



Windows PowerShell Get-Help on Cmdlet 'Connect-AzAccount'

PS:\>Get-HELP Connect-AzAccount -Full

NAME

Connect-AzAccount

SYNOPSIS

Connect to Azure with an authenticated account for use with cmdlets from the Az PowerShell modules.

SYNTAX

```
Connect-AzAccount -AccessToken <System.String> -AccountId <System.String> [-ContextName <System.String>]
[-DefaultProfile
    <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Environment
<System.String>] [-Force] [-GraphAccessToken <System.String>]
    [-KeyVaultAccessToken <System.String>] [-MaxContextPopulation <System.Int32>] [-MicrosoftGraphAccessToken
<System.String>] [-Scope {Process | CurrentUser}]
    [-SkipContextPopulation] [-SkipValidation] [-Subscription <System.String>] [-Tenant <System.String>] [-Confirm] [-WhatIf]
[<CommonParameters>]
```

```
Connect-AzAccount [-AccountId <System.String>] [-AuthScope <System.String>] [-ContextName <System.String>]
[-DefaultProfile
    <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Environment
```

<System.String>] [-Force] [-MaxContextPopulation
 <System.Int32>] [-Scope {Process | CurrentUser}] [-SkipContextPopulation] [-Subscription <System.String>] [-Tenant
<System.String>] [-UseDeviceAuthentication]
 [-Confirm] [-WhatIf] [<CommonParameters>]

Connect-AzAccount [-AccountId <System.String>] [-AuthScope <System.String>] [-ContextName <System.String>]
[-DefaultProfile
 <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Environment
<System.String>] [-Force] [-Identity] [-MaxContextPopulation
 <System.Int32>] [-Scope {Process | CurrentUser}] [-SkipContextPopulation] [-Subscription <System.String>] [-Tenant
<System.String>] [-Confirm] [-WhatIf]
 [<CommonParameters>]

Connect-AzAccount -ApplicationId <System.String> [-AuthScope <System.String>] -CertificateThumbprint
<System.String> [-ContextName <System.String>] [-DefaultProfile
 <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Environment
<System.String>] [-Force] [-MaxContextPopulation
 <System.Int32>] [-Scope {Process | CurrentUser}] [-SendCertificateChain] [-ServicePrincipal] [-SkipContextPopulation]
[-Subscription <System.String>] -Tenant
 <System.String> [-Confirm] [-WhatIf] [<CommonParameters>]

Connect-AzAccount -ApplicationId <System.String> [-ContextName <System.String>] [-DefaultProfile
 <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Environment
<System.String>] -FederatedToken <System.String> [-Force]
 [-MaxContextPopulation <System.Int32>] [-Scope {Process | CurrentUser}] [-ServicePrincipal] [-SkipContextPopulation]
[-Subscription <System.String>] -Tenant
 <System.String> [-Confirm] [-WhatIf] [<CommonParameters>]

Connect-AzAccount -ApplicationId <System.String> [-CertificatePassword <System.Security.SecureString>]
-CertificatePath <System.String> [-ContextName <System.String>]
 [-DefaultProfile <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>]
[-Environment <System.String>] [-Force]
 [-MaxContextPopulation <System.Int32>] [-Scope {Process | CurrentUser}] [-SendCertificateChain] [-ServicePrincipal]

`[-SkipContextPopulation] [-Subscription`

`<System.String>] -Tenant <System.String> [-Confirm] [-WhatIf] [<CommonParameters>]`

`Connect-AzAccount [-AuthScope <System.String>] [-ContextName <System.String>] -Credential
<System.Management.Automation.PSCredential> [-DefaultProfile
<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Environment
<System.String>] [-Force] [-MaxContextPopulation
<System.Int32>] [-Scope {Process | CurrentUser}] -ServicePrincipal [-SkipContextPopulation] [-Subscription
<System.String>] -Tenant <System.String> [-Confirm]
[-WhatIf] [<CommonParameters>]`

