet do you use to configure auto acceptance of meeting requests for resource mailboxes?
- Which cmdlet do you use to list the templates available in SharePoint Online?

Lab-review answers



- Why do you need an **AssignedLicense** and **AssignedLicenses** object to configure licenses?

The **AssignedLicense** object contains the specific license you want to add or remove. The **AssignedLicenses** object contains the **AssignedLicense** object and indicates whether that license should be added or removed.

- Which cmdlet is used to configure auto acceptance of meeting requests for resource mailboxes?

Set-CalendarProcessing

- Which cmdlet do you use to list the templates available in SharePoint Online?

Get-SPOWebTemplate

References

[Get started with the Microsoft Graph PowerShell SDK](#)

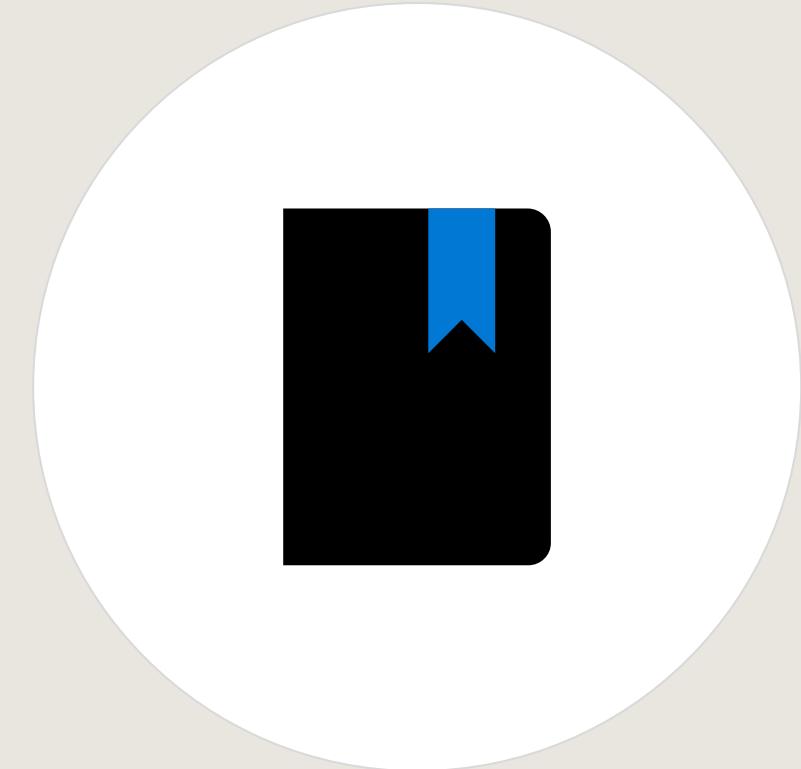
[Overview of Azure Cloud Shell](#)

[Compare groups](#)

[Product names and service plan identifiers for licensing](#)

[Permissions in Exchange Online](#)

[Compare groups](#)



Learning Path Recap

In this learning path, we:

- Learned about giving access to the services in Microsoft 365, to create user accounts and then assign licenses that provide access to the services.
- Explored managing mailboxes, resources, and admin roles in Exchange Online with PowerShell.
- Learned to manage SharePoint Online users and groups with PowerShell.
- Learned how to use Microsoft Teams PowerShell module to perform tasks such as creating teams and managing user permissions.

End of presentation