

Application management documentation

Microsoft Entra ID is an Identity and Access Management (IAM) system. It provides a single place to store information about digital identities. You can configure your software applications to use Microsoft Entra ID as the place where user information is stored.

Fundamentals

OVERVIEW

[What is application management?](#)

CONCEPT

[What type of apps does Microsoft Entra support?](#)

[What is single sign-on?](#)

Develop an app

CONCEPT

[Authentication and authorization concepts](#)

HOW-TO GUIDE

[Build an app using Microsoft identities or social accounts](#)

Integrate an app with the tenant

HOW-TO GUIDE

[Get started with app integration](#)

[Add a preintegrated cloud application](#)

[Register your app](#)

Configure an app

HOW-TO GUIDE

[Configure app properties](#)

[Assign users and groups](#)

[Provision an app](#)

[Configure My Apps](#)

REFERENCE

[Manage applications using Microsoft Graph API](#)

Secure an app

HOW-TO GUIDE

[Set up Conditional Access](#)

[Set up multifactor authentication](#)

[Manage certificates](#)

[Set up tenant restrictions](#)

[Configure token encryption](#)

[Use Defender for Cloud Apps](#)

[Act on overprivileged and suspicious app](#)

Manage app access

OVERVIEW

[Identity governance](#)

[User and admin consent](#)

HOW-TO GUIDE

[Assign roles](#)

[Configure user consent](#)

[Configure admin consent](#)

[Configure permissions classification](#)

[Manage entitlement](#)

Maintain an app

HOW-TO GUIDE

[View, search, sort, filter list of apps in a tenant](#)

[Troubleshoot sign-in errors](#)

[Disable user sign-in for an app](#)

[Remove user access to an app](#)

[Delete an app](#)

Monitor an app

CONCEPT

[Sign in logs](#)

[Usage and insights report](#)

[Audit logs](#)

[Provisioning logs](#)

HOW-TO GUIDE

[Access activity logs](#)

[Download logs](#)

[Set up access reviews](#)

[Assign owners](#)

Remote access to on-premises apps

CONCEPT

[Application proxy](#)

HOW-TO GUIDE

[Plan application proxy deployment](#)

[Set up connectors](#)

[Configure single sign-on](#)

[Publish native client applications](#)

[Publish claims-aware applications](#)

What is application management in Microsoft Entra ID?

Article • 11/29/2024

Application management in Microsoft Entra ID is the process of creating, configuring, managing, and monitoring applications in the cloud. When an [application](#) is registered in a Microsoft Entra tenant, users who are already assigned to it can securely access it. Many types of applications can be registered in Microsoft Entra ID. For more information, see [Application types for the Microsoft identity platform](#).

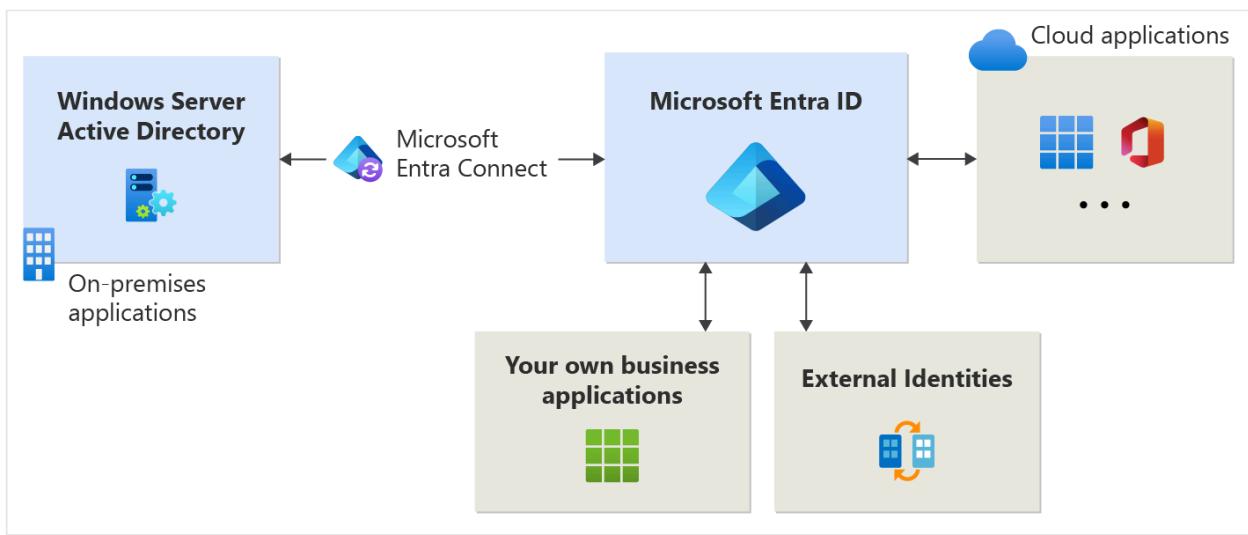
In this article, you learn these important aspects of managing the lifecycle of an application:

- **Develop, add, or connect** – You take different paths depending on whether you're developing your own application, using a preintegrated application, or connecting to an on-premises application.
- **Manage access** – Access can be managed by using single sign-on (SSO), assigning resources, defining the way access is granted and consented to, and using automated provisioning.
- **Configure properties** – Configure the requirements for signing into the application and how the application is represented in user portals.
- **Secure the application** – Manage configuration of permissions, multifactor authentication, Conditional Access, tokens, and certificates.
- **Govern and monitor** – Manage interaction and review activity using entitlement management and reporting and monitoring resources.
- **Clean up** – When your application is no longer needed, clean up your tenant by removing access to it and deleting it.

Develop, add, or connect

There are several ways that you might manage applications in Microsoft Entra ID. The easiest way to start managing an application is to use a preintegrated application from the Microsoft Entra gallery. Developing your own application and registering it in Microsoft Entra ID is an option, or you can continue to use an on-premises application.

The following image shows how these applications interact with Microsoft Entra ID.



Preintegrated applications

Many applications are already preintegrated (shown as **Cloud applications** in the previous image in this article) and can be set up with minimal effort. Each application in the Microsoft Entra gallery has an article available that shows you the steps required to [configure the application](#). For a simple example of how an application can be added to your Microsoft Entra tenant from the gallery, see [Quickstart: Add an enterprise application](#).

Your own applications

If you develop your own business application, you can register it with Microsoft Entra ID to take advantage of the security features that the tenant provides. You can register your application in [App Registrations](#), or you can register it using the [Create your own application](#) link when adding a new application in [Enterprise applications](#). Consider how [authentication](#) is implemented in your application for integration with Microsoft Entra ID.

If you want to make your application available through the gallery, you can [submit a request to make it available](#).

On-premises applications

If you want to continue using an on-premises application, but take advantage of what Microsoft Entra ID offers, connect it with Microsoft Entra ID using [Microsoft Entra application proxy](#). Application Proxy can be implemented when you want to publish on-premises applications externally. Remote users who need access to internal applications can then access them in a secure manner.

Manage access

To [manage access](#) for an application, you want to answer the following questions:

- How is access granted and consented for the application?
- Does the application support SSO?
- Which users, groups, and owners should be assigned to the application?
- Are there other identity providers that support the application?
- Is it helpful to automate the provisioning of user identities and roles?

Access and consent

You can [manage user consent settings](#) to choose whether users can allow an application or service to access user profiles and organizational data. When applications are granted access, users can sign in to applications integrated with Microsoft Entra ID, and the application can access your organization's data to deliver rich data-driven experiences.

In situations where users are unable to consent to the permissions an application is requesting, consider configuring the admin consent workflow. The workflow allows users to provide a justification and request an administrator's review and approval of an application. To learn how to configure admin consent workflow in your Microsoft Entra tenant, see [Configure admin consent workflow](#).

As an administrator, you can [grant tenant-wide admin consent](#) to an application.

Tenant-wide admin consent is necessary when an application requires permissions that regular users aren't allowed to grant. Granting tenant-wide admin consent also allows organizations to implement their own review processes. Always carefully review the permissions the application is requesting before granting consent. When an application is granted tenant-wide admin consent, all users are able to sign into the application unless you configure it to require user assignment.

Single sign-on

Consider implementing SSO in your application. You can manually configure most applications for SSO. The most popular options in Microsoft Entra ID are [SAML-based SSO](#) and [OpenID Connect-based SSO](#). Before you start, make sure that you understand the requirements for SSO and how to [plan for deployment](#). For more information on how to configure SAML-based SSO for an enterprise application in your Microsoft Entra tenant, see [Enable single sign-on for an application by using Microsoft Entra ID](#).

User, group, and owner assignment

By default, all users can access your enterprise applications without being assigned to them. However, if you want to assign the application to a set of users, configure the application to require user assignment and assign the select users to the application. For a simple example of how to create and assign a user account to an application, see [Quickstart: Create and assign a user account](#).

If included in your subscription, [assign groups to an application](#) so that you can delegate ongoing access management to the group owner.

[Assigning owners](#) is a simple way to grant the ability to manage all aspects of Microsoft Entra configuration for an application. As an owner, a user can manage the organization-specific configuration of the application. As a best practice, you should proactively monitor applications in your tenant to ensure they have at least two owners, to avoid the situation of ownerless applications.

Automate provisioning

[Application provisioning](#) refers to automatically creating user identities and roles in the applications that users need to access. In addition to creating user identities, automatic provisioning includes the maintenance and removal of user identities as status or roles change.

Identity providers

Do you have an identity provider that you want Microsoft Entra ID to interact with?

[Home Realm Discovery](#) provides a configuration that allows Microsoft Entra ID to determine which identity provider a user needs to authenticate with when they sign in.

User portals

Microsoft Entra ID provides customizable ways to deploy applications to users in your organization. For example, the [My Apps portal or the Microsoft 365 application launcher](#). My Apps gives users a single place to start their work and find all the applications to which they have access. As an administrator of an application, you should [plan how the users in your organization use My Apps](#).

Configure properties

When you add an application to your Microsoft Entra tenant, you have the opportunity to configure properties that affect the way users can interact with the application. You can enable or disable the ability to sign in and set the application to require user

assignment. You can also determine the visibility of the application, what logo represents the application, and any notes about the application. For more information about the properties that can be configured, see [Properties of an enterprise application](#).

Secure the application

There are several methods available to help you keep your enterprise applications secure. For example, you can [restrict tenant access](#), [manage visibility, data, and analytics](#), and possibly provide [hybrid access](#). Keeping your enterprise applications secure also involves managing configuration of permissions, MFA, Conditional Access, tokens, and certificates.

Permissions

It's important to periodically review and, if necessary, [manage the permissions granted to an application or service](#). Make sure that you only allow the appropriate access to your applications by regularly evaluating whether suspicious activity exists.

[Permission classifications](#) allow you to identify the effect of different permissions according to your organization's policies and risk evaluations. For example, you can use permission classifications in consent policies to identify the set of permissions that users are allowed to consent to.

Multifactor authentication and Conditional Access

Microsoft Entra multifactor authentication helps safeguard access to data and applications, providing another layer of security by using a second form of authentication. There are many methods that can be used for a second-factor authentication. Before you start, [plan the deployment of MFA for your application](#) in your organization.

Organizations can enable MFA with [Conditional Access](#) to make the solution fit their specific needs. Conditional Access policies allow administrators to assign controls to specific [applications, actions, or authentication context](#).

Tokens and certificates

Different types of security tokens are used in an authentication flow in Microsoft Entra ID depending on the protocol used. For example, [SAML tokens](#) are used for the SAML protocol, and [ID tokens](#) and [access tokens](#) are used for the OpenID Connect protocol.

Tokens are signed with the unique certificate that Microsoft Entra ID generates and by specific standard algorithms.

You can provide more security by [encrypting the token](#). You can also manage the information in a token including the [roles that are allowed](#) for the application.

Microsoft Entra ID uses the [SHA-256 algorithm](#) by default to sign the SAML response. Use SHA-256 unless the application requires SHA-1. Establish a process for [managing the lifetime of the certificate](#). The maximum lifetime of a signing certificate is three years. To prevent or minimize outage due to a certificate expiring, use roles and email distribution lists to ensure that certificate-related change notifications are closely monitored.

Govern and monitor

[Entitlement management](#) in Microsoft Entra ID enables you to manage interaction between applications and administrators, catalog owners, access package managers, approvers, and requestors.

Your Microsoft Entra reporting and monitoring solution depends on your legal, security, and operational requirements and your existing environment and processes. There are several logs that are maintained in Microsoft Entra ID. Therefore, you should [plan for reporting and monitoring deployment](#) to maintain the best experience as possible for your application.

Clean up

You can clean up access to applications. For example, [removing a user's access](#). You can also [disable how a user signs in](#). And finally, you can delete the application if it's no longer needed for the organization. For more information on how to delete an enterprise application from your Microsoft Entra tenant, see [Quickstart: Delete an enterprise application](#).

Guided walkthrough

For a guided walkthrough of many of the recommendations in this article, see the [Microsoft 365 Secure your cloud apps with Single Sign On \(SSO\) guided walkthrough](#).

Next steps

- Get started by adding your first enterprise application with the [Quickstart: Add an enterprise application](#).
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Microsoft Entra application management: What's new

Article • 05/06/2025

Welcome to what's new in Microsoft Entra application management documentation. This article lists new docs and those articles that had significant updates in the last three months. To learn what's new with the application management service, see [What's new in Microsoft Entra ID](#).

April 2025

New articles

- [Tutorial: Enforce secret and certificate standards using application management policies](#)

Updated articles

Reviewed the following articles for technical accuracy and clarity:

- [Migrate applications away from secret-based authentication](#)
- [Configure how users consent to applications](#)
- [Review and take action on admin consent requests](#)
- [Tutorial: Configure F5 BIG-IP Access Policy Manager for Kerberos authentication](#)
- [Tutorial: Manage certificates for federated single sign-on](#)

March 2025

Updated articles

Replaced reference examples of Azure AD PowerShell with Microsoft Entra PowerShell in the following articles:

- [Configure Microsoft Entra SAML token encryption](#)
- [Configure permission classifications](#)
- [Delete an enterprise application](#)
- [Disable user sign-in for an application](#)
- [Hide an enterprise application](#)

- Manage users and groups assignment to an application
- Restore a soft deleted enterprise application
- Review permissions granted to enterprise applications
- [Overview of user and admin consent](#) - Revised the article to improve technical accuracy and clarity.

February 2025

New articles

- Enable single sign-on for an enterprise application with a relying party STS

Updated articles

Reviewed the following articles for technical accuracy and clarity:

- Add an OpenID Connect-based single sign-on application
- Configure enterprise application properties
- [Tutorial: Manage certificates for federated single sign-on](#)

Quickstart: Add an enterprise application

Article • 03/31/2025

In this quickstart, you use the Microsoft Entra admin center to add an enterprise application to your Microsoft Entra tenant. Microsoft Entra ID has a gallery that contains thousands of enterprise applications that are already preintegrated. Many of the applications your organization uses are probably already in the gallery. This quickstart uses the application named **Microsoft Entra SAML Toolkit** as an example, but the concepts apply for most [enterprise applications in the gallery](#).

We recommend that you use a nonproduction environment to test the steps in this quickstart.

Prerequisites

To add an enterprise application to your Microsoft Entra tenant, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free](#).
- One of the following roles: Cloud Application Administrator, or Application Administrator.

Add an enterprise application

To add an enterprise application to your tenant:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. Select **New application**.
4. The **Browse Microsoft Entra Gallery** pane opens and displays tiles for cloud platforms, on-premises applications, and featured applications. Applications listed in the **Featured applications** section have icons indicating whether they support federated single sign-on (SSO) and provisioning. Search for and select the application. In this quickstart, **Microsoft Entra SAML Toolkit** is being used.

The screenshot shows the Microsoft Entra admin center interface. On the left, there's a navigation sidebar with various sections like Home, Favorites, Identity, Users, Groups, Devices, Applications, Protection, Identity governance, External Identities, Show more, and Learn & support. The Applications section is expanded, showing Enterprise applications, App registrations, Protection, Identity governance, External Identities, and Show more. The Microsoft Entra SAML Toolkit application is listed under Enterprise applications. To the right, a modal window titled 'Microsoft Entra SAML Toolkit' is open. It contains fields for 'Name' (set to 'Microsoft Entra SAML Toolkit 1'), 'Publisher' (Microsoft Corporation), 'Provisioning' (Automatic provisioning is not supported), 'Single Sign-On Mode' (SAML-based Sign-on), and 'URL' (https://www.microsoft.com/). Below these fields, there's a note about reading a step-by-step tutorial and a link to it. A 'Create' button is at the bottom of the modal.

5. Enter a name that you want to use to recognize the instance of the application. For example, `Microsoft Entra SAML Toolkit 1`.
6. Select **Create**, you're taken to the application that you registered.
7. You should [assign owners to the application](#) as a best practice at this point.

If you choose to install an application that uses OpenID Connect based SSO, instead of seeing a **Create** button, you see a button that redirects you to the application sign-in or sign-up page depending on whether you already have an account there. For more information, see [Add an OpenID Connect based single sign-on application](#). After sign-in, the application is added to your tenant.

Clean up resources

If you're planning to complete the next quickstart, keep the enterprise application that you created. Otherwise, you can consider deleting it to clean up your tenant. For more information, see [Delete an application](#).

Microsoft Graph API

To add an application from the Microsoft Entra gallery programmatically, use the [applicationTemplate: instantiate](#) API in Microsoft Graph.

Next steps

Learn how to create a user account and assign it to the enterprise application that you added.

[Create and assign a user account](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Quickstart: Create and assign a user account

Article • 03/21/2025

In this quickstart, you use the Microsoft Entra admin center to create a user account in your Microsoft Entra tenant. After you create the account, you can assign it to the enterprise application that you added to your tenant.

We recommend that you use a nonproduction environment to test the steps in this quickstart.

Prerequisites

To create a user account and assign it to an enterprise application, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free](#).
- One of the following roles: Cloud Application Administrator, or owner of the service principal. You need the User Administrator role to manage users.
- Completion of the steps in [Quickstart: Add an enterprise application](#).

Create a user account

To create a user account in your Microsoft Entra tenant:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **User Administrator**.
2. Browse to **Identity > Users > All users**
3. Select **New user** at the top of the pane and then, select **Create new user**.
4. In the **User principal name** field, enter the username of the user account. For example, `b.simon@contoso.com`. Be sure to change `contoso.com` to the name of your tenant domain.
5. In the **Display name** field, enter the name of the user of the account. For example, `B.Simon`.
6. Enter the details required for the user under the **Groups and roles**, **Settings**, and **Job info** sections.
7. Select **Create**.

Assign a user account to an enterprise application

To assign a user account to an enterprise application:

1. Sign in to the Microsoft Entra admin center [↗](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
For example, the application that you created in the previous quickstart named **Microsoft Entra SAML Toolkit 1**.
3. In the left pane, select **Users and groups**, and then select **Add user/group**.

The screenshot shows the Microsoft Entra admin center interface. The left sidebar navigation includes Home, Favorites, Identity, Overview, Users, Groups, Devices, Applications (with Enterprise applications selected), App registrations, Protection, Identity governance, External identities, Show more, Protection, Identity governance, and Learn & support. The main content area displays the 'Microsoft Entra SAML Toolkit 1 | Users and groups' page for an Enterprise Application. The top navigation bar shows Home > Enterprise applications > All applications > Browse Microsoft Entra Gallery > Microsoft Entra SAML Toolkit 1. Below the title, there are tabs for Overview, Deployment Plan, Diagnose and solve problems, Manage (Properties, Owners, Roles and administrators, and **Users and groups**), Security (Conditional Access, Permissions, Token encryption), Activity (Sign-in logs, Usage & insights, Audit logs, Provisioning logs), and a search bar. A red box highlights the 'Users and groups' tab under Manage. The right side of the screen shows a table with columns for Display Name, Object Type, and Role assigned. A message at the top states: 'The application will appear for assigned users within My Apps. Set 'visible to users?' to no in properties to prevent this.' A note below says 'Assign users and groups to app-roles for your application here. To create new app-roles for this application, use the application registration.' A search bar at the bottom right contains the placeholder 'First 200 shown, to search all users & gro...'. A magnifying glass icon is in the bottom right corner.

4. On the Add Assignment pane, select **None Selected** under **Users and groups**.
5. Search for and select the user that you want to assign to the application. For example, `b.simon@contoso.com`.
6. Select **Select**.
7. Select **None Selected** under **Select a role** and then select the role that you want to assign to the user. For example, **Standard User**.
8. Select **Select**.
9. Select **Assign** at the bottom of the pane to assign the user to the application.

After you assign the user to the application, ensure you set the application to be visible to the user. To make it visible to assigned users, select **Properties** in the left pane, and then set **Visible to users?** to **Yes**.

Clean up resources

If you're planning to complete the next quickstart, keep the application that you created. Otherwise, you can consider deleting it to clean up your tenant.

Next steps

Learn how to set up single sign-on for an enterprise application.

[Enable single sign-on](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Quickstart: View enterprise applications

Article • 05/13/2025

In this quickstart, you learn how to use the Microsoft Entra admin center to search for and view the enterprise applications configured in your Microsoft Entra tenant.

We recommend that you use a nonproduction environment to test the steps in this quickstart.

Prerequisites

To view applications registered in your Microsoft Entra tenant, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free ↗](#).
- One of the following roles: Cloud Application Administrator, or owner of the service principal.
- Completion of the steps in [Quickstart: Add an enterprise application](#).

View a list of applications

To view the enterprise applications registered in your tenant:

1. Sign in to the [Microsoft Entra admin center ↗](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Entra ID > Enterprise apps > All applications**.

Name	Object ID	Application ID	Homepage URL	Created on
Contoso-app1...	aaaaaaaa-1111-222...	bbbbbbbb-2222-3333...		9/16/2024
Contoso-app2...	ccccccc-4444-555...	ddddddd-5555-6666...		9/24/2024
Contoso-app3...	eeeeeee-6666-77...	fffffff-7777-8888-ggg...		9/5/2024
Contoso-app4...	ggggggg-9999-1...	hhhhhhh-2222-3333...	https://host/contoso...	8/30/2024

3. To view more applications, select **Load more** at the bottom of the list. If there are many applications in your tenant, it might be easier to search for a particular application instead.

of scrolling through the list.

Search for an application

To search for a particular application:

1. Select the **Application Type** filter option. Select **All applications** from the **Application Type** drop-down menu, and choose **Apply**.
2. Enter the name of the application you want to find. If the application is already in your Microsoft Entra tenant, it appears in the search results. For example, you can search for the **Microsoft Entra SAML Toolkit 1** application that is used in the previous quickstarts.
3. Try entering the first few letters of an application name.

Select viewing options

Select options according to what you're looking for:

1. The default filters are **Application Type** and **Application ID starts with**.
2. Under **Application Type**, choose one of these options:
 - **Enterprise Applications** shows non-Microsoft applications.
 - **Microsoft Applications** shows Microsoft applications.
 - **Managed Identities** shows applications that are used to authenticate to services that support Microsoft Entra authentication.
 - **Agent ID (Preview)** shows AI agent identities that are used by AI agents to authenticate to services that support Microsoft Entra authentication.
 - **All Applications** shows both non-Microsoft and Microsoft applications.
3. Under **Application ID starts with**, enter the first few digits of the application ID if you know the application ID.
4. After choosing the options you want, select **Apply**.
5. Select **Add filters** to add more options for filtering the search results. The other options include:
 - **Application Status**
 - **Application Visibility**
 - **Created on**
 - **Assignment required**
 - **Is App Proxy**
 - **Owner**
 - **Identifier URI (Entity ID)**
 - **Homepage URL**

6. To remove any of the filter options already added, select the X icon next to the filter option.

Clean up resources

If you created a test application named **Microsoft Entra SAML Toolkit 1** that was used throughout the quickstarts, you can consider deleting it now to clean up your tenant. For more information, see [Delete an application](#).

Related content

- [Delete an application](#)

Tutorial: Manage application access and security

Article • 04/25/2025

The IT administrator at Fabrikam has added and configured an application from the Microsoft Entra application gallery. They now need to understand the features that are available to manage access to the application and make sure the application is secure. Using the information in this tutorial, an administrator learns how to:

- ✓ Grant consent for the application on behalf of all users
- ✓ Enable multifactor authentication to make sign-in more secure
- ✓ Communicate a term of use to users of the application
- ✓ Create a collection in the My Apps portal

Prerequisites

- An Azure account with an active subscription. [Create an account for free](#).
- One of the following roles: Privileged Role Administrator, Cloud Application Administrator, or Application Administrator.
- An enterprise application that has been configured in your Microsoft Entra tenant.
- At least one user account added and assigned to the application. For more information, see [Quickstart: Create and assign a user account](#).