`Connect-AzAccount [-AuthScope <System.String>] [-ContextName <System.String>] -Credential
<System.Management.Automation.PSCredential> [-DefaultProfile
<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Environment
<System.String>] [-Force] [-MaxContextPopulation
<System.Int32>] [-Scope {Process | CurrentUser}] [-SkipContextPopulation] [-Subscription <System.String>] [-Tenant
<System.String>] [-Confirm] [-WhatIf]
[<CommonParameters>]`

DESCRIPTION

The ``Connect-AzAccount`` cmdlet connects to Azure with an authenticated account for use with cmdlets from the Az PowerShell modules. You can use this authenticated

account only with Azure Resource Manager requests. To add an authenticated account for use with Service Management, use the ``Add-AzureAccount`` cmdlet from the Azure

PowerShell module. If no context is found for the current user, the user's context list is populated with a context for each of their first 25 subscriptions. The list

of contexts created for the user can be found by running ``Get-AzContext -ListAvailable``. To skip this context population, specify the `SkipContextPopulation` switch

parameter. After executing this cmdlet, you can disconnect from an Azure account using ``Disconnect-AzAccount``.

PARAMETERS

-AccessToken <System.String>

Specifies an access token.

> [!CAUTION] > Access tokens are a type of credential. You should take the appropriate security precautions to > keep them confidential. Access tokens also timeout and may prevent long running tasks from > completing.

Required? true

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-AccountId <System.String>

Account Id / User Id / User Name to login with in Default (UserWithSubscriptionId) parameter set; Account ID for access token in AccessToken parameter set;

Account ID for managed service in ManagedService parameter set. Can be a managed service resource ID, or the associated client ID. To use the system assigned identity, leave this field blank.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ApplicationId <System.String>

Application ID of the service principal.

Required? true

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-AuthScope <System.String>

Optional OAuth scope for login, supported pre-defined values: AadGraph, AnalysisServices, Attestation, Batch, DataLake, KeyVault, OperationalInsights, Storage,

Synapse. It also supports resource id like `https://storage.azure.com/`.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-CertificatePassword <System.Security.SecureString>

The password required to access the pkcs#12 certificate file.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-CertificatePath <System.String>

The path of certificate file in pkcs#12 format.

Required? true

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-CertificateThumbprint <System.String>

Certificate Hash or Thumbprint.

Required?	true
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-ContextName <System.String>

Name of the default Azure context for this login. For more information about Azure contexts, see Azure PowerShell context objects

(/powershell/azure/context-persistence).

Required?	false
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-Credential <System.Management.Automation.PSCredential>

Specifies a PSCredential object. For more information about the PSCredential object, type ``Get-Help Get-Credential``. The PSCredential object provides the user ID and password for organizational ID credentials, or the application ID and secret for service principal credentials.

Required?	true
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-DefaultProfile <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>

The credentials, account, tenant, and subscription used for communication with Azure.

Required?	false
-----------	-------

Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-Environment <System.String>

Environment containing the Azure account.

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-FederatedToken <System.String>

Specifies a token provided by another identity provider. The issuer and subject in this token must be first configured to be trusted by the ApplicationId.

> [!CAUTION] > Federated tokens are a type of credential. You should take the appropriate security precautions to keep them confidential. Federated tokens also timeout and may prevent long running tasks from completing.

Required? true
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-Force <System.Management.Automation.SwitchParameter>

Overwrite the existing context with the same name without prompting.

Required? false
Position? named

Default value False
Accept pipeline input? False
Accept wildcard characters? false

-GraphAccessToken <System.String>

AccessToken for Graph Service.

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-Identity <System.Management.Automation.SwitchParameter>

Login using a Managed Service Identity.

Required? true
Position? named
Default value False
Accept pipeline input? False
Accept wildcard characters? false

-KeyVaultAccessToken <System.String>

AccessToken for KeyVault Service.

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-MaxContextPopulation <System.Int32>

Max subscription number to populate contexts after login. Default is 25. To populate all subscriptions to contexts, set to

-1.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-MicrosoftGraphAccessToken <System.String>

Access token to Microsoft Graph

Required?	false
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-Scope <Microsoft.Azure.Commands.Profile.Common.ContextModificationScope>

Determines the scope of context changes, for example, whether changes apply only to the current process, or to all sessions started by this user.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-SendCertificateChain <System.Management.Automation.SwitchParameter>

Specifies if the x5c claim (public key of the certificate) should be sent to the STS to achieve easy certificate rollover in Azure AD.