Grant tenant wide admin consent

For the application that the administrator added to their tenant, they want to set it up so that all users in the organization can use it and not have to individually request consent to use it. To avoid the need for user consent, they can grant consent for the application on behalf of all users in the organization. For more information, see [Consent and permissions overview](#).

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Entra ID > Enterprise apps**.
3. Select the application to which you want to grant tenant-wide admin consent.
4. Under **Security**, select **Permissions**.
5. Carefully review the permissions that the application requires. If you agree with the permissions the application requires, select **Grant admin consent**.

Create a Conditional Access policy

The administrator wants to make sure that only the people they assign to the application can securely sign in. To do this, they can configure a Conditional Access policy for a group of users that enforces multifactor authentication. For more information, see [What is Conditional Access?](#).

Create a group

It's easier for an administrator to manage access to the application by assigning all users of the application to a group. The administrator can then manage access at a group level.

1. In the left menu of the tenant overview, select **Groups > All groups**.
2. Select **New group** at the top of the pane.
3. Enter *MFA-Test-Group* for the name of the group.
4. Select **No members selected**, and then choose the user account that you assigned to the application.
5. Select **Create**.

Create a Conditional Access policy for the group

1. In the left menu of the tenant overview, select **Entra ID > Conditional Access**.
2. Select **+ New policy**, and then select **Create new policy**.
3. Enter a name for the policy, such as *MFA Pilot*.
4. Under **Assignments**, select **Users or workload identities**.
5. On the **Include** tab, choose **Select users and groups**, and then select **Users and groups**.
6. Browse for and select the *MFA-Test-Group* that you previously created, and then choose **Select**.
7. Don't select **Create** yet, you add MFA to the policy in the next section.

Configure multifactor authentication

In this tutorial, the administrator can find the basic steps to configure the application, but they should consider creating a plan for MFA before starting. For more information, see [Plan a Microsoft Entra multifactor authentication deployment](#).

1. Under **Cloud apps or actions**, select **No cloud apps, actions, or authentication contexts selected**. For this tutorial, on the **Include** tab, choose **Select resources**.
2. Search for and select your application, and then select **Select**.
3. Under **Access controls and Grant**, select **0 controls selected**.
4. Check the box for **Require multifactor authentication**, and then choose **Select**.
5. Set **Enable policy** to **On**.
6. To apply the Conditional Access policy, select **Create**.

Test multifactor authentication

1. Open a new browser window in InPrivate or incognito mode and browse to the URL of the application.
2. Sign in with the user account that you assigned to the application. You're required to register for and use Microsoft Entra multifactor authentication. Follow the prompts to complete the process and verify you successfully sign in to the Microsoft Entra admin center.
3. Close the browser window.

Create a terms of use statement

Juan wants to make sure that certain terms and conditions are known to users before they start using the application. For more information, see [Microsoft Entra terms of use](#).

1. In Microsoft Word, create a new document.
2. Type My terms of use, and then save the document on your computer as *mytou.pdf*.
3. Under **Manage**, in the **Conditional Access** menu, select **Terms of use**.
4. In the top menu, select **+ New terms**.
5. In the **Name** textbox, type *My TOU*.
6. In the **Display name** textbox, type *My TOU*.
7. Upload your terms of use PDF file.
8. For **Language**, select **English**.
9. For **Require users to expand the terms of use**, select **On**.
10. For **Enforce with Conditional Access policy templates**, select **Custom policy**.
11. Select **Create**.

Add the terms of use to the policy

1. Browse to **Entra ID > Conditional Access > Policies**.
2. From the list of policies, select the *MFA Pilot* policy.
3. Under **Access controls** and **Grant**, select the controls selected link.
4. Select *My TOU*.
5. Select **Require all the selected controls**, and then choose **Select**.
6. Select **Save**.

Create a collection in the My Apps portal

The My Apps portal enables administrators and users to manage the applications used in the organization. For more information, see [End-user experiences for applications](#).

(!) Note

Applications only appear in a user's my Apps portal after the user is assigned to the application and the application is configured to be visible to users. See [Configure application properties](#) to learn how to make the application visible to users.

By default, all applications are listed together on a single page. But you can use collections to group together related applications and present them on a separate tab, making them easier to find. For example, you can use collections to create logical groupings of applications for specific job roles, tasks, projects, and so on. In this section, you create a collection and assign it to users and groups.

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Entra ID > Enterprise apps**.
3. Under **Manage**, select **App launchers > Collections**.
4. Select **New collection**. In the New collection page, enter a **Name** for the collection (it's recommended to not use "collection" in the name). Then enter a **Description**.
5. Select the **Applications** tab. Select **+ Add application**, and then in the Add applications page, select all the applications you want to add to the collection, or use the Search box to find applications.
6. When you're finished adding applications, select **Add**. The list of selected applications appears. You can use the arrows to change the order of applications in the list.
7. Select the **Owners** tab. Select **+ Add users and groups**, and then in the Add users and groups page, select the users or groups you want to assign ownership to. When you're finished selecting users and groups, choose **Select**.
8. Select the **Users and groups** tab. Select **+ Add users and groups**, and then in the Add users and groups page, select the users or groups you want to assign the collection to. Or use the Search box to find users or groups. When you're finished selecting users and groups, choose **Select**.
9. Select **Review + Create**, and then select **Create**. The properties for the new collection appear.

Check the collection in the My Apps portal

1. Open a new browser window in InPrivate or incognito mode and browse to the [My Apps](#) portal.
2. Sign in with the user account that you assigned to the application.
3. Check that the collection you created appears in the My Apps portal.
4. Close the browser window.

Clean up resources

You can keep the resources for future use, or if you're not going to continue to use the resources created in this tutorial, delete them with the following steps.

Delete the application

1. In the left menu, select **Enterprise applications**. The **All applications** pane opens and displays a list of the applications in your Microsoft Entra tenant. Search for and select the application that you want to delete.
2. In the **Manage** section of the left menu, select **Properties**.
3. At the top of the **Properties** pane, select **Delete**, and then select **Yes** to confirm you want to delete the application from your Microsoft Entra tenant.

Delete the Conditional Access policy

1. Under **Entra ID > Conditional Access > Policies**.
2. Search for and select **MFA Pilot**.
3. Select **Delete** at the top of the pane.

Delete the group

1. Select **Entra ID > Groups**.
2. From the **All groups** page, search for and select the **MFA-Test-Group** group.
3. On the overview page, select **Delete**.

Next steps

For information about how you can make sure that your application is healthy and being used correctly, see:

[Govern and monitor your application](#)

Tutorial: Govern and monitor applications

Article • 04/25/2025

The IT administrator at Fabrikam has added and configured an application from the [Microsoft Entra application gallery](#). They also made sure that access can be managed and that the application is secure by using the information in [Tutorial: Manage application access and security](#). They now need to understand the resources that are available to govern and monitor the application.

Using the information in this tutorial, an administrator of the application learns how to:

- ✓ Create an access review
- ✓ Access the audit logs
- ✓ Access the sign-ins
- ✓ Send logs to Azure Monitor

Prerequisites

- An Azure account with an active subscription. If you don't already have one, [Create an account for free](#).
- One of the following roles: Identity Governance Administrator, Privileged Role Administrator, Cloud Application Administrator, or Application Administrator.
- An enterprise application that has been configured in your Microsoft Entra tenant.

Create an access review

The administrator wants to make sure that users or guests have appropriate access. They decide to ask users of the application to participate in an access review and recertify or attest to their need for access. When the access review is finished, they can then make changes and remove access from users who no longer need it. For more information, see [Manage user and guest user access with access reviews](#).

To create an access review:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **ID Governance > Access reviews**.
3. Select **New access review** to create a new access review.
4. In **Select what to review**, select **Applications**.
5. Select + **Select application(s)**, select the application, and then choose **Select**.
6. Now you can select a scope for the review. Your options are:

- **Guest users only** - This option limits the access review to only the Microsoft Entra B2B guest users in your directory.
- **All users** - This option scopes the access review to all user objects associated with the resource. Select **All users**.

7. Select **Next: Reviews**.

8. In the **Specify reviewers** section, in the **Select reviewers** box, select **Selected user(s) or group(s)**, select **+ Select reviewers**, and then select the user account that is assigned to the application.

9. In the **Specify recurrence of review** section, specify the following selections:

- **Duration (in days)** - Accept the default value of **3**.
- **Review recurrence** - select **One time**.
- **Start date** - Accept today's date as the start date.

10. Select **Next: Settings**.

11. In the **Upon completion settings** section, you can specify what happens after the review finishes. Select **Auto apply results to resource**.

12. Select **Next: Review + Create**.

13. Name the access review. Optionally, give the review a description. The name and description are shown to the reviewers.

14. Review the information and select **Create**.

Start the access review

The access review starts in a few minutes and it appears in your list with an indicator of its status.

By default, Microsoft Entra ID sends an email to reviewers shortly after the review starts. If you choose not to have Microsoft Entra ID send the email, be sure to inform the reviewers that an access review is waiting for them to complete. You can show them the instructions for how to review access to groups or applications. If your review is for guests to review their own access, show them the instructions for how to review access for themselves to groups or applications.

If you've assigned guests as reviewers and they haven't accepted their invitation to the tenant, they won't receive an email from access reviews. They must first accept the invitation before they can begin reviewing.

View the status of an access review

You can track the progress of access reviews as they're completed.

1. Go to **ID Governance > Access reviews**.

2. In the list, select the access review you created.
3. On the **Overview** page, check the progress of the access review.

The **Results** page provides information on each user under review in the instance, including the ability to Stop, Reset, and Download results. To learn more, check out the [Complete an access review of groups and applications in Microsoft Entra access reviews](#) article.

Access the audit logs

The Microsoft Entra audit logs capture a wide variety of activities within your tenant. These logs provide valuable insights into the activities you need to monitor. For more information, see [Audit logs in Microsoft Entra ID](#).

To access the audit logs, go to **Entra ID > Monitoring & health > Audit logs**.

The audit logs capture activities that fall under the following categories. This list is not exhaustive. For a full list of the audit log categories and activities, see [Audit log activities](#).

- Password reset activity
- Password reset registration activity
- Self-service groups activity
- Office365 Group Name Changes
- Account provisioning activity
- Password rollover status
- Account provisioning errors

Access the sign-in logs

The Microsoft Entra sign-in logs capture interactive, non-interactive, managed identity, and service principal sign-ins. For more information, see [Sign-in logs in Microsoft Entra ID](#).

To access the sign-in logs, go to **Entra ID > Monitoring & health > Sign-in logs**.

You also can view application sign-in information from the Enterprise applications area. The sign-in logs open the same logs from **Monitoring & health > Sign-in logs**, but the filter is already set to the selected application. The **Usage & insights** report also summarizes sign-in activity for the application.

Send logs to Azure Monitor

The Microsoft Entra activity logs only store information for seven days for Microsoft Entra ID Free and 30 days for Microsoft Entra ID P1/P2. Depending on your needs, you might require

extra storage to back up the activity logs data.

Using Azure Monitor logs, you can retain the data for longer and enable powerful analysis tools, such as visualization and alerts. For more information about integrating logs with Azure Monitor logs, see [Integrate Microsoft Entra logs with Azure Monitor](#).

To send logs to Azure Monitor, you need a Log Analytics workspace. Once that's created, you configure diagnostic settings to integrate with Log Analytics. There are cost considerations associated with integrating logs with Azure Monitor and Log Analytics, so review this section of [Microsoft Entra activity logs in Azure Monitor](#) before proceeding.

With a Log Analytics workspace configured:

1. Select **Diagnostic settings**, and then select **Add diagnostic setting**. You can also select Export Settings from the Audit Logs or Sign-ins page to get to the diagnostic settings configuration page.
2. Choose the logs you want to stream, select the **Send to Log Analytics workspace** option, and complete the fields.
3. Select **Save**.

After about 15 minutes, verify that events are streamed to your Log Analytics workspace.

Next steps

Advance to the next article to learn how to...

[Manage consent to applications and evaluate consent requests](#)

Tutorial: Manage certificates for federated single sign-on

Article • 04/30/2025

In this tutorial, learn how to manage federation certificates in Microsoft Entra ID by customizing expiration dates and renewing certificates for seamless SAML single sign-on (SSO).

We cover common questions and information related to certificates that Microsoft Entra ID creates to establish federated single sign-on (SSO) to your software as a service (SaaS) applications. Add applications from the Microsoft Entra application gallery or by using a non-gallery application template. Configure the application by using the federated SSO option.

This tutorial is relevant to apps that are configured to use Microsoft Entra SSO through Security Assertion Markup Language (SAML).

In this tutorial, an administrator of the application learns how to:

- ✓ Generate certificates for gallery and non-gallery applications
- ✓ Customize the expiration dates for certificates
- ✓ Add email notification address for certificate expiration dates
- ✓ Renew certificates

Prerequisites

- An Azure account with an active subscription. If you don't already have one, [Create an account for free](#).
- One of the following roles: Privileged Role Administrator, Cloud Application Administrator, or Application Administrator.
- An enterprise application configured in your Microsoft Entra tenant.

Autogenerated certificate for gallery and non-gallery applications

When you add a new application from the gallery and configure SAML-based sign-on, Microsoft Entra ID generates a self-signed certificate for the application that is valid for three years. For details on setting up SAML sign-on, see [Single sign-on to applications in Microsoft Entra ID](#).

To download the active certificate as a security certificate (.cer) file, navigate the following section in the Microsoft Entra admin center:

Entra ID > Enterprise apps > All applications > [Your application] > Single sign-on and select a download link in the **SAML Certificates** heading.

You can choose between the raw (binary) certificate or the Base 64 (base 64-encoded text) certificate. For gallery applications, this section might also show a link to download the certificate as federation metadata XML (an .xml file), depending on the requirement of the application.

You can also download an active or inactive certificate by selecting the **Token Signing Certificate** heading's **Edit** icon (a pencil), which displays the **SAML Signing Certificate** page. Select the ellipsis (...) next to the certificate you want to download, and then choose which certificate format you want.

You have the other option to download the certificate in privacy-enhanced mail (PEM) format. This format is identical to Base64 but with a .pem file name extension, which isn't recognized in Windows as a certificate format. For information on frequently asked questions about certificates, see [Frequently asked questions](#).

Customize the expiration date for your federation certificate and roll it over to a new certificate

By default, Azure configures a certificate to expire after three years when you create it automatically during SAML single sign-on configuration. Because you can't change the date of a certificate after you save it, you have to:

1. Create a new certificate with the desired date.
2. Save the new certificate.
3. Download the new certificate in the correct format.
4. Upload the new certificate to the application.
5. Make the new certificate active in the Microsoft Entra admin center.

The following two sections help you perform these steps.

Create a new certificate

First create and save the new certificate with a different expiration date:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to Entra ID > Enterprise apps > All applications.
3. Enter the name of the existing application in the search box, and then select the application from the search results.

4. Under the **Manage** section, select **Single sign-on**.
5. If the **Select a single sign-on method** page appears, select **SAML**.
6. In the **Set up Single Sign-On with SAML** page, find the **SAML Certificates** heading, and select the **Edit** icon (a pencil). The **SAML Signing Certificate** page appears, which displays the status (**Active** or **Inactive**), expiration date, and thumbprint (a hash string) of each certificate.
7. Select **New Certificate**. A new row appears below the certificate list, where the expiration date defaults to exactly three years after the current date. (Your changes aren't saved yet, so you can still modify the expiration date.)
8. In the new certificate row, hover over the expiration date column and select the **Select Date** icon (a calendar). A calendar control appears, displaying the days of a month of the new row's current expiration date.
9. Use the calendar control to set a new date. You can set any date between the current date and three years after the current date.
10. Select **Save**. The new certificate now appears with a status of **Inactive**, the expiration date that you chose, and a thumbprint.

 **Note**

When you have an existing certificate that is already expired, and you generate a new certificate, the new certificate is considered for signing tokens. It's considered even though it's not yet active. The expired certificate is no longer be used for signing tokens.

11. Select the **X** to return to the **Set up Single Sign-On with SAML** page.

Upload and activate a certificate

Next, download the new certificate in the correct format, upload it to the application, and make it active in Microsoft Entra ID:

1. View more SAML sign-on configuration instructions for the application with either of following options.
 - Select the **configuration guide** link to view in a separate browser window or tab.
 - Browse to the **set up** heading and select **View step-by-step instructions** to view in a sidebar.
2. In the instructions, note the encoding format required for the certificate upload.
3. Follow the instructions in the [Autogenerated certificate for gallery and non-gallery applications](#) section earlier. This step downloads the certificate in the encoding format

required for upload by the application.

4. When you want to roll over to the new certificate, go back to the **SAML Signing Certificate** page, and in the newly saved certificate row, select the ellipsis (...) and select **Make certificate active**. The status of the new certificate changes to **Active**, and the previously active certificate changes to a status of **Inactive**.
5. Continue following the application's SAML sign-on configuration instructions that you displayed earlier, so that you can upload the SAML signing certificate in the correct encoding format.

If your app lacks certificate expiration validation and the certificate matches both Microsoft Entra ID and your app, it remains accessible. This condition is true even if the certificate is expired. Ensure your application can validate the certificate's expiration date.

If you intend to keep certificate expiry validation disabled, then the new certificate shouldn't be created until your scheduled maintenance window for the certificate rollover. If both an expired and an inactive valid certificate exist on the application, Microsoft Entra ID automatically utilizes the valid certificate. In this case, users might experience application outage.

Add email notification addresses for certificate expiration

Microsoft Entra ID sends an email notification 60, 30, and 7 days before the SAML certificate expires. You can add more than one email address to receive notifications. To specify one or more email addresses, you want the notifications to be sent to:

1. In the **SAML Signing Certificate** page, go to the **notification email addresses** heading. By default, this heading uses only the email address of the admin who added the application.
2. Below the final email address, type the email address that should receive the certificate's expiration notice, and then press Enter.
3. Repeat the previous step for each email address you want to add.
4. For each email address you want to delete, select the **Delete** icon (garbage can) next to the email address.
5. Select **Save**.

You can add up to five email addresses to the Notification list (including the email address of the admin who added the application). If you need more people to be notified, use the distribution list emails.

You receive the notification email from azure-noreply@microsoft.com. To avoid the email going to your spam location, add this email to your contacts.

Renew a certificate that is set to expire soon

If a certificate is about to expire, you can renew it using a procedure that results in no significant downtime for your users. To renew an expiring certificate:

1. Follow the instructions in the [Create a new certificate](#) section earlier, using a date that overlaps with the existing certificate. That date limits the amount of downtime caused by the certificate expiration.
2. If the application can automatically roll over a certificate, set the new certificate to active by following these steps.
 - a. Go back to the **SAML Signing Certificate** page.
 - b. In the newly saved certificate row, select the ellipsis (...) and then select **Make certificate active**.
 - c. Skip the next two steps.
3. If the application can only handle one certificate at a time, pick a downtime interval to perform the next step. (Otherwise, if the application doesn't automatically pick up the new certificate but can handle more than one signing certificate, you can perform the next step anytime).
4. Before the old certificate expires, follow the instructions in the [Upload and activate a certificate](#) section earlier. If your application certificate isn't updated after a new certificate is updated in Microsoft Entra ID, authentication on your application might fail.
5. Sign in to the application to make sure that the certificate works correctly.

If your app lacks certificate expiration validation and the certificate matches both Microsoft Entra ID and your app, it remains accessible. This condition is true even if the certificate is expired. Ensure your application can validate certificate expiration.

Related content

- [Application management with Microsoft Entra ID](#)
- [Single sign-on to applications in Microsoft Entra ID](#)
- [Debug SAML-based single sign-on to applications in Microsoft Entra ID](#)

Tutorial: Enforce secret and certificate standards using application management policies

Article • 04/29/2025

In this tutorial, you learn how to enforce secret and certificate standards using application management policies in Microsoft Entra ID.

Ensuring that applications in your organization are using secure authentication is crucial for protecting sensitive data and maintaining the integrity of your systems. Microsoft Entra ID provides a way to enforce secret and certificate restrictions through application management policies. This feature can help you manage what kinds of secrets and keys can be used and ensure that they're rotated regularly. Application management policies can only be updated using Microsoft Graph PowerShell or Microsoft Graph API. To learn more about this feature, see [Microsoft Entra application management policies API overview](#).

Policies can be applied to all applications in your organization or to specific applications. In this tutorial, you learn:

- ✓ Learn about recommended restrictions for secrets and certificates.
- ✓ Read the current application management policy for your tenant.
- ✓ Update the application policy to enforce restrictions.
- ✓ Confirm that the policy has been applied.

Important

Making changes to your application management policy can have a significant impact on your applications and their ability to authenticate. Before making any changes, it's important to understand the implications of those changes and how they might affect your applications. You should test any changes in a non-production environment before applying them to your production environment and make a copy of the current policy settings before you update them.

Prerequisites

- A user account. If you don't already have one, you can [create an account for free](#).
- At least the [Cloud Application Administrator](#) or [Application Administrator](#) role.
- An API client such as [Graph Explorer](#) OR

- Microsoft Graph PowerShell module installed. See [Install the Microsoft Graph PowerShell module](#).

Recommended practices for secrets and certificates

Attacks on applications often target secrets such as passwords, keys, and certificates, to gain unauthorized access to sensitive data. By enforcing restrictions, you can mitigate these risks and ensure that your applications remain secure. The following are our recommended restrictions for secrets and certificates:

- **Disable application passwords / client secrets:** Applications that use client secrets might store them in configuration files, hardcode them in scripts, or risk their exposure in other ways. The complexities of secret management make client secrets susceptible to leaks and attractive to attackers.
- **Disable symmetric key usage in applications:** Symmetric keys are similar to client secrets in that they're shared between the application and the resource it accesses. This means that if an attacker gains access to the symmetric key, they can impersonate the application and access the resource. Symmetric keys are also more difficult to manage than asymmetric keys, as they require both parties to share the same key.
- **Limit asymmetric key (certificate) lifetime to 180 days:** Certificates provide a more secure way to authenticate applications than client secrets. However, they can still be compromised if not managed properly. By limiting the lifetime of certificates, you can reduce the risk of long-lived certificates being exploited by attackers. Certificates should be rotated regularly to ensure that they aren't compromised. The recommended maximum lifetime for certificates is 180 days. This means that you should rotate your certificates at least every 180 days. Setting a shorter lifetime for highly sensitive applications can further reduce the risk of compromise. We also recommend you configure automatic rotation of certificates using Azure Key Vault. To learn more, see [Automate the rotation of a secret for resources that use one set of authentication credentials](#)

To learn more about recommended security practices for Microsoft Entra tenants, see [Configure Microsoft Entra for increased security](#).

Read your tenant application management policy

Before you create a new application management policy, you can read your existing policy to see if it meets your needs. The following example shows how to read the default application

management policy for your tenant. You can also reuse this API request to confirm the policy has been applied later in this tutorial.

Example

The following example reads the default application management policy for your tenant. The response shows the current policy settings.

Connect to Microsoft Graph using the `Connect-MgGraph` cmdlet and the `Policy.Read.All` permission. Sign in with at least the [Cloud Application Administrator](#) role. Then, run the following commands to read the default application management policy for your tenant.

PowerShell

```
Connect-MgGraph -Scopes 'Policy.Read.All'  
# Get the default application management policy  
Get-MgPolicyDefaultAppManagementPolicy | format-list
```

For more info on this cmdlet, see [Get-MgPolicyDefaultAppManagementPolicy](#).

Output

The following example shows the output of the default tenant app management policy. Your policy might differ from the example. If no policy is applied in your organization, the `id` field is set to `00000000-0000-0000-0000-000000000000` and the `isEnabled` field is set to `false`.

Output

```
ApplicationRestrictions      :  
Microsoft.Graph.PowerShell.Models.MicrosoftGraphAppManagementApplicationConfiguration  
DeletedDateTime              :  
Description                  : Default tenant policy that enforces app management  
restrictions on applications and service principals. To apply policy to targeted  
resources, create a new policy under appManagementPolicies collection.  
DisplayName                 : Default app management tenant policy  
Id                         : 00000000-0000-0000-0000-000000000000  
IsEnabled                   : false  
ServicePrincipalRestrictions :  
Microsoft.Graph.PowerShell.Models.MicrosoftGraphAppManagementServicePrincipalConfiguration  
AdditionalProperties         : {[@odata.context,  
https://graph.microsoft.com/v1.0/$metadata#policies/defaultAppManagementPolicy/$en  
tity]}
```

Important

Make a copy of the current policy settings before you update them. This will allow you to revert back to the original settings if needed. You can do this by copying the current policy settings to a file or by taking a screenshot of the settings. You won't be able to find the original settings after you update them if you have not saved them.

Update the application management policy

To implement secret and certificate restrictions, you need to update the default application management policy. This example provides our recommended settings but you can adjust them to suit your needs or even omit certain elements if you don't want to apply them. The following example shows how to update the default application management policy with the recommended settings:

- `passwordCredentials`: Allows you to set policies to restrict attributes for client secrets and symmetric keys. This can be omitted if you don't want to set a policy to restrict these types of credentials.
 - The `restrictionType` parameter allows you to set the type of restriction you want to apply. In this case, you're restricting `passwordAddition`, `customPasswordAddition`, and `symmetricKeyAddition`. These settings will limit the creation of client secrets, custom passwords and symmetric keys.
 - The `state` parameter allows you to enable or disable the restriction. If set to `enabled`, the restriction will be applied. If set to `disabled`, the restriction won't be applied.
 - The `maxLifetime` parameter allows you to set the maximum lifetime of the secret. For `passwordCredentials` you have set the value to `null`. Setting the value to `null` means that the maximum lifetime isn't restricted. This is because you're disabling the creation of client secrets and symmetric keys entirely. If you want to set a maximum lifetime for client secrets, you can set this value to a duration in ISO 8601 format. You'll find an example of this in the next section. For more information on duration formatting, see [ISO 8601 ↴](#).
 - The `restrictForAppsCreatedAfterDateTime` parameter allows you to set a date from which the policy will take effect for new applications. Any applications created before this date will be unaffected by the policy. In this case, you're applying restrictions for applications created after February 20th 2025. Please ensure you update this date to suit your needs. If you want to set different restrictions for applications created before

or after a certain date you can set multiple policies with different `restrictForAppsCreatedAfterDateTime` values.

- `keyCredentials`: Allows you to set parameters for certificates. In this case, you're restricting the lifetime of application certificates to 180 days.
 - The `restrictionType` parameter allows you to set the type of restriction you want to apply. In this case, you're restricting `asymmetricKeyLifetime`. This will limit the lifetime of application certificates to a user-defined value.
 - The `state` parameter allows you to enable or disable the restriction. If set to `enabled`, the restriction will be applied. If set to `disabled`, the restriction won't be applied.
 - The `maxLifetime` parameter allows you to set the maximum lifetime of the certificate. In this case, you're restricting the lifetime of certificates to 180 days. This is done using the ISO 8601 duration format. The prefix `P` indicates that the value is for a period of time, and `180D` indicates that the period is 180 days. You can change the number from `180` to another value to suit your specific needs. For more information on duration formatting, please see [ISO 8601](#).
- The `restrictForAppsCreatedAfterDateTime` parameter allows you to set a date from which the policy will take effect for new applications. Any applications created before this date will be unaffected by the policy. In this case, you're applying restrictions for applications created after February 20th 2025. Please ensure you update this date to suit your needs. If you want to set different restrictions for applications created before or after a certain date you can set multiple policies with different `restrictForAppsCreatedAfterDateTime` values.

Example

The following example updates the default application management policy with the settings discussed in the previous section.

PowerShell

```
Connect-MgGraph -Scopes 'Policy.ReadWrite.All'
Import-Module Microsoft.Graph.Identity.SignIns
# Define the parameters for the application management policy
$params = @{
    isEnabled = $true
    applicationRestrictions = @{
        passwordCredentials = @(
            @{
                restrictionType = "passwordAddition"
        )
    }
}
```

```

        state = "enabled"
        maxLifetime = $null
        restrictForAppsCreatedAfterDateTime = [System.DateTime]::Parse("2025-
02-20T10:37:00Z")
    }
    @{
        restrictionType = "customPasswordAddition"
        state = "enabled"
        maxLifetime = $null
        restrictForAppsCreatedAfterDateTime = [System.DateTime]::Parse("2025-
05-20T10:37:00Z")
    }
    @{
        restrictionType = "symmetricKeyAddition"
        state = "enabled"
        maxLifetime = $null
        restrictForAppsCreatedAfterDateTime = [System.DateTime]::Parse("2025-
02-20T10:37:00Z")
    }
)
keyCredentials = @(
    @{
        restrictionType = "asymmetricKeyLifetime"
        maxLifetime = "P180D"
        restrictForAppsCreatedAfterDateTime = [System.DateTime]::Parse("2025-
02-20T10:37:00Z")
    }
)
}
}

# Update the default application management policy
Update-MgPolicyDefaultAppManagementPolicy -BodyParameter $params

```

For more info on this cmdlet, see [Update-MgPolicyDefaultAppManagementPolicy](#).

Confirm the policy is applied

Once you update your application management policy, you can confirm that it's applied by reading the default application management policy again as [shown earlier](#). The response should show the updated policy with the restrictions you applied.

If it's the first time, you're applying an application management policy the `id` field should have changed from `00000000-0000-0000-0000-000000000000` to a new GUID. This change indicates that the policy is created.

You can also confirm that the policy is applied by creating a new application and checking if the restrictions are enforced. For example, if you try to create a new application with a client secret or symmetric key, you should receive an error indicating that the operation isn't allowed as shown in the below screenshot.

 Client secrets are blocked by a tenant-wide policy. Contact your tenant administrator for more information.

X

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

 New client secret

Description	Expires	Value ⓘ	Secret ID
No client secrets have been created for this application.			

Related content

- To learn how to automate secret rotation, see [Automate the rotation of a secret for resources that use one set of authentication credentials](#).
- To learn more about available restrictions and policy settings, see [Microsoft Entra application management policies API overview](#)
- To learn more about security best practices for your organization, see [Configure Microsoft Entra for increased security](#).
- To learn more about alternatives to authenticating with secrets, see [Migrate applications away from secret-based authentication](#)

Microsoft Graph PowerShell examples for Application Management

Article • 01/23/2025

The following table includes links to PowerShell script examples for Microsoft Entra Application Management.

These samples require the [Microsoft Graph PowerShell](#) SDK module.

[+] [Expand table](#)

Link	Description
Application Management scripts	
Export secrets and certs (app registrations)	Export secrets and certificates for app registrations in Microsoft Entra tenant.
Export secrets and certs (enterprise apps)	Export secrets and certificates for enterprise apps in Microsoft Entra tenant.
Export expiring secrets and certs (app registrations)	Export app registrations with expiring secrets and certificates and their Owners in Microsoft Entra tenant.
Export expiring secrets and certs (enterprise apps)	Export enterprise apps with expiring secrets and certificates and their Owners in Microsoft Entra tenant.
Export secrets and certs expiring beyond required date	Export App Registrations with secrets and certificates expiring beyond the required date in Microsoft Entra tenant. This scenario uses the noninteractive Client_Credentials Oauth flow.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

PowerShell sample: Export secrets and certificates for app registrations

Article • 01/23/2025

This PowerShell script example exports all secrets and certificates for the specified app registrations from your directory into a CSV file.

If you don't have an [Azure subscription](#), create an [Azure free account](#) before you begin.

This sample requires the [Microsoft Graph PowerShell](#) SDK module.

Sample script

PowerShell

```
<#####
#####
```

DISCLAIMER:

This is not an official PowerShell Script. We designed it specifically for the situation you have encountered right now.

Please do not modify or change any preset parameters.

Please note that we will not be able to support the script if it's changed or altered in any way or used in a different situation for other means.

This code-sample is provided "AS IS" without warranty of any kind, either expressed or implied, including but not limited to the implied warranties of merchantability and/or fitness for a particular purpose.

This sample is not supported under any Microsoft standard support program or service.

Microsoft further disclaims all implied warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose.

The entire risk arising out of the use or performance of the sample and documentation remains with you.

In no event shall Microsoft, its authors, or anyone else involved in the

```
creation, production, or  
delivery of the script be liable for any damages whatsoever (including,  
without limitation, damages  
for loss of business profits, business interruption, loss of business  
information, or other  
pecuniary loss) arising out of the use of or inability to use the sample or  
documentation, even if  
Microsoft has been advised of the possibility of such damages.  
#####
#####>
```

```
Connect-MgGraph -Scopes 'Application.Read.All'
```

```
$Messages = @{  
    DurationNotice = @{  
        Info = @(  
            'The operation is running and will take longer the more  
applications the tenant has...'  
            'Please wait...'  
        ) -join ' '  
    }  
    Export = @{  
        Info = 'Where should the CSV file export to?'  
        Prompt = 'Enter the full path in the format of <C:\Users\  
<USER>\Desktop\Users.csv>'  
    }  
}
```

```
Write-Host $Messages.DurationNotice.Info -ForegroundColor yellow
```

```
$Applications = Get-MgApplication -All
```

```
$Logs = @()
```

```
foreach ($App in $Applications) {  
    $AppName = $App.DisplayName  
    $AppID = $App.Id  
    $ApplID = $App.AppId  
  
    $AppCreds = Get-MgApplication -ApplicationId $AppID |  
        Select-Object PasswordCredentials, KeyCredentials  
  
    $Secrets = $AppCreds.PasswordCredentials  
    $Certs = $AppCreds.KeyCredentials
```

```
#####
#####
```

```
$Logs += [PSCustomObject]@{  
    'ApplicationName' = $AppName  
    'ApplicationID' = $ApplID  
    'Secret Name' = $Null  
    'Secret Start Date' = $Null  
    'Secret End Date' = $Null  
    'Certificate Name' = $Null  
    'Certificate Start Date' = $Null  
    'Certificate End Date' = $Null
```

```

'Owner'           = $Null
'Owner_ObjectID' = $Null
}
#####
foreach ($Secret in $Secrets) {
    $StartDate = $Secret.StartDateTime
    $EndDate   = $Secret.EndDateTime
    $SecretName = $Secret.DisplayName

    $Owner      = Get-MgApplicationOwner -ApplicationId $App.Id
    $Username  = $Owner.AdditionalProperties.userPrincipalName -join ';'
    $OwnerId   = $Owner.Id -join ';'

    if ($null -eq $Owner.AdditionalProperties.userPrincipalName) {
        $Username = @(
            $Owner.AdditionalProperties.displayName
            '**<This is an Application>**'
        ) -join ' '
    }
    if ($null -eq $Owner.AdditionalProperties.displayName) {
        $Username = '<<No Owner>>'
    }

    $Logs += [PSCustomObject]@{
        'ApplicationName'      = $AppName
        'ApplicationID'        = $ApplID
        'Secret Name'          = $SecretName
        'Secret Start Date'    = $StartDate
        'Secret End Date'      = $EndDate
        'Certificate Name'     = $Null
        'Certificate Start Date' = $Null
        'Certificate End Date' = $Null
        'Owner'                 = $Username
        'Owner_ObjectID'       = $OwnerId
    }
}

foreach ($Cert in $Certs) {
    $StartDate = $Cert.StartDateTime
    $EndDate   = $Cert.EndDateTime
    $CertName  = $Cert.DisplayName

    $Owner      = Get-MgApplicationOwner -ApplicationId $App.Id
    $Username  = $Owner.AdditionalProperties.userPrincipalName -join ';'
    $OwnerId   = $Owner.Id -join ';'

    if ($null -eq $Owner.AdditionalProperties.userPrincipalName) {
        $Username = @(
            $Owner.AdditionalProperties.displayName
            '**<This is an Application>**'
        ) -join ' '
    }
    if ($null -eq $Owner.AdditionalProperties.displayName) {
        $Username = '<<No Owner>>'
    }
}

```

```

$Logs += [PSCustomObject]@{
    'ApplicationName'      = $AppName
    'ApplicationID'        = $ApplID
    'Secret Name'          = $Null
    'Certificate Name'     = $CertName
    'Certificate Start Date' = $StartDate
    'Certificate End Date' = $EndDate
    'Owner'                = $Username
    'Owner_ObjectID'       = $OwnerID
    'Secret Start Date'    = $Null
    'Secret End Date'      = $Null
}
}

Write-Host $Messages.Export.Info -ForegroundColor Green
$Path = Read-Host -Prompt $Messages.Export.Prompt
$Logs | Export-Csv $Path -NoTypeInformation -Encoding UTF8

```

Script explanation

The script can be used directly without any modifications. The admin is asked about the expiration date and whether they would like to see already expired secrets or certificates or not.

The "Add-Member" command is responsible for creating the columns in the CSV file. You can modify the "\$Path" variable directly in PowerShell, with a CSV file path, in case you'd prefer the export to be non-interactive.

[] Expand table

Command	Notes
Get-MgApplication	Retrieves an application from your directory.
Get-MgApplicationOwner	Retrieves the owners of an application from your directory.

Next steps

For more information on the Microsoft Graph PowerShell module, see [Microsoft Graph PowerShell module overview](#).

For other PowerShell examples for Application Management, see [Azure Microsoft Graph PowerShell examples for Application Management](#).

 **Note:** The author created this article with assistance from AI. [Learn more](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

PowerShell sample: Export secrets and certificates for enterprise apps

Article • 01/23/2025

This PowerShell script example exports all secrets, certificates, and owners for the specified enterprise apps from your directory into a CSV file.

If you don't have an [Azure subscription](#), create an [Azure free account](#) before you begin.

This sample requires the [Microsoft Graph PowerShell](#) SDK module.

Sample script

PowerShell

```
<#####
#####
#DISCLAIMER:
```

This is not an official PowerShell Script. We designed it specifically for the situation you have encountered right now.

Please do not modify or change any preset parameters.

Please note that we will not be able to support the script if it's changed or altered in any way or used in a different situation for other means.

This code-sample is provided "AS IS" without warranty of any kind, either expressed or implied, including but not limited to the implied warranties of merchantability and/or fitness for a particular purpose.

This sample is not supported under any Microsoft standard support program or service.

Microsoft further disclaims all implied warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose.

The entire risk arising out of the use or performance of the sample and documentation remains with you.

In no event shall Microsoft, its authors, or anyone else involved in the

```
creation, production, or  
delivery of the script be liable for any damages whatsoever (including,  
without limitation, damages  
for loss of business profits, business interruption, loss of business  
information, or other  
pecuniary loss) arising out of the use of or inability to use the sample or  
documentation, even if  
Microsoft has been advised of the possibility of such damages.
```

```
#####
#####>
```

```
Connect-MgGraph -Scopes 'Application.Read.All'
```

```
$Messages = @{
    DurationNotice = @{
        Info = @(
            'The operation is running and will take longer the more
applications the tenant has...'
            'Please wait...'
        ) -join ' '
    }
    Export = @{
        Info = 'Where should the CSV file export to?'
        Prompt = 'Enter the full path in the format of <C:\Users\
<USER>\Desktop\Users.csv>'
    }
}
```

```
Write-Host $Messages.DurationNotice.Info -ForegroundColor Yellow
```

```
$EnterpriseApps = Get-MgServicePrincipal -all
```

```
$Logs = @()
```

```
foreach ($EnterpriseApp in $EnterpriseApps) {
    $AppName = $EnterpriseApp.DisplayName
    $AppID = $EnterpriseApp.Id
    $ApplID = $EnterpriseApp.AppId

    $AppCreds = Get-MgServicePrincipal -ServicePrincipalId $AppID |
        Select-Object PasswordCredentials, KeyCredentials
```

```
$Secrets = $AppCreds.PasswordCredentials
$Certs = $AppCreds.KeyCredentials
```

```
#####
#####
```

```
$Logs += [PSCustomObject]@{
    'ApplicationName'      = $AppName
    'ApplicationID'        = $ApplID
    'Secret Name'          = $Null
    'Secret Start Date'    = $Null
    'Secret End Date'      = $Null
    'Certificate Name'     = $Null
    'Certificate Start Date' = $Null
```

```

'Certificate End Date'      = $Null
'Owner'                     = $Null
'Owner_ObjectID'           = $Null
}
#####
foreach ($Secret in $Secrets) {
    $StartDate = $Secret.StartDateTime
    $EndDate   = $Secret.EndDateTime

    $Owner     = Get-MgServicePrincipalOwner -ServicePrincipalId
$EnterpriseApp.Id
    $Username = $Owner.AdditionalProperties.userPrincipalName -join ';'
    $OwnerID  = $Owner.Id -join ';'

    if ($null -eq $Owner.AdditionalProperties.userPrincipalName) {
        $Username = @(
            $Owner.AdditionalProperties.displayName
            '**<This is an Application>**'
        ) -join ' '
    }
    if ($null -eq $Owner.AdditionalProperties.displayName) {
        $Username = '<<No Owner>>'
    }

    $Logs += [PSCustomObject]@{
        'ApplicationName'      = $AppName
        'ApplicationID'        = $AppID
        'Secret Name'          = $SecretName
        'Secret Start Date'    = $StartDate
        'Secret End Date'      = $EndDate
        'Certificate Name'     = $Null
        'Certificate Start Date' = $Null
        'Certificate End Date' = $Null
        'Owner'                 = $Username
        'Owner_ObjectID'       = $OwnerID
    }
}

foreach ($Cert in $Certs) {
    $StartDate = $Cert.StartDateTime
    $EndDate   = $Cert.EndDateTime
    $CertName  = $Cert.DisplayName

    $Owner     = Get-MgServicePrincipalOwner -ServicePrincipalId
$EnterpriseApp.Id
    $Username = $Owner.AdditionalProperties.userPrincipalName -join ';'
    $OwnerID  = $Owner.Id -join ';'

    if ($null -eq $Owner.AdditionalProperties.userPrincipalName) {
        $Username = @(
            $Owner.AdditionalProperties.displayName
            '**<This is an Application>**'
        ) -join ' '
    }
    if ($null -eq $Owner.AdditionalProperties.displayName) {

```

```

        $Username = '<<No Owner>>'
    }

    $Logs += [PSCustomObject]@{
        'ApplicationName'      = $AppName
        'ApplicationID'        = $ApplID
        'Secret Name'          = $Null
        'Certificate Name'     = $CertName
        'Certificate Start Date' = $StartDate
        'Certificate End Date' = $EndDate
        'Owner'                = $Username
        'Owner_ObjectID'       = $OwnerID
        'Secret Start Date'    = $Null
        'Secret End Date'      = $Null
    }
}

Write-Host $Messages.Export.Info -ForegroundColor Green
$Path = Read-Host -Prompt $Messages.Export.Prompt
$Logs | Export-Csv $Path -NoTypeInformation -Encoding UTF8

```

Script explanation

The script can be used directly without any modifications. The admin is asked about the expiration date and whether they would like to see already expired secrets or certificates or not.

The "Add-Member" command is responsible for creating the columns in the CSV file. You can modify the "\$Path" variable directly in PowerShell, with a CSV file path, in case you'd prefer the export to be non-interactive.

[] Expand table

Command	Notes
Get-MgServicePrincipal	Retrieves an enterprise application from your directory.
Get-MgServicePrincipalOwner	Retrieves the owners of an enterprise application from your directory.

Next steps

For more information on the Microsoft Graph PowerShell module, see [Microsoft Graph PowerShell module overview](#).

For other PowerShell examples for Application Management, see [Azure Microsoft Graph PowerShell examples for Application Management](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

PowerShell sample: Export app registrations with expiring secrets and certificates

Article • 01/23/2025

This PowerShell script example exports all app registrations with secrets and certificates expiring in the next X days. It also includes the ones that are expired, if you choose so. The script exports the app registrations along with their owners. It exports the data for the specified apps from your directory. The output is saved in a CSV file.

If you don't have an [Azure subscription](#), create an [Azure free account](#) before you begin.

This sample requires the [Microsoft Graph PowerShell](#) SDK module.

Sample script

PowerShell

```
<#####
#####
#####
```

DISCLAIMER:

This is not an official PowerShell Script. We designed it specifically for the situation you have encountered right now.

Please do not modify or change any preset parameters.

Please note that we will not be able to support the script if it's changed or altered in any way or used in a different situation for other means.

This code-sample is provided "AS IS" without warranty of any kind, either expressed or implied, including but not limited to the implied warranties of merchantability and/or fitness for a particular purpose.

This sample is not supported under any Microsoft standard support program or service.

Microsoft further disclaims all implied warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose.

The entire risk arising out of the use or performance of the sample and documentation remains with you.

In no event shall Microsoft, its authors, or anyone else involved in the creation, production, or delivery of the script be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the sample or documentation, even if Microsoft has been advised of the possibility of such damages.

```
#####
#####>
```

```
Connect-MgGraph -Scopes 'Application.Read.All'
```

```
$Messages = @{
    ExpirationDays = @{
        Info = 'Filter the applications to log by the number of days until their secrets expire.'
        Prompt = 'Enter the number of days until the secrets expire as an integer.'
    }
    AlreadyExpired = @{
        Info = 'Would you like to see Applications with already expired secrets as well?'
        Prompt = 'Enter Yes or No'
    }
    DurationNotice = @{
        Info = @(
            'The operation is running and will take longer the more applications the tenant has...'
            'Please wait...'
        ) -join ' '
    }
    Export = @{
        Info = 'Where should the CSV file export to?'
        Prompt = 'Enter the full path in the format of <C:\Users\<USER>\Desktop\Users.csv>'
    }
}
```

```
Write-Host $Messages.ExpirationDays.Info -ForegroundColor Green
$DaysUntilExpiration = Read-Host -Prompt $Messages.ExpirationDays.Prompt
```

```
Write-Host $Messages.AlreadyExpired.Info -ForegroundColor Green
$IncludeAlreadyExpired = Read-Host -Prompt $Messages.AlreadyExpired.Prompt
```

```
$Now = Get-Date
```

```
Write-Host $Messages.DurationNotice.Info -ForegroundColor yellow
```

```

$Applications = Get-MgApplication -all

$Logs = @()

foreach ($App in $Applications) {
    $AppName = $App.DisplayName
    $AppID   = $App.Id
    $ApplID  = $App.AppId

    $AppCreds = Get-MgApplication -ApplicationId $AppID | 
        Select-Object PasswordCredentials, KeyCredentials

    $Secrets = $AppCreds.PasswordCredentials
    $Certs   = $AppCreds.KeyCredentials

    foreach ($Secret in $Secrets) {
        $StartDate  = $Secret.StartDateTime
        $EndDate    = $Secret.EndDateTime
        $SecretName = $Secret.DisplayName

        $Owner      = Get-MgApplicationOwner -ApplicationId $App.Id
        $Username  = $Owner.AdditionalProperties.userPrincipalName -join ';'
        $OwnerID   = $Owner.Id -join ','

        if ($null -eq $Owner.AdditionalProperties.userPrincipalName) {
            $Username = @(
                $Owner.AdditionalProperties.displayName
                '**<This is an Application>**'
            ) -join ' '
        }
        if ($null -eq $Owner.AdditionalProperties.displayName) {
            $Username = '<<No Owner>>'
        }
    }

    $RemainingDaysCount = ($EndDate - $Now).Days

    if ($IncludeAlreadyExpired -eq 'No') {
        if ($RemainingDaysCount -le $DaysUntilExpiration -and
$RemainingDaysCount -ge 0) {
            $Logs += [PSCustomObject]@{
                'ApplicationName'      = $AppName
                'ApplicationID'        = $ApplID
                'Secret Name'          = $SecretName
                'Secret Start Date'   = $StartDate
                'Secret End Date'     = $EndDate
                'Certificate Name'    = $Null
                'Certificate Start Date' = $Null
                'Certificate End Date' = $Null
                'Owner'                = $Username
                'Owner_ObjectID'       = $OwnerID
            }
        }
    } elseif ($IncludeAlreadyExpired -eq 'Yes') {
        if ($RemainingDaysCount -le $DaysUntilExpiration) {
            $Logs += [PSCustomObject]@{

```

```

        'ApplicationName'      = $AppName
        'ApplicationID'        = $ApplID
        'Secret Name'          = $SecretName
        'Secret Start Date'    = $StartDate
        'Secret End Date'      = $EndDate
        'Certificate Name'     = $Null
        'Certificate Start Date' = $Null
        'Certificate End Date' = $Null
        'Owner'                 = $Username
        'Owner_ObjectID'       = $OwnerID
    }
}
}

foreach ($Cert in $Certs) {
    $StartDate = $Cert.StartDateTime
    $EndDate   = $Cert.EndDateTime
    $CertName  = $Cert.DisplayName

    $Owner      = Get-MgApplicationOwner -ApplicationId $App.Id
    $Username   = $Owner.AdditionalProperties.userPrincipalName -join ';'
    $OwnerID   = $Owner.Id -join ';'

    if ($null -eq $Owner.AdditionalProperties.userPrincipalName) {
        $Username = @(
            $Owner.AdditionalProperties.displayName
            '**<This is an Application>**'
        ) -join ' '
    }
    if ($null -eq $Owner.AdditionalProperties.displayName) {
        $Username = '<<No Owner>>'
    }

    $RemainingDaysCount = ($EndDate - $Now).Days

    if ($IncludeAlreadyExpired -eq 'No') {
        if ($RemainingDaysCount -le $DaysUntilExpiration -and
$RemainingDaysCount -ge 0) {
            $Logs += [PSCustomObject]@{
                'ApplicationName'      = $AppName
                'ApplicationID'        = $ApplID
                'Secret Name'          = $Null
                'Certificate Name'     = $CertName
                'Certificate Start Date' = $StartDate
                'Certificate End Date' = $EndDate
                'Owner'                 = $Username
                'Owner_ObjectID'       = $OwnerID
                'Secret Start Date'    = $Null
                'Secret End Date'      = $Null
            }
        }
    } elseif ($IncludeAlreadyExpired -eq 'Yes') {
        if ($RemainingDaysCount -le $DaysUntilExpiration) {
            $Logs += [PSCustomObject]@{

```

```

        'ApplicationName'      = $AppName
        'ApplicationID'       = $AppID
        'Secret Name'         = $Null
        'Certificate Name'    = $CertName
        'Certificate Start Date' = $StartDate
        'Certificate End Date' = $EndDate
        'Owner'                = $Username
        'Owner_ObjectID'      = $OwnerID
        'Secret Start Date'   = $Null
        'Secret End Date'     = $Null
    }
}
}
}

Write-Host $Messages.Export.Info -ForegroundColor Green
$Path = Read-Host -Prompt $Messages.Export.Prompt
$Logs | Export-Csv $Path -NoTypeInformation -Encoding UTF8

```

Script explanation

The script can be used directly without any modifications. The admin is asked about the expiration date and whether they would like to see already expired secrets or certificates or not.