Required?	false
Position?	named

Default value False
Accept pipeline input? False
Accept wildcard characters? false

-ServicePrincipal <System.Management.Automation.SwitchParameter>

Indicates that this account authenticates by providing service principal credentials.

Required? true
Position? named
Default value False
Accept pipeline input? False
Accept wildcard characters? false

-SkipContextPopulation <System.Management.Automation.SwitchParameter>

Skips context population if no contexts are found.

Required? false
Position? named
Default value False
Accept pipeline input? False
Accept wildcard characters? false

-SkipValidation <System.Management.Automation.SwitchParameter>

Skip validation for access token.

Required? false
Position? named
Default value False
Accept pipeline input? False
Accept wildcard characters? false

-Subscription <System.String>

Subscription Name or ID.

Required? false
Position? named
Default value None
Accept pipeline input? True (ByValue)
Accept wildcard characters? false

-Tenant <System.String>

Optional tenant name or ID.

> [!NOTE] > Due to limitations of the current API, you must use a tenant ID instead of a tenant name when >
connecting with a business-to-business (B2B) account.

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-UseDeviceAuthentication <System.Management.Automation.SwitchParameter>

Use device code authentication instead of a browser control.

Required? false
Position? named
Default value False
Accept pipeline input? False
Accept wildcard characters? false

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

Required? false
Position? named

Default value False
Accept pipeline input? False
Accept wildcard characters? false

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

Required? false
Position? named
Default value False
Accept pipeline input? False
Accept wildcard characters? false

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug,
ErrorAction, ErrorVariable, WarningAction, WarningVariable,
OutBuffer, PipelineVariable, and OutVariable. For more information, see
about_CommonParameters (<https://go.microsoft.com/fwlink/?LinkID=113216>).

INPUTS

System.String

OUTPUTS

Microsoft.Azure.Commands.Profile.Models.Core.PSAzureProfile

NOTES

----- Example 1: Connect to an Azure account -----

Connect-AzAccount

Please select the account you want to login with.

Retrieving subscriptions for the selection...

[Tenant and subscription selection]

No	Subscription name	Subscription ID	Tenant domain name
[1]	Subscription1	xxxx-xxxx-xxxx-xxxx	xxxxxxxxx.xxxxxxxxxxxx.com
[2]	Subscription2	xxxx-xxxx-xxxx-xxxx	xxxxxxxxx.xxxxxxxxxxxx.com
...			
[9]	Subscription9	xxxx-xxxx-xxxx-xxxx	xxxxxxxxx.xxxxxxxxxxxx.com

Select a tenant and subscription: 1 <requires user's input here>

Subscription name	Tenant domain name
Subscription1	xxxxxxxxx.xxxxxxxxxxxx.com

[Announcements]

Share your feedback regarding your experience with `Connect-AzAccount` at: <https://aka.ms/azloginfeedback>

If you encounter any problem, please open an issue at: <https://aka.ms/azpsissue>

SubscriptionName	Tenant
Subscription1	xxxxxxxxx.xxxxxxxxxxxx.com

Example 2: Connect to Azure using organizational ID credentials

```
$Credential = Get-Credential
```

```
Connect-AzAccount -Credential $Credential
```

Account	SubscriptionName	TenantId	Environment
-----	-----	-----	
azureuser@contoso.com	Subscription1	xxxx-xxxx-xxxx-xxxx	AzureCloud

Example 3: Connect to Azure using a service principal account

```
$SecurePassword = ConvertTo-SecureString -String "Password123!" -AsPlainText -Force
```

```
$TenantId = 'yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyy'
```

```
$ApplicationId = 'zzzzzzzz-zzzz-zzzz-zzzz-zzzzzzzz'
```

```
$Credential = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $ApplicationId,  
$SecurePassword
```

```
Connect-AzAccount -ServicePrincipal -TenantId $TenantId -Credential $Credential
```

Account	SubscriptionName	TenantId	Environment
-----	-----	-----	
xxxx-xxxx-xxxx-xxxx	Subscription1	xxxx-xxxx-xxxx-xxxx	AzureCloud