The "Add-Member" command is responsible for creating the columns in the CSV file. The "New-Object" command creates an object to be used for the columns in the CSV file export. You can modify the "\$Path" variable directly in PowerShell, with a CSV file path, in case you'd prefer the export to be non-interactive.

[\[+\] Expand table](#)

Command	Notes
Get-MgApplication	Retrieves an application from your directory.
Get-MgApplicationOwner	Retrieves the owners of an application from your directory.

Next steps

For more information on the Microsoft Graph PowerShell module, see [Microsoft Graph PowerShell module overview](#).

For other PowerShell examples for Application Management, see [Azure Microsoft Graph PowerShell examples for Application Management](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

PowerShell sample: Export enterprise apps with expiring secrets and certificates

Article • 01/23/2025

This PowerShell script example exports all enterprise applications with secrets and certificates expiring in the next X days. It also includes the ones that are expired, if you choose so. The script exports the enterprise applications along with their owners. It performs this action for the specified enterprise apps from your directory. The output is saved in a CSV file.

If you don't have an [Azure subscription](#), create an [Azure free account](#) before you begin.

This sample requires the [Microsoft Graph PowerShell](#) SDK module.

Sample script

PowerShell

```
<#####
#####
DISCLAIMER:

This is not an official PowerShell Script. We designed it specifically for
the situation you have
encountered right now.

Please do not modify or change any preset parameters.

Please note that we will not be able to support the script if it's changed
or altered in any way
or used in a different situation for other means.

This code-sample is provided "AS IS" without warranty of any kind, either
expressed or implied,
including but not limited to the implied warranties of merchantability
and/or fitness for a
particular purpose.

This sample is not supported under any Microsoft standard support program or
service.

Microsoft further disclaims all implied warranties including, without
limitation, any implied
```

warranties of merchantability or of fitness for a particular purpose.

The entire risk arising out of the use or performance of the sample and documentation remains with you.

In no event shall Microsoft, its authors, or anyone else involved in the creation, production, or delivery of the script be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the sample or documentation, even if Microsoft has been advised of the possibility of such damages.

```
#####
#####>
```

```
Connect-MgGraph -Scopes 'Application.Read.All'

$Applications = Get-MgServicePrincipal -all
$Logs = @()

$Messages = @{
    ExpirationDays = @{
        Info = 'Filter the applications to log by the number of days until their secrets expire.'
        Prompt = 'Enter the number of days until the secrets expire as an integer.'
    }
    AlreadyExpired = @{
        Info = 'Would you like to see Applications with already expired secrets as well?'
        Prompt = 'Enter Yes or No'
    }
    DurationNotice = @{
        Info = @(
            'The operation is running and will take longer the more applications the tenant has...'
            'Please wait...'
        ) -join ''
    }
    Export = @{
        Info = 'Where should the CSV file export to?'
        Prompt = 'Enter the full path in the format of <C:\Users\<USER>\Desktop\Users.csv>'
    }
}

Write-Host $Messages.ExpirationDays.Info -ForegroundColor Green
$DaysUntilExpiration = Read-Host -Prompt $Messages.ExpirationDays.Prompt

Write-Host $Messages.AlreadyExpired.Info -ForegroundColor Green
$IncludeAlreadyExpired = Read-Host -Prompt $Messages.AlreadyExpired.Prompt
```

```

$Now = Get-Date

Write-Host $Messages.DurationNotice.Info -ForegroundColor yellow

foreach ($App in $Applications) {
    $AppName = $App.DisplayName
    $AppID   = $App.Id
    $ApplID  = $App.AppId

    $AppCreds = Get-MgServicePrincipal -ServicePrincipalId $AppID |
        Select-Object PasswordCredentials, KeyCredentials

    $Secrets = $AppCreds.PasswordCredentials
    $Certs   = $AppCreds.KeyCredentials

    foreach ($Secret in $Secrets) {
        $StartDate = $Secret.StartDateTime
        $EndDate   = $Secret.EndDateTime
        $SecretName = $Secret.DisplayName

        $Owner      = Get-MgServicePrincipalOwner -ServicePrincipalId $App.Id
        $Username  = $Owner.AdditionalProperties.userPrincipalName -join ';'
        $OwnerID   = $Owner.Id -join ','

        if ($null -eq $Owner.AdditionalProperties.userPrincipalName) {
            $Username = @(
                $Owner.AdditionalProperties.displayName
                '**<This is an Application>**'
            ) -join ' '
        }

        if ($null -eq $Owner.AdditionalProperties.displayName) {
            $Username = '<<No Owner>>'
        }
    }

    $RemainingDaysCount = $EndDate - $Now |
        Select-Object -ExpandProperty Days

    if ($IncludeAlreadyExpired -eq 'No') {
        if ($RemainingDaysCount -le $DaysUntilExpiration -and
$RemainingDaysCount -ge 0) {
            $Logs += [PSCustomObject]@{
                'ApplicationName'      = $AppName
                'ApplicationID'        = $ApplID
                'Secret Name'          = $SecretName
                'Secret Start Date'   = $StartDate
                'Secret End Date'     = $EndDate
                'Certificate Name'    = $Null
                'Certificate Start Date' = $Null
                'Certificate End Date' = $Null
                'Owner'                = $Username
                'Owner_ObjectID'       = $OwnerID
            }
        }
    }
}

```

```

} elseif ($IncludeAlreadyExpired -eq 'Yes') {
    if ($RemainingDaysCount -le $DaysUntilExpiration) {
        $Logs += [pscustomobject]@{
            'ApplicationName'      = $AppName
            'ApplicationID'        = $AppID
            'Secret Name'          = $SecretName
            'Secret Start Date'    = $StartDate
            'Secret End Date'      = $EndDate
            'Certificate Name'     = $Null
            'Certificate Start Date' = $Null
            'Certificate End Date' = $Null
            'Owner'                 = $Username
            'Owner_ObjectID'       = $OwnerID
        }
    }
}

foreach ($Cert in $Certs) {
    $StartDate = $Cert.StartDateTime
    $EndDate   = $Cert.EndDateTime
    $CertName  = $Cert.DisplayName

    $RemainingDaysCount = $EndDate - $Now |
        Select-Object -ExpandProperty Days

    $Owner      = Get-MgServicePrincipalOwner -ServicePrincipalId $App.Id
    $Username   = $Owner.AdditionalProperties.userPrincipalName -join ';'
    $OwnerID   = $Owner.Id -join ';'

    if ($null -eq $Owner.AdditionalProperties.userPrincipalName) {
        $Username = @(
            $Owner.AdditionalProperties.displayName
            '**<This is an Application>**'
        ) -join ' '
    }
    if ($null -eq $Owner.AdditionalProperties.displayName) {
        $Username = '<<No Owner>>'
    }

    if ($IncludeAlreadyExpired -eq 'No') {
        if ($RemainingDaysCount -le $DaysUntilExpiration -and
$RemainingDaysCount -ge 0) {
            $Logs += [pscustomobject]@{
                'ApplicationName'      = $AppName
                'ApplicationID'        = $AppID
                'Secret Name'          = $Null
                'Certificate Name'     = $CertName
                'Certificate Start Date' = $StartDate
                'Certificate End Date' = $EndDate
                'Owner'                 = $Username
                'Owner_ObjectID'       = $OwnerID
                'Secret Start Date'    = $Null
                'Secret End Date'      = $Null
            }
        }
    }
}

```

```

        }
    }
} elseif ($IncludeAlreadyExpired -eq 'Yes') {
    if ($RemainingDaysCount -le $DaysUntilExpiration) {
        $Logs += [pscustomobject]@{
            'ApplicationName'      = $AppName
            'ApplicationID'        = $AppID
            'Certificate Name'     = $CertName
            'Certificate Start Date' = $StartDate
            'Certificate End Date' = $EndDate
            'Owner'                 = $Username
            'Owner_ObjectID'       = $OwnerID
            'Secret Start Date'    = $Null
            'Secret End Date'      = $Null
        }
    }
}
}

Write-Host $Messages.Export.Info -ForegroundColor Green
$Path = Read-Host -Prompt $Messages.Export.Prompt
$Logs | Export-Csv $Path -NoTypeInformation -Encoding UTF8

```

Script explanation

The script can be used directly without any modifications. The admin is asked about the expiration date and whether they would like to see already expired secrets or certificates or not.

The "Add-Member" command is responsible for creating the columns in the CSV file. The "New-Object" command creates an object to be used for the columns in the CSV file export. You can modify the "\$Path" variable directly in PowerShell, with a CSV file path, in case you'd prefer the export to be non-interactive.

[+] Expand table

Command	Notes
Get-MgServicePrincipal	Retrieves an enterprise application from your directory.
Get-MgServicePrincipalOwner	Retrieves the owners of an enterprise application from your directory.

Next steps

For more information on the Microsoft Graph PowerShell module, see [Microsoft Graph PowerShell module overview](#).

For other PowerShell examples for Application Management, see [Azure Microsoft Graph PowerShell examples for Application Management](#).

Feedback

Was this page helpful?



Yes



No

[Provide product feedback](#) ↗

PowerShell sample: Export apps with secrets and certificates expiring beyond the required date

Article • 01/23/2025

This PowerShell script example exports all app registrations' secrets and certificates expiring beyond a required period. It performs this task for the specified apps from your directory. The script runs non-interactively. The output is saved in a CSV file.

If you don't have an [Azure subscription](#), create an [Azure free account](#) before you begin.

Sample script

PowerShell

```
<#####
#####
```

```
#####
```

```
DISCLAIMER:
```

This is not an official PowerShell Script. We designed it specifically for the situation you have encountered right now.

Please do not modify or change any preset parameters.

Please note that we will not be able to support the script if it's changed or altered in any way or used in a different situation for other means.

This code-sample is provided "AS IS" without warranty of any kind, either expressed or implied, including but not limited to the implied warranties of merchantability and/or fitness for a particular purpose.

This sample is not supported under any Microsoft standard support program or service.

Microsoft further disclaims all implied warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose.

The entire risk arising out of the use or performance of the sample and documentation remains with you.

In no event shall Microsoft, its authors, or anyone else involved in the creation, production, or delivery of the script be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the sample or documentation, even if Microsoft has been advised of the possibility of such damages.

```
#####
####>

$loginURL = 'https://login.microsoftonline.com'
$resource = 'https://graph.microsoft.com'

#PARAMETERS TO CHANGE
$ClientID      = 'App ID'
$ClientSecret   = 'APP Secret'
$TenantName     = 'TENANT.onmicrosoft.com'

$Months = 'Number of months'
$Path   = 'add a path here\File.csv'
#####
#Repeating Function to get an Access Token based on the parameters:
function Get-RefreshedToken($LoginURL, $ClientID, $ClientSecret,
$TenantName) {
    $RequestParameters = @{
        Method = 'POST'
        Uri    = "$LoginURL/$TenantName/oauth2/v2.0/token"
        Body   = @{
            grant_type      = 'client_credentials'
            client_id       = $ClientID
            client_secret   = $ClientSecret
            scope           = 'https://graph.microsoft.com/.default'
        }
    }
    Invoke-RestMethod @RequestParameters
}

#BUILD THE ACCESS TOKEN
$RefreshParameters = @{
    LoginURL      = $loginURL
    ClientID      = $ClientID
    ClientSecret   = $ClientSecret
    TenantName     = $TenantName
}
$OAuth      = Get-RefreshedToken @RefreshParameters
$Identity  = $OAuth.access_token

#####
$HeaderParams = @{


```

```

'Authorization' = "$(OAuth.token_type) $($Identity)"
}

$AppsSecrets = 'https://graph.microsoft.com/v1.0/applications'

$ApplicationsList = Invoke-WebRequest -Headers $HeaderParams -Uri
$AppsSecrets -Method GET

$Logs      = @()
$NextCounter = 0

do {
    $ApplicationEvents = $ApplicationsList.Content |
        ConvertFrom-Json |
        Select-Object -ExpandProperty value

    foreach ($ApplicationEvent in $ApplicationEvents) {
        $IDs      = $ApplicationEvent.id
        $AppName = $ApplicationEvent.displayName
        $AppID   = $ApplicationEvent.appId
        $Secrets = $ApplicationEvent.passwordCredentials

        $NextCounter++

        foreach ($Secret in $Secrets) {
            $StartDate      = $Secret.startDateTime
            $EndDate        = $Secret.endDateTime
            $pos           = $StartDate.IndexOf('T')
            $LeftPart       = $StartDate.Substring(0, $pos)
            $Position       = $EndDate.IndexOf('T')
            $LeftPartEnd    = $EndDate.Substring(0, $pos)
            $DateStringStart = [Datetime]::ParseExact($LeftPart, 'yyyy-MM-
dd', $null)
            $DateStringEnd   = [Datetime]::ParseExact($LeftPartEnd, 'yyyy-
MM-dd', $null)
            $OptimalDate     = $DateStringStart.AddMonths($Months)

            if ($OptimalDate -lt $DateStringEnd) {
                $Log = [PSCustomObject]@{
                    'Application'      = $AppName
                    'AppID'            = $AppID
                    'Secret Start Date' = $DateStringStart
                    'Secret End Date'  = $DateStringEnd
                }
            }

            $OwnerRequestParams = @{
                Headers = $HeaderParams
                Uri     =
"https://graph.microsoft.com/v1.0/applications/$IDs/owners"
                Method  = 'GET'
            }
            $ApplicationsOwners = Invoke-WebRequest @OwnerRequestParams

            $Users = $ApplicationsOwners.Content |
                ConvertFrom-Json |
                Select-Object -ExpandProperty value
        }
    }
}

```

```

        foreach ($User in $Users) {
            $Owner = $User.displayname
            $Log | Add-Member -MemberType NoteProperty -Name
'AppOwner' -Value $Owner
        }

        $Logs += $Log
    }
}

If ($NextCounter -eq 100) {
    $OData = $ApplicationsList.Content | ConvertFrom-Json
    $AppsSecrets = $OData.'@odata.nextLink'
    try {
        $ListRequestParams = @{
            UseBasicParsing = $true
            Headers         = $HeaderParams
            Uri             = $AppsSecrets
            Method          = 'GET'
            ContentType     = 'application/Json'
        }
        $ApplicationsList = Invoke-WebRequest @ListRequestParams
    } catch {
        $_
    }
}

$NextCounter = 0

Start-Sleep -Seconds 1
}
}

} while ($AppsSecrets -ne $null)

$Logs | Export-Csv $Path -NoTypeInformation -Encoding UTF8

```

Script explanation

This script is working non-interactively. The admin using it needs to change the values in the "#PARAMETERS TO CHANGE" section. They need to enter their own App ID, Application Secret, and Tenant Name. They also need to specify the period for the apps' credentials expiration. Finally, they need to set the path where the CSV is exported.

This script uses the [Client_Credential Oauth Flow](#) The function "RefreshToken" builds the access token based on the values of the parameters modified by the admin.

The "Add-Member" command is responsible for creating the columns in the CSV file.

Command	Notes
Invoke-WebRequest	Sends HTTP and HTTPS requests to a web page or web service. It parses the response and returns collections of links, images, and other significant HTML elements.

Next steps

For more information on the Microsoft Graph PowerShell module, see [Microsoft Graph PowerShell overview](#).

For other PowerShell examples for Application Management, see [Microsoft Graph PowerShell examples for Application Management](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Overview of the Microsoft Entra application gallery

Article • 12/06/2024

The Microsoft Entra application gallery is a collection of software as a service (SaaS) [applications](#) that are preintegrated with Microsoft Entra ID. The collection contains thousands of applications that make it easy to deploy and configure [single sign-on \(SSO\)](#) and [automated user provisioning](#).

To find the gallery when signed into your tenant, browse to **Identity > Applications > Enterprise applications > All applications > New application**.

The applications available from the gallery follow the SaaS model that allows users to connect to and use cloud-based applications over the Internet. Common examples are email, calendaring, and office tools (such as Microsoft Office 365).

The following are benefits of using applications available in the gallery:

- Users find the best possible SSO experience for the application.
- Configuration of the application is simple and minimal.
- A quick search finds the needed application.
- Free, Basic, and Premium Microsoft Entra users can all use the application.
- Users can easily find [step-by-step configuration tutorials](#) that are available for onboarding gallery applications.

Applications in the gallery

The gallery contains thousands of applications that are preintegrated into Microsoft Entra ID. When using the gallery, you choose from using applications from specific cloud platforms, featured applications, or you search for the application that you want to use.

Search for applications

If you don't find the application that you're looking for in the featured applications, you can search for a specific application by name.

The Microsoft Entra App Gallery is a catalog of thousands of apps that make it easy to deploy and configure single sign-on (SSO) and automated user provisioning. When deploying an app from the App Gallery, you leverage prebuilt templates to connect your users more securely to their apps. Browse or create your own application here. If you are wanting to publish an application you have developed into the Microsoft Entra Gallery for other organizations to discover and use, you can file a request using the process described in [this article](#).

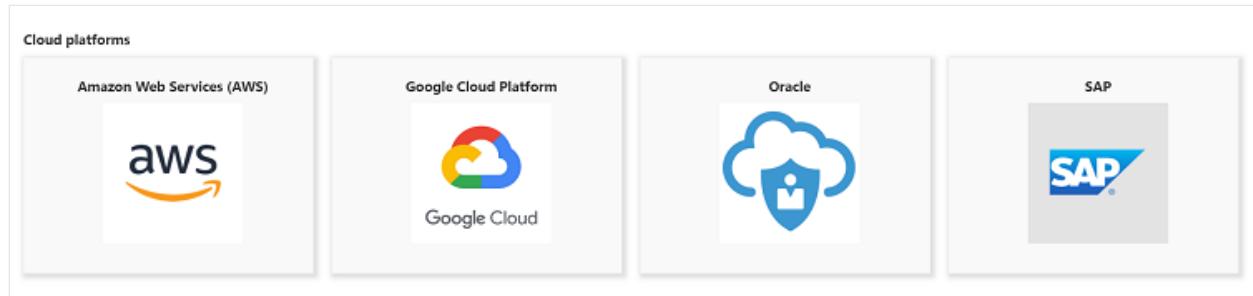


When searching for an application, you can also specify specific filters, such as single sign-on options, automated provisioning, and categories.

- **Single sign-on options** – You can search for applications that support these SSO options: SAML, OpenID Connect (OIDC), Password, or Linked. For more information about these options, see [Plan a single sign-on deployment in Microsoft Entra ID](#).
- **User account management** – The only option available is [automated provisioning](#).
- **Categories** – When an application is added to the gallery it can be classified in a specific category. Many categories are available such as **Business management**, **Collaboration**, or **Education**.

Cloud platforms

Applications that are specific to major cloud platforms, such as AWS, Google, or Oracle can be found by selecting the appropriate platform.



On-premises applications

There are five ways on-premises applications can be connected to Microsoft Entra ID. One is using Microsoft Entra application proxy for single sign-on. If your application supports single-sign on via SAML or Kerberos, then from the on-premises section of the Microsoft Entra gallery, you can undertake the following tasks:

- Configure Application Proxy to enable remote access to an on-premises application.
- Use the documentation to learn more about how to use Application Proxy to secure remote access to on-premises applications.
- Manage any private network connectors that you created.

The screenshot shows the 'On-premises applications' section of the Microsoft Entra ID application gallery. It includes three cards:

- Add an on-premises application**: Configure Microsoft Entra application proxy to enable secure remote access.
- Learn about Application Proxy**: Learn how to use Application Proxy to provide secure remote access to your on-premises applications.
- On-premises application provisioning**: Automate creating, updating, and deleting users in applications on-premises or in a virtual machine.

If your application uses Kerberos and also requires group memberships, then you can populate Windows Server AD groups from corresponding groups in Microsoft Entra. For more information, see [group writeback with Microsoft Entra Cloud Sync](#).

The second is using the provisioning agent to provision to an on-premises application that has its own user store and doesn't rely upon Windows Server AD. You can configure provisioning to [on-premises applications that support SCIM](#), that use [SQL databases](#), that use an [LDAP directory](#), or support a [SOAP or REST provisioning API](#).

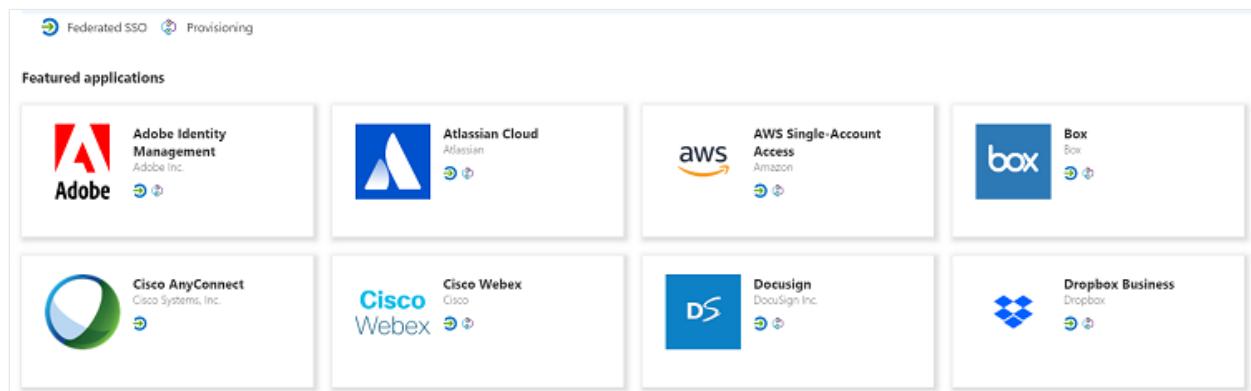
The third is using Microsoft Entra Private Access, by configuring a Global Secure Access app for per-app connections. For more information, see [Learn about Microsoft Entra Private Access](#).

The fourth is to use the application's own connector. If you have [SAP S/4HANA On-premise](#), then provision users from Microsoft Entra ID to SAP Cloud Identity Directory. SAP Cloud Identity Services then provisions the users that are in the SAP Cloud Identity Directory into the downstream SAP applications, such as [SAP S/4HANA On-Premise](#), through the SAP cloud connector. For more information, see [plan deploying Microsoft Entra for user provisioning with SAP source and target apps](#).

The fifth is to use a third party integration technology. In cases where an application doesn't support standards such as SCIM, partners have custom ECMA connectors and SCIM gateways to integrate Microsoft Entra ID with more applications, including on-premises applications. For more information, see the list of [available partner-driven integrations](#).

Featured applications

A collection of featured applications is listed by default when you open the Microsoft Entra gallery. Each application is marked with a symbol to enable you to identify whether it supports federated SSO or automated provisioning.



- **Federated SSO** - When you set up [SSO](#) to work between multiple identity providers, it results to federation. An SSO implementation based on federation

protocols improves security, reliability, user experiences, and implementation. Some applications implement federated SSO as SAML-based or as OIDC-based. For SAML applications, when you select create, the application is added to your tenant. For OIDC applications, the administrator must first sign up or sign-in on the application's website to add the application to Microsoft Entra ID.

- **Provisioning** - Microsoft Entra ID to SaaS [application provisioning](#) refers to automatically creating user identities and roles in the SaaS applications that users need access to.

Create your own application

When you select the **Create your own application** link near the top of the pane, you see a new pane that lists the following choices:

- **Register an application to integrate with Microsoft Entra ID (App you're developing)** – This choice is meant for developers who want to work on the integration of their application that uses OpenID Connect with Microsoft Entra ID. This choice doesn't provide an opportunity to publish your application to the gallery. It's only for development purposes to work on integration.
- **Integrate any other application you don't find in the gallery (Non-gallery)** – This choice is meant for an administrator to make a SAML-based application that isn't in the gallery available to users in their organization. By integrating the application, the administrator can configure, secure, and monitor its use. This choice doesn't provide a way to publish the application to the gallery. It does provide secure access to the application for users in your tenant.
- **Configure Application Proxy for secure remote access to an on-premises application** – This choice is meant for an administrator to enable SSO and secure remote access for web applications hosted on-premises by connecting with Application Proxy.

Request new gallery application

After you successfully integrate an application with Microsoft Entra ID and thoroughly tested it, you file a request for it to be added to the gallery. Publishing an application to the gallery from the portal isn't supported but there's a process that you can follow to request it to be added. For more information about publishing to the gallery, select [Request new gallery application](#).

Next steps

- Get started by adding your first enterprise application with the [Quickstart: Add an enterprise application](#).
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Submit a request to publish your application in Microsoft Entra application gallery

Article • 10/09/2024

You can publish applications you develop in the Microsoft Entra application gallery, which is a catalog of thousands of apps. When you publish your applications, they're made publicly available for users to add to their tenants. For more information, see [Overview of the Microsoft Entra application gallery](#).

To publish your application in the Microsoft Entra application gallery, you need to complete the following tasks:

- Make sure that you complete the prerequisites.
- Create and publish documentation.
- Submit your application.
- Join the Microsoft partner network.

Note

We are currently not accepting new SSO or provisioning requests while we focus on the [Secure Future Initiative](#). Update requests will be processed on a case-by-case basis.

Prerequisites

To publish your application in the gallery, you must first read and agree to specific [terms and conditions](#).

- Implement support for *single sign-on (SSO)*. To learn more about supported options, see [Plan a single sign-on deployment](#).
 - For password SSO, make sure that your application supports form authentication so that password vaulting can be used.
 - For federated applications (SAML/WS-Fed), the application should preferably support [software-as-a-service \(SaaS\) model](#) but it is not mandatory and it can be an on-premises application as well. Enterprise gallery applications must support multiple user configurations and not any specific user.

- For OpenID Connect, most applications work well as a multitenant application implementing the [Microsoft Entra consent framework](#). Refer to [this link](#) to convert the application into multitenant. If your application requires additional per-instance configuration, such as customers needing to control their own secrets and certificates, you can publish a single-tenant Open ID Connect application.
- Provisioning is optional yet highly recommended. To learn more about Microsoft Entra SCIM, see [build a SCIM endpoint and configure user provisioning with Microsoft Entra ID](#).

You can sign up for a free, test Development account. It's free for 90 days and you get all of the premium Microsoft Entra features with it. You can also extend the account if you use it for development work: [Join the Microsoft 365 Developer Program](#).

Create and publish documentation

Provide app documentation for your site

Ease of adoption is an important factor for those that make decisions about enterprise software. Documentation that is clear and easy to follow helps your users adopt technology and it reduces support costs.

Create documentation that includes the following information at minimum:

- An introduction to your SSO functionality
 - Protocols
 - Version and SKU
 - List of supported identity providers with documentation links
- Licensing information for your application
- Role-based access control for configuring SSO
- SSO Configuration Steps
 - UI configuration elements for SAML with expected values from the provider
 - Service provider information to be passed to identity providers
- If you use OIDC/OAuth, a list of permissions required for consent, with business justifications
- Testing steps for pilot users
- Troubleshooting information, including error codes and messages
- Support mechanisms for users
- Details about your SCIM endpoint, including supported resources and attributes

App documentation on the Microsoft site

When your SAML application is added to the gallery, documentation is created that explains the step-by-step process. For an example, see [Tutorials for integrating SaaS applications with Microsoft Entra ID](#). This documentation is created based on your submission to the gallery. You can easily update the documentation if you make changes to your application by using your GitHub account.

For OIDC application, there is no application specific documentation, we have only the generic [tutorial](#) for all the OpenID Connect applications.

Submit your application

After you've tested that your application works with Microsoft Entra ID, submit your application request in the [Microsoft Application Network portal](#).

If you see a "Request Access" page, then fill in the business justification and select **Request Access**.

After your account is added, you can sign in to the Microsoft Application Network portal and submit the request by selecting the **Submit Request (ISV)** tile on the home page. If you see the "Your sign-in was blocked" error while logging in, see [Troubleshoot sign-in to the Microsoft Application Network portal](#).

Implementation-specific options

On the application **Registration** form, select the feature that you want to enable. Select **OpenID Connect & OAuth 2.0, SAML 2.0/WS-Fed, or Password SSO(UserName & Password)** depending on the feature that your application supports.

If you're implementing a **SCIM** 2.0 endpoint for user provisioning, select **User Provisioning (SCIM 2.0)**. Download the schema to provide in the onboarding request. For more information, see [Export provisioning configuration and roll back to a known good state](#). The schema that you configured is used when testing the non-gallery application to build the gallery application.

If you wish to register an MDM application in the Microsoft Entra application gallery, select **Register an MDM app**.

You can track application requests by customer name at the Microsoft Application Network portal. For more information, see [Application requests by Customers](#).

Update or Remove the application from the Gallery

You can submit your application update request in the [Microsoft Application Network portal](#).

If you see a "Request Access" page, then fill in the business justification and select **Request Access**.

After the account is added, you can sign in to the Microsoft Application Network portal and submit the request by selecting the **Submit Request (ISV)** tile on the home page and select **Update my application's listing in the gallery** and select one of the following options as per your choice -

- If you want to update an application's SSO feature, select **Update my application's Federated SSO feature**.
- If you want to update Password SSO feature, select **Update my application's Password SSO feature**.
- If you want to upgrade your listing from Password SSO to Federated SSO, select **Upgrade my application from Password SSO to Federated SSO**.
- If you want to update an MDM listing, select **Update my MDM app**.
- If you want to update an existing User Provisioning integration, select **Improve my application's User Provisioning feature**.
- If you want to remove the application from Microsoft Entra application gallery, select **Remove my application listing from the gallery**.

If you see the **Your sign-in was blocked** error while logging in, see [Troubleshoot sign-in to the Microsoft Application Network portal](#).

Join the Microsoft partner network

The Microsoft Partner Network provides instant access to exclusive programs, tools, connections, and resources. To join the network and create your go-to-market plan, see [Reach commercial customers](#).

Next steps

- Learn more about managing enterprise applications with [What is application management in Microsoft Entra ID?](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Integrating Microsoft Entra ID with applications getting started guide

Article • 12/06/2024

This article summarizes the process for integrating applications with Microsoft Entra ID. Each of the following sections contains a brief summary of a more detailed article so you can identify which parts of this getting started guide are relevant to you.

To download in-depth deployment plans, see [Next steps](#).

Take inventory

Before integrating applications with Microsoft Entra ID, it's important to know where you are and where you want to go. The following questions are intended to help you think about your Microsoft Entra application integration project.

Application inventory

- Where are all of your applications? Who owns them?
- What kind of authentication do your applications require?
- Who needs access to which applications?
- Do you want to deploy a new application?
 - Will you build it in-house and deploy it on an Azure compute instance?
 - Will you use one that is available in the Azure Application Gallery?

User and group inventory

- Where do your user accounts reside?
 - On-premises Active Directory
 - Microsoft Entra ID
 - Within a separate application database that you own
 - In unsanctioned applications
 - All of the listed options
- What permissions and role assignments do individual users currently have? Do you need to review their access or are you sure that your user access and role assignments are appropriate now?
- Are groups already established in your on-premises Active Directory?
 - How are your groups organized?
 - Who are the group members?

- What permissions/role assignments do the groups currently have?
- Will you need to clean up user/group databases before integrating? (This is an important question. Garbage in, garbage out.)

Access management inventory

- How do you currently manage user access to applications? Does that need to change? Have you considered other ways to manage access, such as with [Azure RBAC](#) for example?
- Who needs access to what?

Maybe you don't have the answers to all of these questions up front but that's okay. This guide can help you answer some of those questions and make some informed decisions.

Find unsanctioned cloud applications with Cloud Discovery

As mentioned the previous section, there might be applications that your organization manages until now. As part of the inventory process, it's possible to find unsanctioned cloud applications. See [Set up Cloud Discovery](#).

Integrating applications with Microsoft Entra ID

The following articles discuss the different ways applications integrate with Microsoft Entra ID, and provide some guidance.

- [Determining which Active Directory to use](#)
- [Using applications in the Azure application gallery](#)
- [Integrating SaaS applications tutorials list](#)

Capabilities for apps not listed in the Microsoft Entra gallery

You can add any application that already exists in your organization, or any third-party application from a vendor who isn't already part of the Microsoft Entra gallery.

Depending on your [license agreement](#), the following capabilities are available:

- Self-service integration of any application that supports [Security Assertion Markup Language \(SAML\) 2.0](#) identity providers (SP-initiated or IdP-initiated)

- Self-service integration of any web application that has an HTML-based sign-in page using [password-based SSO](#)
- Self-service connection of applications that use the [System for Cross-Domain Identity Management \(SCIM\) protocol for user provisioning](#)
- Ability to add links to any application in the [Office 365 app launcher](#) or [My Apps](#)

If you're looking for developer guidance on how to integrate custom apps with Microsoft Entra ID, see [Authentication Scenarios for Microsoft Entra ID](#). When you develop an app that uses a modern protocol like [OpenId Connect/OAuth](#) to authenticate users, register it with the Microsoft identity platform. You can register by using the [App registrations](#) experience in the Azure portal.

Authentication Types

Each of your applications might have different authentication requirements. With Microsoft Entra ID, signing certificates can be used with applications that use SAML 2.0, WS-Federation, or OpenID Connect Protocols and Password Single Sign On. For more information about application authentication types, see [Managing certificates for federated single sign-on in Microsoft Entra ID](#) and [Password based single sign on](#).

Enabling SSO with Microsoft Entra application proxy

With Microsoft Entra application proxy, you can provide access to applications located inside your private network securely, from anywhere and on any device. After you install a private network connector within your environment, it can be easily configured with Microsoft Entra ID.

Integrating custom applications

If you want to add your custom application to the Azure Application Gallery, see [Publish your app to the Microsoft Entra app gallery](#).

Managing access to applications

The following articles describe ways you can manage access to applications once they're integrated with Microsoft Entra ID using Microsoft Entra Connectors and Microsoft Entra ID.

- [Managing access to apps using Microsoft Entra ID](#)
- [Automating with Microsoft Entra Connectors](#)
- [Assigning users to an application](#)

- [Assigning groups to an application](#)
- [Sharing accounts](#)

Next steps

For in-depth information, you can download Microsoft Entra deployment plans from [GitHub](#). For gallery applications, you can download deployment plans for single sign-on, Conditional Access, and user provisioning through the [Microsoft Entra admin center](#).

To download a deployment plan from the Microsoft Entra admin center:

1. Sign in to the [Microsoft Entra admin center](#).
2. Select **Enterprise Applications | Pick an App | Deployment Plan**.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Overview of enterprise application ownership in Microsoft Entra ID

Article • 12/06/2024

A user in Microsoft Entra ID is automatically added as an application owner when they register an application. The ownership of an enterprise application is assigned by default only when a user with no administrator roles creates a new application registration. In all other cases, ownership isn't assigned by default to an enterprise application. Users can be owners of enterprise applications but groups can't be assigned as owners.

As an owner of an enterprise application in Microsoft Entra ID, a user can manage the organization-specific configuration of the application, such as single sign-on, provisioning, and user assignment. An owner can also add or remove other owners. Unlike Privileged Role Administrators, owners can manage only the enterprise applications they own. The owners have the same permissions as application administrators scoped to an individual application. To learn more about the permissions that an owner of an application has, see [Ownership permissions](#)

Note

The application may have more permissions than the owner, and thus would be an elevation of privilege over what the owner has access to as a user. An application owner can create or update users or other objects while impersonating the application. The elevation of privilege to owners can raise a security concern in some cases depending on the application's permissions.

FAQ

What do you do with applications where the owner is no longer with the organization?

If you have an ownerless application in your tenant, you can access the audit log for the application to investigate other users who might be involved in configuring the application. However, there are limitations on how long audit logs are stored. See [Microsoft Entra audit log reporting](#).

You might also see other users who scope permissions on the application by navigating to **Roles and Administrators** tab. Once you find the right person to own the application,

a user with a highly privileged administrative role in the organization can assign the new owner for the application. See [Assign enterprise application owners](#).

As a best practice, we recommend proactive monitoring applications in your environment to ensure there are at least two owners, where possible, to avoid the situation of ownerless apps. Additionally, you should utilize the serviceManagementReference property on the application object to reference the team contact information from your enterprise Service or Asset Management Database. The serviceManagementReference property ensures you have team contact even if an individual leaves the organization.

How can I find enterprise applications that are ownerless or at risk of being ownerless in my organization?

To learn how to identify ownerless enterprise apps or apps with only one owner using Microsoft Graph API, see [List ownerless applications](#).

How do you add yourself as an owner of an enterprise application?

Existing owners of an application can add other users as the owners. Also, users with a privileged role such as Application Administrator or the Cloud Application Administrator can assign owners to applications in the organization. If you aren't an administrator, work with an administrator in your organization to [assign you as the owner](#) of the application.

How can you find all the applications that you own?

- You can navigate to [Enterprise Applications](#), then select [All Applications](#)
- Select [Add filter](#), then use [owned by](#) to search for apps owned by you or any other person.

Next steps

- [Assign enterprise application owners](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Properties of an enterprise application

Article • 01/31/2025

This article describes the properties that you can configure for an enterprise application in your Microsoft Entra tenant. To configure the properties, see [Configure enterprise application properties](#).

Enabled for users to sign in?

If this option is set to **Yes**, then assigned users are able to sign in to the application from the My Apps portal, the User access URL, or by navigating to the application URL directly. If assignment is required, then only users who are assigned to the application are able to sign-in. If assignment is required, applications must be assigned to get a token.

If this option is set to **No**, then no users are able to sign in to the application, even if they're assigned to it. Tokens aren't issued for the application. This setting not only prevents users from signing in but also restricts service principals from accessing the application using application permissions.

Name

This property is the name of the application that users see on the My Apps portal. Administrators see the name when they manage access to the application. Other tenants see the name when integrating the application into their directory.

We recommend that you choose a name that users can understand. It's important because this name is visible in the various portals, such as My Apps and Microsoft 365 Launcher.

Homepage URL

If the application is custom-developed, the homepage URL is the URL that a user can use to sign in to the application. For example, it's the URL that is launched when the application is selected in the My Apps portal. If this application is from the Microsoft Entra Gallery, this URL is where you can go to learn more about the application or its vendor.

The homepage URL can't be edited within enterprise applications. The homepage URL must be edited on the application object.

Logo

This property is the application logo that users see on the My Apps portal and the Office 365 application launcher. Administrators also see the logo in the Microsoft Entra gallery.

Custom logos must be exactly 215x215 pixels in size and be in the PNG format. You should use a solid color background with no transparency in your application logo. The logo file size can't be over 100 KB.

Application ID

This property is the unique identifier for the application in your directory. You can use this application ID if you ever need help from Microsoft Support. You can also use the identifier to perform operations using the Microsoft Graph APIs or the Microsoft Graph PowerShell SDK.

Object ID

This ID is the unique identifier of the service principal object associated with the application. This identifier can be useful when performing management operations against this application using PowerShell or other programmatic interfaces. This identifier is different than the identifier for the application object.

The identifier is used to update information for the local instance of the application, such as assigning users and groups to the application. The identifier can also be used to update the properties of the enterprise application or to configure single-sign on.

Assignment required

This setting controls who or what in the directory can obtain an access token for the application. You can use this setting to further lock down access to the application and let only specified users and applications obtain access tokens.

This option determines whether or not an application appears on the My Apps portal. To show the application there, assign an appropriate user or group to the application. This option has no effect on users' access to the application when you configure it for any of the other single sign-on modes.

If this option is set to **Yes**, then users and other applications or services must first be assigned this application before being able to access it.

If this option is set to **No**, then all users are able to sign in, and other applications and services are able to obtain an access token to the application. This option also allows any external users that could be invited into your organization to sign in.

This option only applies to the following types of applications and services:

- Applications using Security Assertion Markup Language (SAML)
- OpenID Connect
- OAuth 2.0
- WS-Federation for user sign
- Application Proxy applications with Microsoft Entra preauthentication enabled
- Applications or services for which other applications or service are requesting access tokens

Visible to users

Makes the application visible in My Apps and the Microsoft 365 Launcher

If this option is set to **Yes**, then assigned users see the application on the My Apps portal and Microsoft 365 app launcher.

If this option is set to **No**, then no users see this application on their My Apps portal and Microsoft 365 launcher.

Make sure that a homepage URL is included or else the application can't be launched from the My Apps portal.

Regardless of whether assignment is required or not, only assigned users are able to see this application in the My Apps portal. If you want certain users to see the application in the My Apps portal, but everyone to be able to access it, assign the users in the **Users and Groups** tab, and set assignment required to **No**.

Notes

You can use this field to add any information that is relevant for the management of the application. The field is a free text field with a maximum size of 1,024 characters.

Next steps

Learn where to go to configure the properties of an enterprise application.

- [Configure enterprise application properties](#)

 **Note:** The author created this article with assistance from AI. [Learn more](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

What is single sign-on in Microsoft Entra ID?

Article • 09/30/2024

This article provides you with information about the single sign-on (SSO) options that are available to you. It also outlines an introduction to planning a single sign-on deployment when using Microsoft Entra ID. Single sign-on is an authentication method that allows users to sign in using one set of credentials to multiple independent software systems. Using SSO means a user doesn't have to sign in to every application they use. With SSO, users can access all needed applications without being required to authenticate using different credentials. For a brief introduction, see [Microsoft Entra single sign-on ↗](#).

Many applications already exist in Microsoft Entra ID that you can use with SSO. You have several options for SSO depending on the needs of the application and how it's implemented. Take time to plan your SSO deployment before you create applications in Microsoft Entra ID. The management of applications can be made easier by using the My Apps portal.

Single sign-on options

Choosing an SSO method depends on how the application is configured for authentication. Cloud applications can use federation-based options, such as OpenID Connect, and SAML. The application can also use password-based SSO, linked-based SSO, or SSO can be disabled.

- **Federation** - When you set up SSO to work between multiple identity providers, it's called federation. An SSO implementation based on federation protocols improves security, reliability, end-user experiences, and implementation.

With federated single sign-on, Microsoft Entra authenticates the user to the application by using their Microsoft Entra account. This method is supported for [SAML 2.0](#), WS-Federation, or [OpenID Connect](#) applications. Federated SSO is the richest mode of SSO. Use federated SSO with Microsoft Entra ID when an application supports it, instead of password-based SSO and Active Directory Federation Services (AD FS).

There are some scenarios where the SSO option isn't present for an enterprise application. If the application was registered using [App registrations](#) in the portal, then the single sign-on capability is configured to use OpenID Connect. In this

case, the single sign-on option doesn't appear in the navigation under enterprise applications. OpenID Connect is an authentication protocol built on top of OAuth 2.0, which is an authorization protocol. OpenID Connect uses OAuth 2.0 to handle the authorization part of the process. When a user tries to log in, OpenID Connect verifies their identity based on the authentication performed by an authorization server. Once the user is authenticated, OAuth 2.0 is used to grant the application access to the user's resources without exposing their credentials.

Single sign-on isn't available when an application is hosted in another tenant.

Single sign-on is also not available if your account doesn't have the required permissions (Cloud Application Administrator, Application Administrator, or owner of the service principal). Permissions can also cause a scenario where you can open single sign-on but might not be able to save.

<https://www.youtube-nocookie.com/embed/CjarTgjKcX8> ↗

- **Password** - On-premises applications can use a password-based method for SSO. This choice works when applications are configured for Application Proxy.

With password-based SSO, users sign in to the application with a username and password the first time they access it. After the first sign-on, Microsoft Entra ID provides the username and password to the application. Password-based SSO enables secure application password storage and replay using a web browser extension or mobile app. This option uses the existing sign-in process provided by the application, enables an administrator to manage the passwords, and doesn't require the user to know the password. For more information, see [Add password-based single sign-on to an application](#).

- **Linked** - Linked sign-on can provide a consistent user experience while you migrate applications over a period of time. If you're migrating applications to Microsoft Entra ID, you can use linked-based SSO to quickly publish links to all the applications you intend to migrate. Users can find all the links in the My Apps or Microsoft 365 portals.

After a user has authenticated with a linked application, an account needs to be created before the user is provided single sign-on access. Provisioning this account can either occur automatically, or it can occur manually by an administrator. You can't apply Conditional Access policies or multifactor authentication to a linked application because a linked application doesn't provide single sign-on capabilities through Microsoft Entra ID. When you configure a linked application, you're simply adding a link that appears for launching the application. For more information, see [Add linked single sign-on to an application](#).

- **Disabled** - When SSO is disabled, it isn't available for the application. When single sign-on is disabled, users might need to authenticate twice. First, users authenticate to Microsoft Entra ID, and then they sign in to the application.

Disable SSO when:

- You're not ready to integrate this application with Microsoft Entra single sign-on
- You're testing other aspects of the application
- An on-premises application doesn't require users to authenticate, but you want them to. With SSO disabled, the user needs to authenticate.

If you configured the application for SP-initiated SAML-based SSO and you change the SSO mode to disabled, it doesn't stop users from signing in to the application outside the MyApps portal. To stop users from signing in from outside My apps portal, you need to disable the ability for users to sign in.

Plan SSO deployment

Web applications are hosted by various companies and made available as a service. Some popular examples of web applications include Microsoft 365, GitHub, and Salesforce. There are thousands of others. People access web applications using a web browser on their computer. Single sign-on makes it possible for people to navigate between the various web applications without having to sign in multiple times. For more information, see [Plan a single sign-on deployment](#).

How you implement SSO depends on where the application is hosted. Hosting matters because of the way network traffic is routed to access the application. Users don't need to use the Internet to access on-premises applications (hosted on a local network). If the application is hosted in the cloud, users need the Internet to use it. Cloud hosted applications are also called Software as a Service (SaaS) applications.

For cloud applications, federation protocols are used. You can also use single sign-on for on-premises applications. You can use Application Proxy to configure access for your on-premises application. For more information, see [Remote access to on-premises applications through Microsoft Entra application proxy](#).

My Apps

If you're a user of an application, you likely don't care much about SSO details. You just want to use the applications that make you productive without having to type your

password so much. You can find and manage your applications at the My Apps portal. For more information, see [Sign in and start apps from the My Apps portal](#).

Next steps

- [Plan for single sign-on deployment](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Plan a single sign-on deployment

Article • 04/30/2025

This article provides information that you can use to plan your single sign-on (SSO) deployment in Microsoft Entra ID. When you plan your SSO deployment with your applications in Microsoft Entra ID, you need to consider the following questions:

- What are the administrative roles required for managing the application?
- Does the Security Assertion Markup Language (SAML) application certificate need to be renewed?
- Who needs to be notified of changes related to the implementation of SSO?
- What licenses are needed to ensure effective management of the application?
- Are shared and guest user accounts used to access the application?
- Do I understand the options for SSO deployment?

Administrative roles

Always use the role with the fewest permissions available to accomplish the required task within Microsoft Entra ID. Review the different roles that are available and choose the right one to solve your needs for each persona for the application. Some roles might need to be applied temporarily and removed after the deployment is completed.

[] Expand table

Persona	Roles	Microsoft Entra role (if necessary)
Help desk admin	Tier 1 support view the sign-in logs to resolve issues.	None
Identity admin	Configure and debug when issues involve Microsoft Entra ID	Cloud Application Administrator
Application admin	User attestation in application, configuration on users with permissions	None
Infrastructure admins	Certificate rollover owner	Cloud Application Administrator
Business owner/stakeholder	User attestation in application, configuration on users with permissions	None

To learn more about Microsoft Entra administrative roles, see [Microsoft Entra built-in roles](#).

Certificates

When you enable federation on SAML application, Microsoft Entra ID creates a certificate that is by default valid for three years. You can customize the expiration date for that certificate if needed. Ensure that you have processes in place to renew certificates before their expiration.