Example 4: Use an interactive login to connect to a specific tenant and subscription

```
Connect-AzAccount -Tenant 'xxxx-xxxx-xxxx-xxxx' -SubscriptionId 'yyyy-yyyy-yyyy-yyyy'
```

Account	SubscriptionName	TenantId	Environment
-----	-----	-----	
azureuser@contoso.com	Subscription1	xxxx-xxxx-xxxx-xxxx	AzureCloud

----- Example 5: Connect using a Managed Service Identity -----

Connect-AzAccount -Identity

Set-AzContext -Subscription Subscription1

Account	SubscriptionName	TenantId	Environment
-----	-----	-----	
MSI@50342	Subscription1	xxxx-xxxx-xxxx-xxxx	AzureCloud

Example 6: Connect using Managed Service Identity login and ClientId

\$identity = Get-AzUserAssignedIdentity -ResourceGroupName 'myResourceGroup' -Name 'myUserAssignedIdentity'

Get-AzVM -ResourceGroupName contoso -Name testvm | Update-AzVM -IdentityType UserAssigned -IdentityId \$identity.Id

Connect-AzAccount -Identity -AccountId \$identity.ClientId # Run on the virtual machine

Account	SubscriptionName	TenantId	Environment
-----	-----	-----	
yyyy-yyyy-yyyy-yyyy	Subscription1	xxxx-xxxx-xxxx-xxxx	AzureCloud

----- Example 7: Connect using certificates -----

\$Thumbprint = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'

\$TenantId = 'yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyy'

\$ApplicationId = '00000000-0000-0000-0000-00000000'

Connect-AzAccount -CertificateThumbprint \$Thumbprint -ApplicationId \$ApplicationId -Tenant \$TenantId
-ServicePrincipal

Account	SubscriptionName	TenantId	Environment
---------	------------------	----------	-------------

-----	-----	-----	
-------	-------	-------	--

xxxxxxxx-xxxx-xxxx-xxxxxxxx	Subscription1	yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyy	AzureCloud
-----------------------------	---------------	----------------------------------	------------

Account : xxxxxxxx-xxxx-xxxx-xxxxxxxx

SubscriptionName : MyTestSubscription

SubscriptionId : zzzzzzzz-zzzz-zzzz-zzzz-zzzzzzzz

TenantId : yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyy

Environment : AzureCloud

----- Example 8: Connect with AuthScope -----

Connect-AzAccount -AuthScope Storage

Account	SubscriptionName	TenantId	Environment
---------	------------------	----------	-------------

-----	-----	-----	
-------	-------	-------	--

yyyy-yyyy-yyyy-yyyy	Subscription1	xxxx-xxxx-xxxx-xxxx	AzureCloud
---------------------	---------------	---------------------	------------

----- Example 9: Connect using certificate file -----

```
$SecurePassword = ConvertTo-SecureString -String "Password123!" -AsPlainText -Force
```

```
$TenantId = 'yyyyyyyy-yyyy-yyyy-yyyy-yyyy'
```

```
$ApplicationId = 'zzzzzzzz-zzzz-zzzz-zzzz-zzzzzzzz'
```

```
Connect-AzAccount -ServicePrincipal -ApplicationId $ApplicationId -TenantId $TenantId -CertificatePath  
'./certificatefortest.pfx' -CertificatePassword $SecurePassword
```

Account	SubscriptionName	TenantId	Environment
-----	-----	-----	
xxxxxxxx-xxxx-xxxx-xxxxxxxx	Subscription1	yyyyyyyy-yyyy-yyyy-yyyy-yyyy	AzureCloud

----- Example 10: Connect interactively using WAM -----

```
Update-AzConfig -EnableLoginByWam $true
```

```
Connect-AzAccount
```

Account	SubscriptionName	TenantId	Environment
-----	-----	-----	
xxxxxxxx-xxxx-xxxx-xxxxxxxx	Subscription1	yyyyyyyy-yyyy-yyyy-yyyy-yyyy	AzureCloud

RELATED LINKS

Online Version: <https://learn.microsoft.com/powershell/module/az.accounts/connect-azaccount>