You change that certificate duration in the Microsoft Entra admin center. Make sure to document the expiration and know how to manage your certificate renewal. It's important to identify the right roles and email distribution lists involved with managing the lifecycle of the signing certificate. The following roles are recommended:

- Owner for updating user properties in the application
- Owner On-Call for application troubleshooting support
- Closely monitored email distribution list for certificate-related change notifications

Set up a process for how to handle a certificate change between Microsoft Entra ID and your application. By having this process in place, you can help prevent or minimize an outage due to a certificate expiring or a forced certificate rollover. For more information, see [Manage certificates for federated single sign-on in Microsoft Entra ID](#).

Communications

Communication is critical to the success of any new service. Proactively communicate to your users about the upcoming experience change. Communicate when change is to take place, and how to gain support if they experience issues. Review the options for how users are to access their SSO-enabled applications, and craft your communications to match your selection.

Implement your communication plan. Make sure you're letting your users know that a change is coming, when it arrives, and what to do now. Also, make sure that you provide information about how to seek assistance.

Licensing

Ensure the application is covered by the following licensing requirements:

- **Microsoft Entra ID licensing** - SSO for preintegrated enterprise applications is free. However, the number of objects in your directory and the features you wish to deploy might require more licenses. For a full list of license requirements, see [Microsoft Entra pricing](#).
- **Application licensing** - You need the appropriate licenses for your applications to meet your business needs. Work with the application owner to determine whether the users

assigned to the application have the appropriate licenses for their roles within the application. If Microsoft Entra ID manages the automatic provisioning based on roles, the roles assigned in Microsoft Entra ID must align with the number of licenses owned within the application. Improper number of licenses owned in the application might lead to errors during the provisioning or updating of a user account.

Shared accounts

From the sign-in perspective, applications with shared accounts aren't different from enterprise applications that use password SSO for individual users. However, there are more steps required when planning and configuring an application meant to use shared accounts.

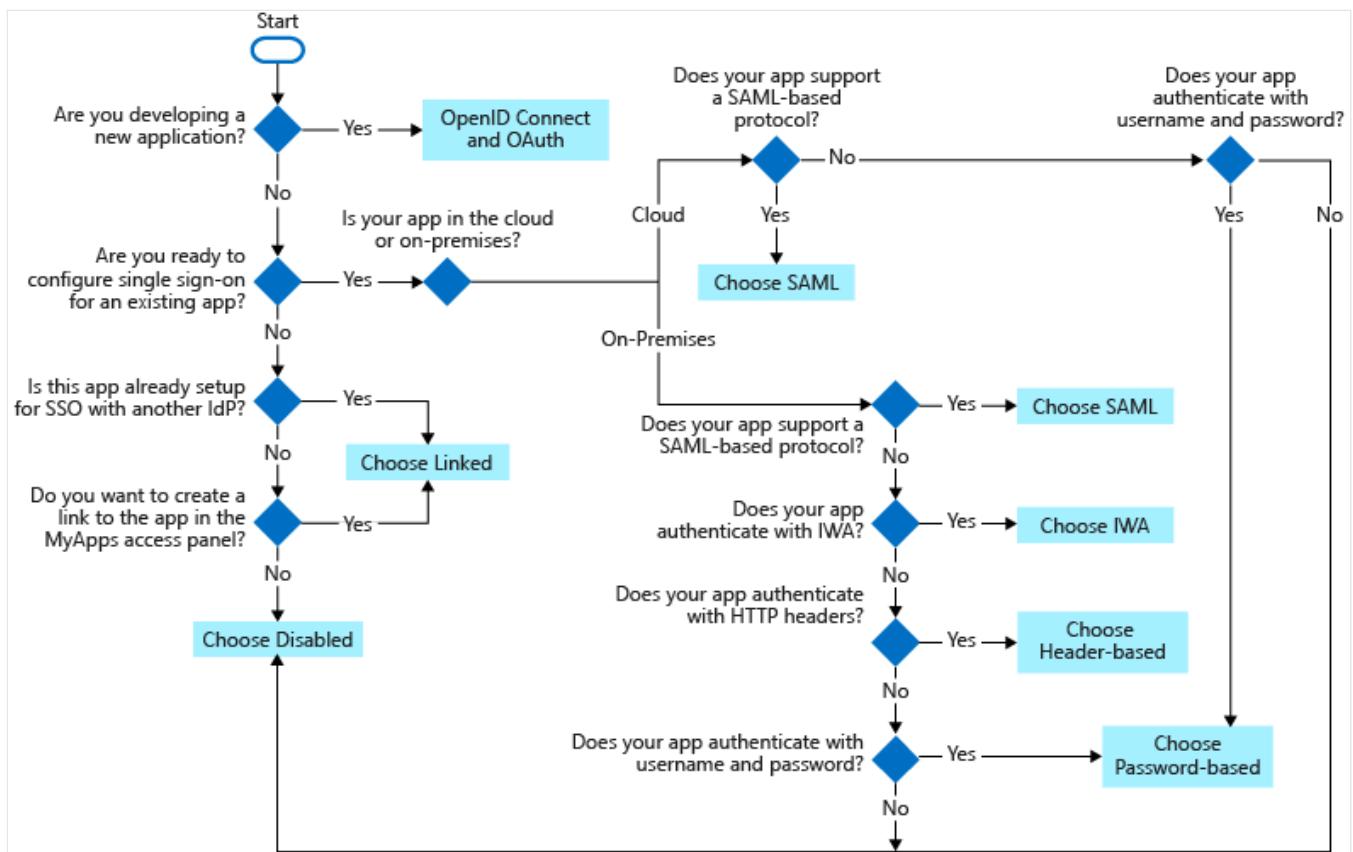
- Work with users to document the following information:
 - The set of users in the organization who are to use the application.
 - The existing set of credentials in the application associated with the set of users.
- For each combination of user set and credentials, create a security group in the cloud or on-premises based on your requirements.
- Reset the shared credentials. After the application is deployed in Microsoft Entra ID, individuals don't need the password of the shared account. Microsoft Entra ID stores the password and you should consider setting it to be long and complex.
- Configure automatic rollover of the password if the application supports it. That way, not even the administrator who did the initial setup knows the password of the shared account.

Single sign-on options

There are several ways you can configure an application for SSO. Choosing an SSO method depends on how the application is configured for authentication.

- Cloud applications can use OpenID Connect, OAuth, SAML, password-based, or linked for SSO. Single sign-on can also be disabled.
- On-premises applications can use password-based, Integrated Windows Authentication, header-based, or linked for SSO. The on-premises choices work when applications are configured for [Application Proxy](#).

This flowchart can help you decide which SSO method is best for your situation.



The following SSO protocols are available to use:

- **OpenID Connect and OAuth** - Choose OpenID Connect and OAuth 2.0 if the application you're connecting to supports it. For more information, see [OAuth 2.0 and OpenID Connect protocols on the Microsoft identity platform](#). For steps to implement OpenID Connect SSO, see [Set up OIDC-based single sign-on for an application in Microsoft Entra ID](#).
 - **SAML** - Choose SAML whenever possible for existing applications that don't use OpenID Connect or OAuth. For more information, see [single sign-on SAML protocol](#).
 - **Password-based** - Choose password-based when the application has an HTML sign-in page. Password-based SSO is also known as password vaulting. Password-based SSO enables you to manage user access and passwords to web applications that don't support identity federation. It's also useful where several users need to share a single account, such as to your organization's social media app accounts.

Password-based SSO supports applications that require multiple sign-in fields for applications that require more than just username and password fields to sign in. You can customize the labels of the username and password fields your users see on My Apps when they enter their credentials. For steps to implement password-based SSO, see [Password-based single sign-on](#).

- **Linked** - Choose linked when the application is configured for SSO in another identity provider service. The linked option lets you configure the target location when a user

selects the application in your organization's end user portals. You can add a link to a custom web application that currently uses federation, such as Active Directory Federation Services (ADFS).

You can also add links to specific web pages that you want to appear on your user's access panels and to an app that doesn't require authentication. The Linked option doesn't provide sign-on functionality through Microsoft Entra credentials. For steps to implement linked SSO, see [Linked single sign-on](#).

- **Disabled** - Choose disabled SSO when the application isn't ready to be configured for SSO.
- **Integrated Windows Authentication (IWA)** - Choose IWA single sign-on for applications that use IWA, or for claims-aware applications. For more information, see [Kerberos Constrained Delegation for single sign-on to your applications with Application Proxy](#).
- **Header-based** - Choose header-based single sign-on when the application uses headers for authentication. For more information, see [Header-based SSO](#).

Next steps

- [Enable single sign-on for applications by using Microsoft Entra ID](#).

Migrate applications away from secret-based authentication

Article • 04/29/2025

Applications that use client secrets might store them in configuration files, hardcode them in scripts, or risk their exposure in other ways. Secret management complexities make secrets susceptible to leaks and attractive to attackers. Client secrets, when exposed, provide attackers with legitimate credentials to blend their activities with legitimate operations, making it easier to bypass security controls. If an attacker compromises an application's client secret, they can escalate their privileges within the system, leading to broader access and control, depending on the permissions of the application. Replacing a compromised certificate can be incredibly time-consuming and disruptive. For these reasons, Microsoft recommends that all of our customers move away from password or certificate-based authentication to token-based authentication.

In this article, we highlight resources and best practices to help you migrate your applications away from secret-based authentication to more secure and user-friendly authentication methods.

Why migrate applications away from secret-based authentication?

Migrating applications away from secret-based authentication offers several benefits:

- **Improved security:** Secret-based authentication is susceptible to leaks and attacks. Migrating to more secure authentication methods, such as managed identities, improves security.
- **Reduced complexity:** Managing secrets can be complex and error-prone. Migrating to more secure authentication methods reduces complexity and improves security.
- **Scalability:** Migrating to more secure authentication methods helps you scale your applications securely.
- **Compliance:** Migrating to more secure authentication methods helps you meet compliance requirements and security best practices.

Best practices for migrating applications away from secret-based authentication

To migrate applications away from secret-based authentication, consider the following best practices:

Use managed identities for Azure resources

Managed identities are a secure way to authenticate applications to cloud services without the need to manage credentials or to have credentials in your code. Azure services use this identity to authenticate to services that support Microsoft Entra authentication. To learn more, see [Assign a managed identity access to an application role](#).

For applications that can't be migrated in the short term, rotate the secret and ensure they use secure practices such as using Azure Key Vault. Azure Key Vault helps you safeguard cryptographic keys and secrets used by cloud applications and services. Keys, secrets, and certificates are protected without you having to write the code yourself, and you can easily use them from your applications. To learn more, see [Azure Key Vault](#).

Deploy Conditional Access policies for workload identities

Conditional Access for workload identities enables you to block service principals from outside of known public IP ranges, based on risk detected by Microsoft Entra Protection or in combination with authentication contexts. To learn more, see [Conditional Access for workload identities](#).

Important

Workload Identities Premium licenses are required to create or modify Conditional Access policies scoped to service principals. In directories without appropriate licenses, existing Conditional Access policies for workload identities continue to function, but can't be modified. For more information, see [Microsoft Entra Workload ID](#).

Implement secret scanning

Secret scanning for your repository checks for any secrets that might already exist in your source code across history and push protection prevents any new secrets from being exposed in source code. To learn more, see [Secret scanning](#).

Deploy application authentication policies to enforce secure authentication practices

Application management policies allow IT admins to enforce best practices for how apps in their organizations should be configured. For example, an admin might configure a policy to block the use or limit the lifetime of password secrets. To learn more, see [Tutorial: Enforce secret and certificate standards using application management policies](#) and [Microsoft Entra application management policies API overview](#).

 **Important**

Premium licenses are required to implement application authentication policy management, for more information, see [Microsoft Entra licensing](#).

Use federated identity for service accounts

Identity federation allows you to access Microsoft Entra protected resources without needing to manage secrets (for supported scenarios) by creating a trust relationship between an external identity provider (IdP) and an app in Microsoft Entra ID by configuring a federated identity credential. To learn more, see [Overview of federated identity credentials in Microsoft Entra ID](#).

Create a least-privileged custom role to rotate application credentials

Microsoft Entra roles allow you to grant granular permissions to your admins, abiding by the principle of least privilege. A custom role can be created to rotate application credentials, ensuring that only the necessary permissions are granted to complete the task. To learn more, see [Create a custom role in Microsoft Entra ID](#).

Ensure you have a process to triage and monitor applications

This process should include regular security assessments, vulnerability scanning, and incident response procedures. Awareness of the security posture of your applications is essential to maintaining a secure environment.

Related content

- [Develop using Zero Trust principles](#).
- [Zero Trust identity and access management development best practices](#)

Manage access to an application

Article • 09/24/2024

Integrating an app into your organization's identity system brings challenges in access management, usage evaluation, and reporting. IT Administrators or help desk staff usually need to oversee app access. Access assignment can fall to a general or divisional IT team, but ideally, business decision makers should be involved, giving approval before IT completes the process.

Other organizations invest in integration with an existing automated identity and access management system, like Role-Based Access Control (RBAC) or Attribute-Based Access Control (ABAC). Both the integration and rule development tend to be specialized and expensive. Monitoring or reporting on either management approach has its own separate, costly, and complex investment.

How does Microsoft Entra ID help?

Microsoft Entra ID supports extensive access management for configured applications, enabling organizations to easily achieve the right access policies ranging from automatic, attribute-based assignment (ABAC or RBAC scenarios) through delegation and including administrator management. With Microsoft Entra ID, you can easily achieve complex policies, combining multiple management models for a single application and can even reuse management rules across applications with the same audiences.

With Microsoft Entra ID, usage and assignment reporting is fully integrated, enabling administrators to easily report on assignment state, assignment errors, and even usage.

Assigning users and groups to an app

Microsoft Entra application assignment focuses on two primary assignment modes:

- **Individual assignment** An IT admin with directory Cloud Application Administrator permissions can select individual user accounts and grant them access to the application.
- **Group-based assignment (requires Microsoft Entra ID P1 or P2)** An IT admin with directory Cloud Application permissions can assign a group to the application. Specific users' access is determined by whether they're members of the group at the time they try to access the application. In other words, an administrator can effectively create an assignment rule stating "any current member of the assigned

group has access to the application." With this assignment option, administrators can benefit from any of Microsoft Entra group management options, including attribute-based [dynamic membership groups](#), external system groups (for example, on-premises Active Directory or Workday), or Administrator-managed or self-service-managed groups. A single group can be easily assigned to multiple apps, making sure that applications with assignment affinity can share assignment rules, reducing the overall management complexity.

 **Note**

[**Nested group**](#) memberships aren't supported for group-based assignment to applications at this time.

With these two assignment modes, administrators can achieve any desirable assignment management approach.

Requiring user assignment for an app

With certain types of applications, you have the option of requiring users to be assigned to the application. By doing so, you prevent everyone from signing in except those users you explicitly assign to the application. The following types of applications support this option:

- Applications configured for federated single sign-on (SSO) with SAML-based authentication
- Application Proxy applications that use Microsoft Entra Pre-Authentication
- Applications, which are built on the Microsoft Entra application platform that use OAuth 2.0 / OpenID Connect Authentication after a user or admin consents to that application. Certain enterprise applications offer more control over who is allowed to sign in.

When user assignment is required, only those users you assign to the application (either through direct user assignment or based on group membership) are able to sign in.

They can access the app on the My Apps portal or by using a direct link.

When user assignment isn't required, unassigned users don't see the app on their My Apps, but they can still sign in to the application itself (also known as SP-initiated sign-on) or they can use the **User Access URL** in the application's **Properties** page (also known as IDP-initiated sign on). For more information on requiring user assignment configurations, See [Configure an application](#)

This setting doesn't affect whether or not an application appears on My Apps. Applications appear on users' My Apps portal once you assign a user or group to the application.

Note

When an application requires assignment, user consent for that application isn't allowed. This is true even if users consent for that app would have otherwise been allowed. Be sure to [grant tenant-wide admin consent](#) to apps that require assignment.

For some applications, the option to require user assignment isn't available in the application's properties. In these cases, you can use PowerShell to set the `appRoleAssignmentRequired` property on the service principal.

Determining the user experience for accessing apps

Microsoft Entra ID provides [several customizable ways to deploy applications](#) to end users in your organization:

- Microsoft Entra My Apps
- Microsoft 365 application launcher
- Direct sign-on to federated apps (service-pr)
- Deep links to federated, password-based, or existing apps

You can determine whether users assigned to an enterprise app can see it in My Apps and Microsoft 365 application launcher.

Example: Complex application assignment with Microsoft Entra ID

Consider an application like Salesforce. In many organizations, Salesforce is primarily used by the marketing and sales teams. Often, members of the marketing team have highly privileged access to Salesforce, while members of the sales team get limited access. In many cases, a broad population of information workers gets restricted access to the application. Exceptions to these rules complicate matters. It's often the prerogative of the marketing or sales leadership teams to grant a user access or change their roles independently of these generic rules.

With Microsoft Entra ID, applications like Salesforce can be preconfigured for single sign-on (SSO) and automated provisioning. Once the application is configured, an

Administrator can take the one-time action to create and assign the appropriate groups. In this example, an administrator could execute the following assignments:

- [Dynamic groups](#) can be defined to automatically represent all members of the marketing and sales teams using attributes like department or role:
 - All members of marketing groups would be assigned to the "marketing" role in Salesforce
 - All members of sales team groups would be assigned to the "sales" role in Salesforce. A further refinement could use multiple groups that represent regional sales teams assigned to different Salesforce roles.
- To enable the exception mechanism, a self-service group could be created for each role. For example, the "Salesforce marketing exception" group can be created as a self-service group. The group can be assigned to the Salesforce marketing role and the marketing leadership team can be made owner. Members of the marketing leadership team could add or remove users, set a join policy, or even approve or deny individual users' requests to join. This mechanism is supported through an information worker appropriate experience that doesn't require specialized training for owners or members.

In this case, all assigned users would be automatically provisioned to Salesforce. As they're added to different groups, their role assignment is updated in Salesforce. Users can discover and access Salesforce through My Apps, Office web clients, or by navigating to their organizational Salesforce sign in page. Administrators can easily view usage and assignment status using Microsoft Entra ID reporting.

Administrators can employ [Microsoft Entra Conditional Access](#) to set access policies for specific roles. These policies can include whether access is permitted outside the corporate environment and even multifactor authentication or device requirements to achieve access in various cases.

Access to Microsoft applications

Microsoft Applications (like Exchange, SharePoint, Yammer, and so on) are assigned and managed a bit differently than non-Microsoft SaaS applications or other applications, which you integrate with Microsoft Entra ID for single sign-on.

There are three main ways that a user can get access to a Microsoft-published application.

- For applications in the Microsoft 365 or other paid suites, users are granted access through **license assignment** either directly to their user account, or through a

group using our group-based license assignment capability.

- For applications that Microsoft or a non-Microsoft organization publishes freely for anyone to use, users can be granted access through [user consent](#). The users sign in to the application with their Microsoft Entra work or school account and allow it to have access to some limited set of data on their account.
- For applications that Microsoft or a non-Microsoft organization publishes freely for anyone to use, users can also be granted access through [administrator consent](#). This means that an administrator has determined the application can be used by everyone in the organization, so they sign in to the application with a [Privileged Role Administrator](#) role and grant access to everyone in the organization.

Some applications combine these methods. For example, certain Microsoft applications are part of a Microsoft 365 subscription, but still require consent.

Users can access Microsoft 365 applications through their Office 365 portals. You can also show or hide Microsoft 365 applications in the My Apps with the [Office 365 visibility toggle](#) in your directory's [User settings](#).

As with enterprise apps, you can [assign users](#) to certain Microsoft applications via the Microsoft Entra admin center or, using PowerShell.

Preventing application access through local accounts

Microsoft Entra ID enables your organization to set up single sign-on to protect how users authenticate to applications with conditional access, multi-factor authentication, etc. Some applications historically have their own local user store and allow users to sign into the application using local credentials or an application-specific backup authentication method, instead of using single sign-on. These application capabilities could be misused and allow users to retain access to applications even after they are no longer assigned to the application in Microsoft Entra ID or can no longer sign into Microsoft Entra ID, and could allow attackers to attempt to compromise the application without appearing in the Microsoft Entra ID logs. To ensure that sign ins to these applications are protected by Microsoft Entra ID:

- Identify which applications connected to your directory for single sign-on allow end users to bypass single sign-on with a local application credential or a backup authentication method. You will need to review the documentation provided by the application provider to understand if this is possible, and what settings are available. Then, in those applications, disable the settings that allow end users to

bypass SSO. Test the end user experience has been secured by opening a browser in InPrivate, connecting to the applications' sign in page, providing the identity of a user in your tenant, and verify that there is no option to sign in other than via Microsoft Entra.

- If your application provides an API to manage user passwords, remove the local passwords or set a unique password for each user using the APIs. This will prevent end users from signing into the application with local credentials.
- If your application provides an API to manage users, configure Microsoft Entra user provisioning to that application to disable or delete user accounts when users are no longer in scope of the application or the tenant.

Next steps

- [Protecting apps with Conditional Access](#)
- [Self-service group management/SSAA](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Advanced certificate signing options in a SAML token

Article • 01/29/2025

Today Microsoft Entra ID supports thousands of preintegrated applications in the Microsoft Entra App Gallery. Over 500 of the applications support single sign-on by using the [Security Assertion Markup Language \(SAML\)](#) 2.0 protocol, such as the [NetSuite](#) application. When a customer authenticates to an application through Microsoft Entra ID by using SAML, Microsoft Entra ID sends a token to the application (via an HTTP POST). The application then validates and uses the token to sign in the customer instead of prompting for a username and password. These SAML tokens are signed with the unique certificate generated in Microsoft Entra ID and by specific standard algorithms.

Microsoft Entra ID uses some of the default settings for the gallery applications. The default values are set up based on the application's requirements.

In Microsoft Entra ID, you can set up certificate signing options and the certificate signing algorithm.

Certificate signing options

Microsoft Entra ID supports three certificate signing options:

- **Sign SAML assertion.** This default option is set for most of the gallery applications. If you select this option, Microsoft Entra ID as an Identity Provider (IdP) signs the SAML assertion and certificate with the [X.509](#) certificate of the application.
- **Sign SAML response.** If you select this option, Microsoft Entra ID as an IdP signs the SAML response with the X.509 certificate of the application.
- **Sign SAML response and assertion.** If you select this option, Microsoft Entra ID as an IdP signs the entire SAML token with the X.509 certificate of the application.

Certificate signing algorithms

Microsoft Entra ID supports two signing algorithms, or secure hash algorithms (SHAs), to sign the SAML response:

- **SHA-256.** Microsoft Entra ID uses this default algorithm to sign the SAML response. It's the newest algorithm and is more secure than SHA-1. Most of the applications support the SHA-256 algorithm. If an application supports only SHA-1 as the signing algorithm, you can change it. Otherwise, we recommend that you use the SHA-256 algorithm for signing the SAML response.
- **SHA-1.** This algorithm is older, and is less secure than SHA-256. If an application supports only this signing algorithm, you can select this option in the **Signing Algorithm** drop-down list. Microsoft Entra ID then signs the SAML response with the SHA-1 algorithm.

Prerequisites

To change an application's SAML certificate signing options and the certificate signing algorithm, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free](#).
- One of the following roles: Cloud Application Administrator, Application Administrator.

Change certificate signing options and signing algorithm

To change an application's SAML certificate signing options and the certificate signing algorithm:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. Enter the name of the existing application in the search box, and then select the application from the search results.

Next, change the certificate signing options in the SAML token for that application:

1. In the left pane of the application overview page, select **Single sign-on**.
2. If the **Set up Single Sign-On with SAML** page appears, go to step 5.
3. If the **Set up Single Sign-On with SAML** page doesn't appear, select **Change single sign-on modes**.
4. In the **Select a single sign-on method** page, select **SAML**. If **SAML** isn't available, the application doesn't support SAML, and you may ignore the rest of this

procedure and article.

5. In the **Set up Single Sign-On with SAML** page, find the **SAML Signing Certificate** heading and select the **Edit** icon (a pencil). The **SAML Signing Certificate** page appears.
6. In the **Signing Option** drop-down list, choose **Sign SAML response**, **Sign SAML assertion**, or **Sign SAML response and assertion**. Descriptions of these options appear earlier in this article in the [Certificate signing options](#).
7. In the **Signing Algorithm** drop-down list, choose **SHA-1** or **SHA-256**. Descriptions of these options appear earlier in this article in the [Certificate signing algorithms](#) section.
8. If you're satisfied with your choices, select **Save** to apply the new SAML signing certificate settings. Otherwise, select the **X** to discard the changes.

Next steps

- [Configure single sign-on to applications that are not in the Microsoft Entra App Gallery](#)
- [Troubleshoot SAML-based single sign-on](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Restrict access to a tenant

Article • 11/29/2024

Large organizations that emphasize security want to move to cloud services like Microsoft 365, but need to know that their users only can access approved resources. Traditionally, companies restrict domain names or IP addresses when they want to manage access. This approach fails in a world where software as a service (or SaaS) apps are hosted in a public cloud, running on shared domain names like `outlook.office.com` and `login.microsoftonline.com`. Blocking these addresses would keep users from accessing Outlook on the web entirely, instead of merely restricting them to approved identities and resources.

The Microsoft Entra solution to this challenge is a feature called tenant restrictions. With tenant restrictions, organizations can control access to SaaS cloud applications, based on the Microsoft Entra tenant the applications use for [single sign-on](#). For example, you might want to allow access to your organization's Microsoft 365 applications, while preventing access to other organizations' instances of these same applications.

With tenant restrictions, organizations can specify the list of tenants that users on their network are permitted to access. Microsoft Entra ID then only grants access to these permitted tenants - all other tenants are blocked, even ones that your users might be guests in.

This article focuses on tenant restrictions for Microsoft 365, but the feature protects all apps that send the user to Microsoft Entra ID for single sign-on. If you use SaaS apps with a different Microsoft Entra tenant from the tenant used by your Microsoft 365, make sure that all required tenants are permitted. (For example, in B2B collaboration scenarios). For more information about SaaS cloud apps, see the [Active Directory Marketplace](#).

The tenant restrictions feature also supports [blocking the use of all Microsoft consumer applications](#) (MSA apps) such as OneDrive, Hotmail, and Xbox.com. This functionality uses a separate header to the `login.live.com` endpoint, and is detailed at the end of this article.

How it works

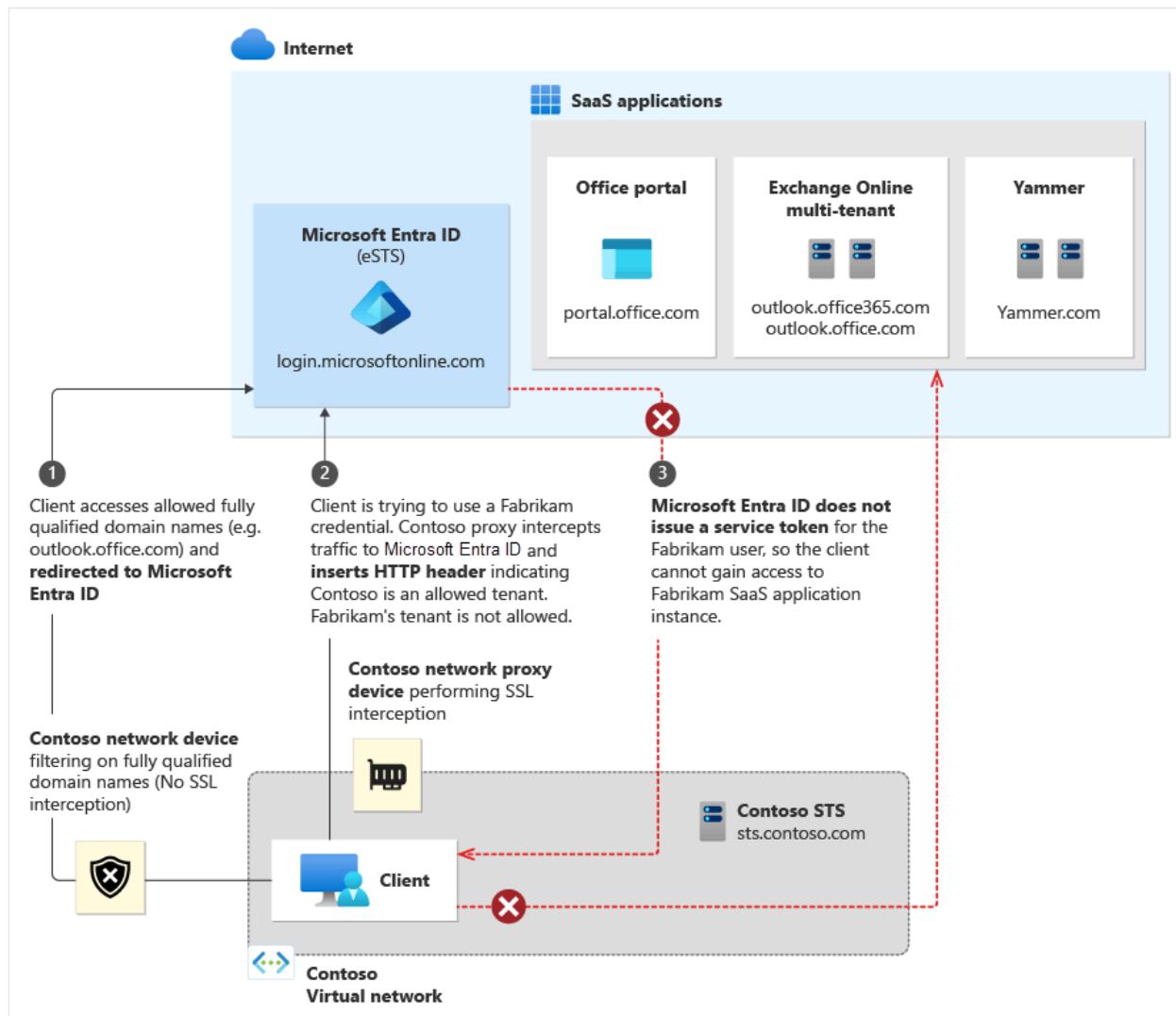
The overall solution comprises the following components:

1. **Microsoft Entra ID:** If the `Restrict-Access-To-Tenants: <permitted tenant list>` header is present, Microsoft Entra-only issues security tokens for the permitted

tenants.

2. **On-premises proxy server infrastructure:** This infrastructure is a proxy device capable of Transport Layer Security (TLS) inspection. You must configure the proxy to insert the header containing the list of permitted tenants into traffic destined for Microsoft Entra ID.
3. **Client software:** To support tenant restrictions, client software must request tokens directly from Microsoft Entra ID, so that the proxy infrastructure can intercept traffic. Browser-based Microsoft 365 applications currently support tenant restrictions, as do Office clients that use modern authentication (like OAuth 2.0).
4. **Modern Authentication:** Cloud services must use modern authentication to use tenant restrictions and block access to all nonpermitted tenants. You must configure Microsoft 365 cloud services to use modern authentication protocols by default. For the latest information on Microsoft 365 support for modern authentication, read [Updated Office 365 modern authentication](#).

The following diagram illustrates the high-level traffic flow. Tenant restrictions require TLS inspection only on traffic to Microsoft Entra ID, not to the Microsoft 365 cloud services. This distinction is important, because the traffic volume for authentication to Microsoft Entra ID is typically much lower than traffic volume to SaaS applications like Exchange Online and SharePoint Online.



Set up tenant restrictions

There are two steps to get started with tenant restrictions. First, make sure that your clients can connect to the right addresses. Second, configure your proxy infrastructure.

URLs and IP addresses

To use tenant restrictions, your clients must be able to connect to the following Microsoft Entra URLs to authenticate:

- `login.microsoftonline.com`
- `login.microsoft.com`
- `login.windows.net`

Additionally, to access Office 365, your clients must also be able to connect to the fully qualified domain names (FQDNs), URLs, and IP addresses defined in [Office 365 URLs and IP address ranges](#).

Proxy configuration and requirements

The following configuration is required to enable tenant restrictions through your proxy infrastructure. This guidance is generic, so you should refer to your proxy vendor's documentation for specific implementation steps.

Prerequisites

- The proxy must be able to perform TLS interception, HTTP header insertion, and filter destinations using FQDNs/URLs.
- Clients must trust the certificate chain presented by the proxy for TLS communications. For example, if certificates from an internal public key infrastructure (PKI) are used, the internal issuing root certificate authority certificate must be trusted.
- Microsoft Entra ID P1 or P2 licenses are required for use of tenant restrictions.

Configuration

For each outgoing request to `login.microsoftonline.com`, `login.microsoft.com`, and `login.windows.net`, insert two HTTP headers: *Restrict-Access-To-Tenants* and *Restrict-Access-Context*.

ⓘ Note

Do not include subdomains under `*.login.microsoftonline.com` in your proxy configuration. Doing so will include `device.login.microsoftonline.com` and will interfere with Client Certificate authentication, which is used in Device Registration and Device-based Conditional Access scenarios. Configure your proxy server to exclude `device.login.microsoftonline.com` and `enterpriseregistration.windows.net` from TLS break-and-inspect and header injection.

The headers should include the following elements:

- For *Restrict-Access-To-Tenants*, use a value of <permitted tenant list>, which is a comma-separated list of tenants you want to allow users to access. Any domain that is registered with a tenant can be used to identify the tenant in this list, and the directory ID itself. For an example of all three ways of describing a tenant, the name/value pair to allow Contoso, Fabrikam, and Microsoft looks like: `Restrict-`

`Access-To-Tenants: contoso.com,fabrikam.onmicrosoft.com,aaaabbbb-0000-cccc-1111-dddd2222eeee`

- For *Restrict-Access-Context*, use a value of a single directory ID, declaring which tenant is setting the tenant restrictions. For example, to declare Contoso as the tenant that set the tenant restrictions policy, the name/value pair looks like:

`Restrict-Access-Context: bbbbcccc-1111-dddd-2222-eeee3333ffff`. You *must* use your own directory ID here to get logs for these authentications. If you use any directory ID other than your own, the sign-in logs *appear in someone else's tenant, with all personal information removed. For more information, see [Admin experience](#).

To find your directory ID:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Global Reader**.
2. Browse to **Identity > Overview > Overview**.
3. Copy the **Tenant ID** value.

To validate that a directory ID or domain name refer to the same tenant, use that ID or domain in place of <tenant> in this URL:

`https://login.microsoftonline.com/<tenant>/v2.0/.well-known/openid-configuration`. If the results with the domain and the ID are the same, they refer to the same tenant.

To prevent users from inserting their own HTTP header with nonapproved tenants, the proxy needs to replace the *Restrict-Access-To-Tenants* header if it's already present in the incoming request.

Clients must be forced to use the proxy for all requests to `login.microsoftonline.com`, `login.microsoft.com`, and `login.windows.net`. For example, if PAC files are used to direct clients to use the proxy, end users shouldn't be able to edit or disable the PAC files.

The user experience

This section describes the experience for both end users and admins.

End-user experience

An example user is on the Contoso network, but is trying to access the Fabrikam instance of a shared SaaS application like Outlook online. If Fabrikam is a nonpermitted tenant for the Contoso instance, the user sees an access denial message. The denial message says you're trying to access a resource that belongs to an organization unapproved by your IT department.



user@example.com

External access blocked by policy

Your network administrator has restricted what organizations can be accessed. Contact your IT department to unblock access.

[Read more about tenant restrictions](#)

[Sign out to protect your account](#)

Admin experience

While configuration of tenant restrictions is done on the corporate proxy infrastructure, admins can access the tenant restrictions reports in the Microsoft Entra admin center directly. To view the reports:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Global Reader](#).
2. Browse to **Identity > Overview > Tenant restrictions**.

The admin for the tenant specified as the Restricted-Access-Context tenant can use this report to see sign-ins blocked because of the tenant restrictions policy, including the identity used and the target directory ID. Sign-ins are included if the tenant setting the restriction is either the user tenant or resource tenant for the sign-in.

The report might contain limited information, such as target directory ID, when a user who is in a tenant other than the Restricted-Access-Context tenant signs in. In this case, user identifiable information, such as name and user principal name, is masked to protect user data in other tenants (For example, "`{PII Removed}@domain.com`" or `00000000-0000-0000-000000000000` in place of usernames and object IDs as appropriate).

Like other reports in the Microsoft Entra admin center, you can use filters to specify the scope of your report. You can filter on a specific time interval, user, application, client, or

status. If you select the **Columns** button, you can choose to display data with any combination of the following fields:

- **User** - this field can have personal data removed, where its value is set to `00000000-0000-0000-0000-000000000000`.
- **Application**
- **Status**
- **Date**
- **Date (UTC)** - where UTC is Coordinated Universal Time
- **IP Address**
- **Client**
- **Username** - this field can have personal data removed, where its value is set to `{PII Removed}@domain.com`
- **Location**
- **Target tenant ID**

Microsoft 365 support

Microsoft 365 applications must meet two criteria to fully support tenant restrictions:

1. The client used supports modern authentication.
2. Modern authentication is enabled as the default authentication protocol for the cloud service.

For the latest information on which Office clients currently support modern authentication, see [Updated Office 365 modern authentication](#). That page also includes links to instructions for enabling modern authentication on specific Exchange Online and Skype for Business Online tenants. SharePoint Online already enables Modern authentication by default. Teams only supports modern auth, and doesn't support legacy auth, so this bypass concern doesn't apply to Teams.

Microsoft 365 browser-based applications (such as the Office Portal, Yammer, SharePoint sites, and Outlook on the Web.) currently support tenant restrictions. Thick clients (Outlook, Skype for Business, Word, Excel, PowerPoint, and more) can enforce tenant restrictions only when using modern authentication.

Outlook and Skype for Business clients that support modern authentication might still be able to use legacy protocols against tenants where modern authentication isn't enabled, effectively bypassing tenant restrictions. Tenant restrictions might block applications that use legacy protocols if they contact `login.microsoftonline.com`, `login.microsoft.com`, or `login.windows.net` during authentication.

For Outlook on Windows, customers might choose to implement restrictions preventing end users from adding nonapproved mail accounts to their profiles. For example, see the [Prevent adding nondefault Exchange accounts](#) group policy setting.

Azure RMS and Office Message Encryption incompatibility

The [Azure Rights Management Service \(Azure RMS\)](#) and [Office Message Encryption](#) features aren't compatible with tenant restrictions. These features rely on signing your users into other tenants in order to get decryption keys for the encrypted documents. Because tenant restrictions blocks access to other tenants, encrypted mail and documents sent to your users from untrusted tenants aren't accessible.

Testing

If you want to try out tenant restrictions before implementing it for your whole organization, you have two options: a host-based approach using a tool like Fiddler, or a staged rollout of proxy settings.

Fiddler for a host-based approach

Fiddler is a free web debugging proxy that can be used to capture and modify HTTP/HTTPS traffic, it includes inserting HTTP headers. To configure Fiddler to test tenant restrictions, perform the following steps:

1. [Download and install Fiddler](#).
2. Configure Fiddler to decrypt HTTPS traffic, per [Fiddler's help documentation](#).
3. Configure Fiddler to insert the *Restrict-Access-To-Tenants* and *Restrict-Access-Context* headers using custom rules:
 - a. In the Fiddler Web Debugger tool, select the Rules menu and select **Customize Rules...** to open the CustomRules file.
 - b. Add the following lines within the `OnBeforeRequest` function. Replace <List of tenant identifiers> with a domain registered with your tenant (for example, `contoso.onmicrosoft.com`). Replace <directory ID> with your tenant's Microsoft Entra GUID identifier. You **must** include the correct GUID identifier in order for the logs to appear in your tenant.

```

// Allows access to the listed tenants.
if (
    oSession.HostnameIs("login.microsoftonline.com") ||
    oSession.HostnameIs("login.microsoft.com") ||
    oSession.HostnameIs("login.windows.net")
)
{
    oSession.oRequest["Restrict-Access-To-Tenants"] = "<List of
tenant identifiers>";
    oSession.oRequest["Restrict-Access-Context"] = "<Your directory
ID>";
}

// Blocks access to consumer apps
if (
    oSession.HostnameIs("login.live.com")
)
{
    oSession.oRequest["sec-Restrict-Tenant-Access-Policy"] =
"restrict-msa";
}

```

If you need to allow multiple tenants, use a comma to separate the tenant names. For example:

```

oSession.oRequest["Restrict-Access-To-Tenants"] =
"contoso.onmicrosoft.com,fabrikam.onmicrosoft.com";

```

4. Save and close the CustomRules file.

After you configure Fiddler, you can capture traffic by going to the **File** menu and selecting **Capture Traffic**.

Staged rollout of proxy settings

Depending on the capabilities of your proxy infrastructure, you might be able to stage the rollout of settings to your users. See the following high-level options for consideration:

1. Use PAC files to point test users to a test proxy infrastructure, while normal users continue to use the production proxy infrastructure.
2. Some proxy servers might support different configurations using groups.

For specific details, refer to your proxy server documentation.

Blocking consumer applications

Applications from Microsoft that support both consumer accounts and organizational accounts such as OneDrive can sometimes be hosted on the same URL. This means that users that must access that URL for work purposes also have access to it for personal use. This option might not be permitted under your operating guidelines.

Some organizations attempt to fix this problem by blocking `login.live.com` in order to block personal accounts from authenticating. This fix has several downsides:

1. Blocking `login.live.com` blocks the use of personal accounts in B2B guest scenarios, which can intrude on visitors and collaboration.
2. [Autopilot requires the use of login.live.com](#) in order to deploy. Intune and Autopilot scenarios can fail when `login.live.com` is blocked.
3. Organizational telemetry and Windows updates that rely on the `login.live.com` service for device IDs [cease to work](#).

Configuration for consumer apps

While the `Restrict-Access-To-Tenants` header functions as an allowlist, the Microsoft account (MSA) block works as a deny signal, telling the Microsoft account platform to not allow users to sign in to consumer applications. To send this signal, the `sec-Restrict-Tenant-Access-Policy` header is injected to traffic visiting `login.live.com` using the same corporate proxy or firewall as mentioned in the [proxy configuration and requirements](#) section of this article. The value of the header must be `restrict-msa`.

When the header is present and a consumer app is attempting to sign in a user directly, that sign-in is blocked.

At this time, authentication to consumer applications doesn't appear in the [admin logs](#), as `login.live.com` is hosted separately from Microsoft Entra ID.

What the header does and doesn't block

The `restrict-msa` policy blocks the use of consumer applications, but allows through several other types of traffic and authentication:

1. User-less traffic for devices. This option includes traffic for Autopilot, Windows Update, and organizational telemetry.
2. B2B authentication of consumer accounts. Users with Microsoft accounts that are [invited to collaborate with a tenant](#) authenticate to `login.live.com` in order to access a resource tenant.
 - a. This access is controlled using the `Restrict-Access-To-Tenants` header to allow or deny access to that resource tenant.

3. "Passthrough" authentication, used by many Azure apps and Office.com, where apps use Microsoft Entra ID to sign in consumer users in a consumer context.
 - a. This access is also controlled using the `Restrict-Access-To-Tenants` header to allow or deny access to the special "passthrough" tenant (`f8cdef31-a31e-4b4a-93e4-5f571e91255a`). If this tenant doesn't appear in your `Restrict-Access-To-Tenants` list of allowed domains, Microsoft Entra ID blocks consumer accounts from signing into these apps.

Platforms that don't support TLS break and inspect

Tenant restrictions depends on injection of a list of allowed tenants in the HTTPS header. This dependency requires Transport Layer Security Inspection (TLSI) to break and inspect traffic. For environments where the client's side isn't able to break and inspect the traffic to add headers, tenant restrictions doesn't work.

Take the example of Android 7.0 and onwards. Android changed how it handles trusted certificate authorities (CAs) to provide safer defaults for secure app traffic. For more information, see [Changes to Trusted Certificate Authorities in Android Nougat ↗](#).

Following the recommendation from Google, Microsoft client apps ignore user certificates by default. This policy makes such apps unable to work with tenant restrictions, since the certificates used by the network proxy are installed in the user certificate store, which client apps don't trust.

For such environments that can't break and inspect traffic to add the tenant restrictions parameters onto the header, other features of Microsoft Entra ID can provide protection. The following list provides more information on such Microsoft Entra features.

- [Conditional Access: Only allow use of managed/compliant devices](#)
- [Conditional Access: Manage access for guest/external users](#)
- [B2B Collaboration: Restrict outbound rules by Cross-tenant access for the same tenants listed in the parameter "Restrict-Access-To-Tenants"](#)
- [B2B Collaboration: Restrict invitations to B2B users to the same domains listed in the "Restrict-Access-To-Tenants" parameter](#)
- [Application management: Restrict how users consent to applications](#)
- [Intune: Apply App Policy through Intune to restrict usage of managed apps to only the UPN of the account that enrolled the device](#) - Check the section under, **Allow only configured organization accounts in apps** subheading.

However, some specific scenarios can only be covered using tenant restrictions.

Next steps

- Read about [Updated Office 365 modern authentication ↗](#)
 - Review the [Office 365 URLs and IP address ranges ↗](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Enforce signed SAML authentication requests

Article • 07/19/2024

SAML Request Signature Verification is a functionality that validates the signature of signed authentication requests. An App Admin now can enable and disable the enforcement of signed requests and upload the public keys that should be used to do the validation.

If enabled Microsoft Entra ID validates the requests against the public keys configured. There are some scenarios where the authentication requests can fail:

- Protocol not allowed for signed requests. Only SAML protocol is supported.
- Request not signed, but verification is enabled.
- No verification certificate configured for SAML request signature verification. For more information about the certificate requirements, see [Certificate signing options](#).
- Signature verification failed.
- Key identifier in request is missing and two most recently added certificates don't match with the request signature.
- Request signed but algorithm missing.
- No certificate matching with provided key identifier.
- Signature algorithm not allowed. Only RSA-SHA256 is supported.

ⓘ Note

A `Signature` element in `AuthnRequest` elements is optional. If `Require Verification certificates` is not checked, Microsoft Entra ID does not validate signed authentication requests if a signature is present. Requestor verification is provided for by only responding to registered Assertion Consumer Service URLs.

If `Require Verification certificates` is checked, SAML Request Signature Verification will work for SP-initiated(service provider/relying party initiated) authentication requests only. Only the application configured by the service provider will have the access to the private and public keys for signing the incoming SAML Authentication Requests from the application. The public key should be uploaded to allow the verification of the request, in which case Microsoft Entra ID will have access to only the public key.

Enabling [Require Verification certificates](#) will not allow IDP-initiated authentication requests (like SSO testing feature, MyApps or M365 app launcher) to be validated as the IDP would not possess the same private keys as the registered application.

Prerequisites

To configure SAML request signature verification, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free ↗](#).
- One of the following roles: Cloud Application Administrator, Application Administrator, or owner of the service principal.

💡 Tip

Steps in this article might vary slightly based on the portal you start from.

Configure SAML Request Signature Verification

1. Sign in to the [Microsoft Entra admin center ↗](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. Enter the name of the existing application in the search box, and then select the application from the search results.
4. Navigate to **Single sign-on**.
5. In the **Single sign-on** screen, scroll to the subsection called **Verification certificates** under **SAML Certificates**.

Microsoft Entra admin center

Search resources, services, and docs (G+/)

Home > Enterprise applications | All applications > TestApp

TestApp | SAML-based Sign-on

Enterprise Application

Overview Deployment Plan Diagnose and solve problems

Manage Properties Owners Roles and administrators Users and groups Single sign-on Provisioning Application proxy Self-service Custom security attributes

Security Conditional Access Permissions Token encryption

Activity Sign-in logs Usage & insights Audit logs Provisioning logs Access reviews Troubleshooting + Support

Identifier (Entity ID) **Required**
Reply URL (Assertion Consumer Service URL) https://TestApp-a111a1111111.msappproxy.net/
Sign on URL **Optional**
Relay State (Optional) **Optional**
Logout Url (Optional) **Optional**

Attributes & Claims
⚠ Fill out required fields in Step 1
givenname user.givenname
surname user.surname
emailaddress user.mail
name user.userprincipalname
Unique User Identifier user.userprincipalname

SAML Certificates
Token signing certificate
Status Active
Thumbprint 1A111AA1111AAAA1111A111AA1111AA1A11AAA1
Expiration 12/5/2028, 12:16:57 PM
Notification Email example@microsoft.com
App Federation Metadata Url https://login.microsoftonline.com/2b6...
Certificate (Base64) Download
Certificate (Raw) Download
Federation Metadata XML Download

Verification certificates (optional)
Required No
Active 0
Expired 0

Identifier (Entity ID) **Required**
Reply URL (Assertion Consumer Service URL) https://TestApp-a111a1111111.msappproxy.net/
Sign on URL **Optional**
Relay State (Optional) **Optional**
Logout Url (Optional) **Optional**

Attributes & Claims
⚠ Fill out required fields in Step 1
givenname user.givenname
surname user.surname
emailaddress user.mail
name user.userprincipalname
Unique User Identifier user.userprincipalname

SAML Certificates
Token signing certificate
Status Active
Thumbprint 1A111AA1111AAAA1111A111AA1111AA1A11AAA1
Expiration 12/5/2028, 12:16:57 PM
Notification Email example@microsoft.com
App Federation Metadata Url https://login.microsoftonline.com/2b6...
Certificate (Base64) Download
Certificate (Raw) Download
Federation Metadata XML Download

Verification certificates (optional)
Required No
Active 0
Expired 0

6. Select **Edit**.

7. In the new pane, you're able to enable the verification of signed requests and opt-in for weak algorithm verification in case your application still uses RSA-SHA1 to sign the authentication requests.
8. To enable the verification of signed requests, select **Require verification certificates** and upload a verification public key that matches with the private key used to sign the request.

Verification certificates

X

Requiring verification certificates will impact certain admin and end-user experiences, such as the Single sign-on testing feature, and the My Apps and M365 app launcher experiences.
[Learn more](#)

Verification certificates are used to verify requests coming from this application to Microsoft Entra ID.
[Learn more](#)

Require verification certificates

Allow requests signed with RSA-SHA1

 Upload certificate

Thumbprint

Key Id

Start date

Expiration date

You do not have any verification certificates.

[Save](#)

[Discard](#)

9. Once you have your verification certificate uploaded, select **Save**.

10. When the verification of signed requests is enabled, the test experience is disabled as the service provider has to sign the request.

5

Test single sign-on with Test Verification Certs

Test to see if single sign-on is working. Users will need to be added to Users and groups before they can sign in.

 Testing single-sign on is currently disabled. To enable single sign-on testing, request verification must be disabled in the "SAML certificates" settings. [Learn more](#)

[Test](#)

11. If you want to see the current configuration of an enterprise application, you can navigate to the **Single Sign-on** screen and see the summary of your configuration under **SAML Certificates**. There you're able to see if the verification of signed requests is enabled and the count of Active and Expired verification certificates.

Verification certificates (optional)

 Edit

Enabled	Yes
Active	1
Expired	0

Next steps

- Find out [How Microsoft Entra ID uses the SAML protocol](#)

- Learn the format, security characteristics, and contents of [SAML tokens in Microsoft Entra ID](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Configure Microsoft Entra SAML token encryption

Article • 03/06/2025

ⓘ Note

Token encryption is a Microsoft Entra ID P1 or P2 feature. To learn more about Microsoft Entra editions, features, and pricing, see [Microsoft Entra pricing](#).

SAML token encryption enables the use of encrypted SAML assertions with an application that supports it. When configured for an application, Microsoft Entra ID encrypts the SAML assertions it emits for that application. It encrypts the SAML assertions using the public key obtained from a certificate stored in Microsoft Entra ID. The application must use the matching private key to decrypt the token before it can be used as evidence of authentication for the signed in user.

Encrypting the SAML assertions between Microsoft Entra ID and the application provides more assurance that the content of the token can't be intercepted, and personal or corporate data compromised.

Even without token encryption, Microsoft Entra SAML tokens are never passed on the network in the clear. Microsoft Entra ID requires token request/response exchanges to take place over encrypted HTTPS/TLS channels so that communications between the IDP, browser, and application take place over encrypted links. Consider the value of token encryption for your situation compared with the overhead of managing more certificates.

To configure token encryption, you need to upload an X.509 certificate file that contains the public key to the Microsoft Entra application object that represents the application.

To obtain the X.509 certificate, you can download it from the application itself. You can also get it from the application vendor in cases where the application vendor provides encryption keys. If the application expects you to provide a private key, you can create it using cryptography tools. The private key portion is uploaded to the application's key store and the matching public key certificate uploaded to Microsoft Entra ID.

Microsoft Entra ID uses AES-256 to encrypt the SAML assertion data.

Prerequisites

To configure SAML token encryption, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free](#).
- One of the following roles:
 - Cloud Application Administrator
 - Application Administrator
 - owner of the service principal

Configure enterprise application SAML token encryption

This section describes how to configure an enterprise application's SAML token encryption. These applications are set up from the **Enterprise applications** pane in the Microsoft Entra admin center, either from the application gallery or a non-gallery app. For applications registered through the **App registrations** experience, follow the [Configure registered application SAML token encryption](#) guidance.

To configure enterprise application's SAML token encryption, follow these steps:

1. Obtain a public key certificate that matches a private key configured in the application.

Create an asymmetric key pair to use for encryption. Or, if the application supplies a public key to use for encryption, follow the application's instructions to download the X.509 certificate.

The public key should be stored in an X.509 certificate file in .cer format. You can copy the contents of the certificate file to a text editor and save it as a .cer file. The certificate file should contain only the public key and not the private key.

If the application uses a key that you created for your instance, follow the instructions provided by your application for installing the private key that the application is to use to decrypt tokens from your Microsoft Entra tenant.

2. Add the certificate to the application configuration in Microsoft Entra ID.

Configure token encryption in the Microsoft Entra admin center

You can add the public cert to your application configuration within the Microsoft Entra admin center.

1. Sign in to the Microsoft Entra admin center [as at least a Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. Enter the name of the existing application in the search box, and then select the application from the search results.
4. On the application's page, select **Token encryption**.

(!) Note

The **Token encryption** option is only available for SAML applications that have been set up from the **Enterprise applications** pane in the Microsoft Entra admin center, either from the application gallery or a non-gallery app. For other applications, this option is disabled.

5. On the **Token encryption** page, select **Import Certificate** to import the .cer file that contains your public X.509 certificate.

The screenshot shows the Microsoft Entra admin center interface. At the top, there are two buttons: 'Import Certificate' (highlighted with a red box) and 'Got feedback?'. Below these, a message says 'Please import and make active a certificate to enable token encryption'. A note below explains that SAML token encryption enables encrypted SAML assertions between Microsoft Entra and the application, providing assurance against interception. At the bottom, there is a table header with columns: Status, Key Id, Start Date, Expiration Date, and Thumbprint.

Status	Key Id	Start Date	Expiration Date	Thumbprint
--------	--------	------------	-----------------	------------

6. Once the certificate is imported, and the private key is configured for use on the application side, activate encryption by selecting the ... next to the thumbprint status, and then select **Activate token encryption** from the options in the dropdown menu.
7. Select **Yes** to confirm activation of the token encryption certificate.
8. Confirm that the SAML assertions emitted for the application are encrypted.

To deactivate token encryption in the Microsoft Entra admin center

1. In the Microsoft Entra admin center, browse to **Identity > Applications > Enterprise applications > All applications**, and then select the application that has SAML token encryption enabled.

2. On the application's page, select **Token encryption**, find the certificate, and then select the ... option to show the dropdown menu.
3. Select **Deactivate token encryption**.

Configure registered application SAML token encryption

This section describes how to configure a registered application's SAML token encryption. These applications are set up from the **App registrations** pane in the Microsoft Entra admin center. For enterprise application, follow the [Configure enterprise application SAML token encryption](#) guidance.

Encryption certificates are stored on the application object in Microsoft Entra ID with an `encrypt` usage tag. You can configure multiple encryption certificates and the one that's active for encrypting tokens is identified by the `tokenEncryptionKeyId` attribute.

You need the application's object ID to configure token encryption using Microsoft Graph API or PowerShell. You can find this value programmatically, or by going to the application's **Properties** page in the Microsoft Entra admin center and noting the **Object ID** value.

When you configure a keyCredential using Graph, PowerShell, or in the application manifest, you should generate a GUID to use for the `keyId`.

To configure token encryption for an application registration, follow these steps:

Portal

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Identity > Applications > App registrations > All applications**.
3. Enter the name of the existing application in the search box, and then select the application from the search results.
4. In the application's page, select **Manifest** to edit the [application manifest](#).

The following example shows an application manifest configured with two encryption certificates, and with the second selected as the active one using the `tokenEncryptionKeyId`.

JSON

```
{
  "id": "00aa00aa-bb11-cc22-dd33-44ee44ee44ee",
  "accessTokenAcceptedVersion": null,
  "allowPublicClient": false,
  "appId": "00001111-aaaa-2222-bbbb-3333cccc4444",
  "appRoles": [],
  "oauth2AllowUrlPathMatching": false,
  "createdDateTime": "2017-12-15T02:10:56Z",
  "groupMembershipClaims": "SecurityGroup",
  "informationalUrls": {
    "termsOfService": null,
    "support": null,
    "privacy": null,
    "marketing": null
  },
  "identifierUris": [
    "https://testapp"
  ],
  "keyCredentials": [
    {
      "customKeyIdentifier": "Tog/01Hv1LtdsbPU5nPphbMduD=",
      "endDate": "2039-12-31T23:59:59Z",
      "keyId": "aaaaaaaa-0b0b-1c1c-2d2d-333333333333",
      "startDate": "2018-10-25T21:42:18Z",
      "type": "AsymmetricX509Cert",
      "usage": "Encrypt",
      "value": <Base64EncodedKeyFile>
      "displayName": "CN=SAMLEncryptTest"
    },
    {
      "customKeyIdentifier": "A1bC2dE3fH4iJ5kL6mN7oP8qR9sT0u=",
      "endDate": "2039-12-31T23:59:59Z",
      "keyId": "bbbbbbbb-1c1c-2d2d-3e3e-444444444444",
      "startDate": "2018-10-25T21:42:18Z",
      "type": "AsymmetricX509Cert",
      "usage": "Encrypt",
      "value": <Base64EncodedKeyFile>
      "displayName": "CN=SAMLEncryptTest2"
    }
  ],
  "knownClientApplications": [],
  "logoUrl": null,
  "logoutUrl": null,
  "name": "Test SAML Application",
  "oauth2AllowIdTokenImplicitFlow": true,
  "oauth2AllowImplicitFlow": false,
  "oauth2Permissions": [],
  "oauth2RequirePostResponse": false,
  "orgRestrictions": [],
  "parentalControlSettings": {
    "countriesBlockedForMinors": [],
    "legalAgeGroupRule": "Allow"
  }
}
```

```
        },
        "passwordCredentials": [],
        "preAuthorizedApplications": [],
        "publisherDomain": null,
        "replyUrlsWithType": [],
        "requiredResourceAccess": [],
        "samlMetadataUrl": null,
        "signInUrl": "https://127.0.0.1:444/applications/default.aspx?
metadata=customappssso|ISV9.1|primary|z"
        "signInAudience": "AzureADMyOrg",
        "tags": [],
        "tokenEncryptionKeyId": "bbbbbbbb-1c1c-2d2d-3e3e-444444444444"
    }
```

Related content

- Find out [How Microsoft Entra ID uses the SAML protocol](#)
- Learn the format, security characteristics, and contents of [SAML tokens in Microsoft Entra ID](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

End-user experiences for applications

Article • 04/29/2025

Microsoft Entra ID provides several customizable ways to deploy applications to end users in your organization:

- Microsoft Entra My Apps
- Microsoft 365 application launcher
- Direct sign-on to federated apps
- Deep links to federated, password-based, or existing apps

Which method you choose to deploy in your organization is your discretion.

Microsoft Entra My Apps

My Apps is a web-based portal that allows an organization user in Microsoft Entra ID to view and launch apps which they're granted access to by an admin. If you're an end user with [Microsoft Entra ID P1 or P2](#), you can also utilize self-service group management capabilities through My Apps.

By default, all applications are listed together on a single page. But you can use collections to group together related applications and present them on a separate tab, making them easier to find. For example, you can use collections to create logical groupings of applications for specific job roles, tasks, projects, and so on. For information, see [Create collections on the My Apps portal](#).

[My Apps](#) is separate from the Microsoft Entra admin center and doesn't require users to have an Azure subscription or Microsoft 365 subscription.

For more information on Microsoft Entra My Apps, see the [introduction to My Apps](#).

Microsoft 365 application launcher

Microsoft 365 application launcher is the recommended app launching solution for organizations using Microsoft 365.

For more information about the Office 365 application launcher, see [Have your app appear in the Office 365 app launcher](#).

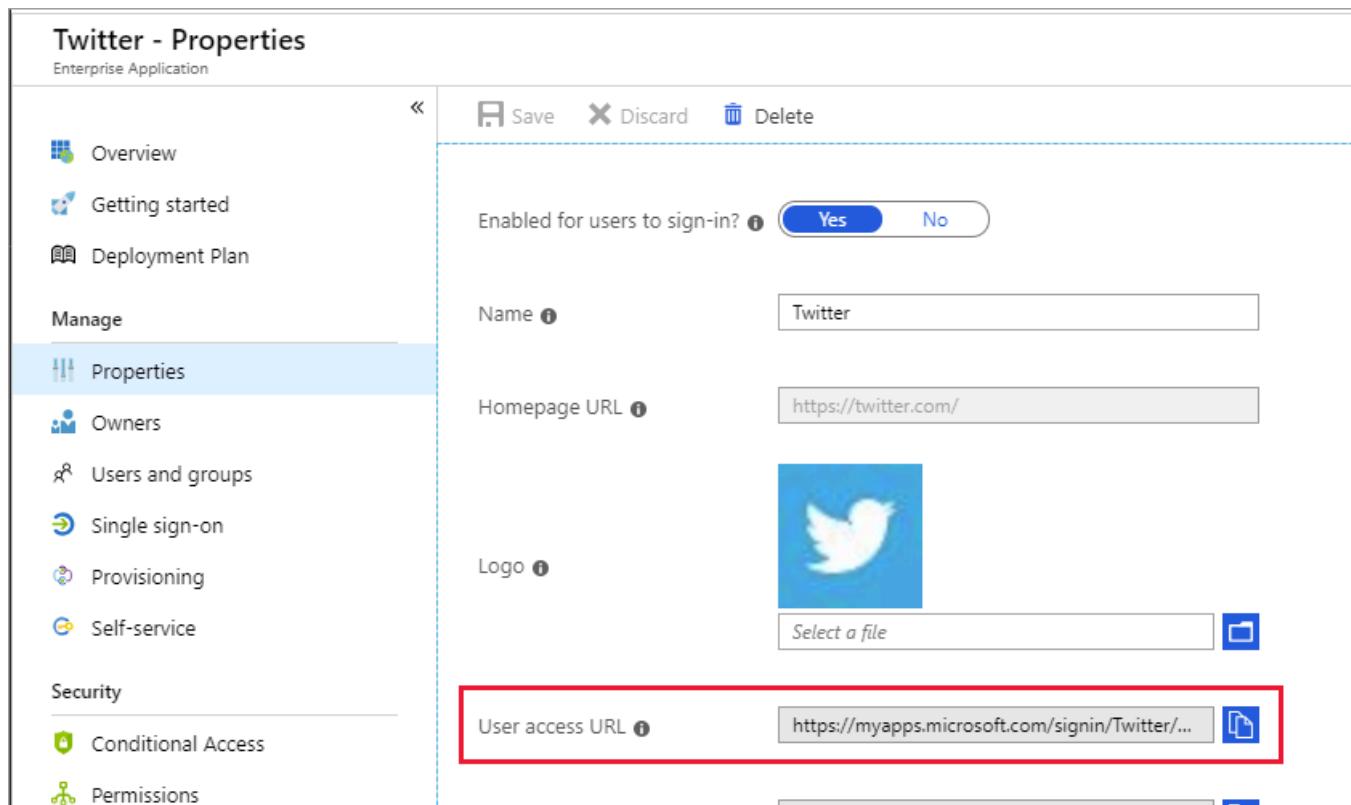
Direct sign-on to federated apps

Most federated applications that support SAML 2.0, WS-Federation, or OpenID connect also support the ability for users to start at the application. The users then get signed in through Microsoft Entra ID either by automatic redirection or by selecting a link to sign in. Direct sign-on is a service provider-initiated sign-on, and most federated applications in Microsoft Entra application gallery support it. See the documentation linked from the app's single sign-on configuration wizard in the Microsoft Entra admin center for details.

Direct sign-on links

Microsoft Entra ID also supports direct single sign-on links to individual applications that support password-based single sign-on, linked single sign-on, and any form of federated single sign-on.

Direct sign-on links are crafted URLs that send a user through the Microsoft Entra sign-in process for a specific application. The user doesn't need to launch the application from My Apps or Microsoft 365. These **User access URLs** can be found under the properties of available enterprise applications. In the Microsoft Entra admin center, select **Entra ID > Enterprise apps**. Select the application, and then select **Properties**.



The screenshot shows the 'Twitter - Properties' page in the Microsoft Entra Admin Center. The left sidebar lists various application management sections: Overview, Getting started, Deployment Plan, Manage (selected), Properties (highlighted with a blue background), Owners, Users and groups, Single sign-on, Provisioning, Self-service, Security, Conditional Access, and Permissions. The main right pane displays the application's details. At the top are Save, Discard, and Delete buttons. Below them is a section titled 'Enabled for users to sign-in?' with Yes and No buttons, where Yes is selected. The 'Name' field contains 'Twitter'. The 'Homepage URL' field contains 'https://twitter.com/'. A logo icon for Twitter is shown, and there is a 'Select a file' button for changing the logo. At the bottom, the 'User access URL' field contains 'https://myapps.microsoft.com/signin/Twitter/...' and has a copy icon to its right. This URL is highlighted with a red rectangular box.

Direct sign-on links can be copied and pasted anywhere you want to provide a sign-in link to the selected application. They can be placed in an email, or in any custom web-based portal that you set up for user application access. The following URL is an example of a Microsoft Entra ID direct single sign-on URL for X:

<https://myapps.microsoft.com/signin/X/230848d52c8745d4b05a60d29a40fcfd>

Similar to organization-specific URLs for My Apps, you can further customize direct sign-on URL by adding one of the active or verified domains for your directory after the *myapps.microsoft.com* domain. Customizing direct sign-on URL ensures any organizational branding is loaded immediately on the sign-in page without the user needing to enter their user ID first:

```
https://myapps.microsoft.com/contosobuild.com/signin/X/230848d52c8745d4b05a60d29a40fc
```

When an authorized user selects one of these application-specific links, they first see their organizational sign-in page (assuming they aren't already signed in). After sign-in, they're redirected to their app without stopping at My Apps first. If the user is missing prerequisites to access the application, such as the password-based single sign browser extension, then the link prompts the user to install the missing extension. The link URL also remains constant if the single sign-on configuration for the application changes.

These links use the same access control mechanisms as My Apps and Microsoft 365. Only those users or groups who are assigned to the application in the Microsoft Entra admin center are able to successfully authenticate. However, any user who is unauthorized sees a message explaining that they aren't granted access. The unauthorized user is given a link to load My Apps to view available applications that they do have access to.

Manage preview settings

As an admin, you can choose to try out new app launcher features while they are in preview. Enabling a preview feature means that the feature is turned on for your organization. The preview feature is also reflected in the My Apps portal and other app launchers for all your users.

To enable or disable previews for your app launchers:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Entra ID > Enterprise apps**.
3. On the left menu, select **App launchers**, then select **Settings**.
4. Under **Preview settings**, toggle the checkboxes for the previews you want to enable or disable. To opt into a preview, toggle the associated checkbox to the checked state. To opt out of a preview, toggle the associated checkbox to the unchecked state.
5. Select **Save**. Wait a few minutes for the changes to take effect. Navigate to the My Apps portal and verify that the preview you enabled or disabled is reflected.

Related content

- Quickstart Series on Application Management
- What is single sign-on?
- Integrating Microsoft Entra ID with applications getting started guide

Cloud app visibility and control

Article • 10/23/2023

To get the full benefit of cloud apps and services, an IT team must find the right balance of supporting access while maintaining control to protect critical data. Microsoft Defender for Cloud Apps provides rich visibility, control over data travel, and sophisticated analytics to identify and combat cyber threats across all your Microsoft and third-party cloud services.

Discover and manage shadow IT in your network

When IT admins are asked how many cloud apps they think their employees use, on average they say 30 or 40, when in reality, the average is over 1,000 separate apps being used by employees in your organization. Shadow IT helps you know and identify which apps are being used and what your risk level is. Eighty percent of employees use unsanctioned apps that no one has reviewed and may not be compliant with your security and compliance policies. And because your employees are able to access your resources and apps from outside your corporate network, it's no longer enough to have rules and policies on your firewalls.

Use Microsoft Cloud App Discovery (a Microsoft Entra ID P1 feature) to discover which apps are being used, explore the risk of these apps, configure policies to identify new risky apps, and unsanction these apps in order to block them natively using your proxy or firewall appliance.

- Discover and identify Shadow IT
- Evaluate and analyze
- Manage your apps
- Advanced Shadow IT discovery reporting
- Control sanctioned apps

Learn more

- [Discover and manage shadow IT in your network](#)
- [Discovered apps with Defender for Cloud Apps](#)

User session visibility and control

In today's workplace, it's often not enough to know what's happening in your cloud environment after the fact. You want to stop breaches and leaks in real time before employees intentionally or inadvertently put your data and your organization at risk. Together with Microsoft Entra ID, Microsoft Defender for Cloud Apps delivers these capabilities in a holistic and integrated experience with Conditional Access App Control.

Session control uses a reverse proxy architecture and is uniquely integrated with Microsoft Entra Conditional Access. Microsoft Entra Conditional Access allows you to enforce access controls on your organization's apps based on certain conditions. The conditions define who (user or group of users) and what (which cloud apps) and where (which locations and networks) a Conditional Access policy is applied to. After you've determined the conditions, you can route users to Defender for Cloud Apps where you can protect data in real time.

With this control you can:

- Control file downloads
- Monitor B2B scenarios
- Control access to files
- Protect documents on download

Learn more

- [Protect apps with Session Control in Defender for Cloud Apps](#)

Advanced app visibility and controls

App connectors use the APIs of app providers to enable greater visibility and control by Microsoft Defender for Cloud Apps over the apps you connect to. Defender for Cloud Apps leverages the APIs provided by the cloud provider. Each service has its own framework and API limitations such as throttling, API limits, dynamic time-shifting API windows, and others. The Defender for Cloud Apps product team worked with these services to optimize the use of APIs and provide the best performance. Taking into account different limitations services impose on their APIs, the Defender for Cloud Apps engines use their maximum allowed capacity. Some operations, such as scanning all files in the tenant, require numerous API calls so they're spread over a longer period. Expect some policies to run for several hours or days.

Learn more

- [Connect apps in Defender for Cloud Apps](#)

Next steps

- Discover and manage shadow IT in your network
- Discovered apps with Defender for Cloud Apps
- Protect apps with Session Control in Defender for Cloud Apps
- Connect apps in Defender for Cloud Apps

Home Realm Discovery for an Application

Article • 11/29/2024

Home Realm Discovery (HRD) enables Microsoft Entra ID to identify the appropriate identity provider (IdP) for user authentication during sign-in. When users sign in to a Microsoft Entra tenant to access a resource or the common sign-in page, they enter a user name (UPN). Microsoft Entra ID uses this information to determine the correct sign-in location.

Users are directed to one of the following identity providers for authentication:

- The user's home tenant (which might be the same as the resource tenant).
- Microsoft account, if the user is a guest in the resource tenant using a consumer account.
- An on-premises identity provider like Active Directory Federation Services (ADFS).
- Another identity provider federated with the Microsoft Entra tenant.

Auto-acceleration

Organizations might configure domains in their Microsoft Entra tenant to federate with another IdP, such as ADFS, for user authentication. When users sign in to an application, they initially see a Microsoft Entra sign-in page. If they belong to a federated domain, they're redirected to the IdP's sign-in page for that domain. Administrators might want to bypass the initial Microsoft Entra ID page for specific applications, a process known as "sign-in auto-acceleration."

Microsoft advises against configuring auto-acceleration as it can hinder stronger authentication methods like FIDO and collaboration. For more information, see [Enable passwordless security key sign-in](#). To learn how to prevent sign-in auto-acceleration, see [Disable auto-acceleration sign-in](#).

Auto-acceleration can streamline sign-in for tenants federated with another IdP. You can configure it for individual applications. To learn how to force auto-acceleration using HRD, See [Configure auto-acceleration](#).

Note

Configuring an application for auto-acceleration prevents users from using managed credentials (like FIDO) and guest users from signing in. Directing users to

a federated IdP for authentication bypasses the Microsoft Entra sign-in page, preventing guest users from accessing other tenants or external IdPs like Microsoft accounts.

Control auto-acceleration to a federated IdP in three ways:

- Use a domain hint on authentication requests for an application.
- Configure an HRD policy to [force auto-acceleration](#).
- Configure an HRD policy to [ignore domain hints](#) for specific applications or domains.

Domain Confirmation Dialog

Starting April 2023, organizations using auto-acceleration or smart links might encounter a new screen in the sign-in UI, called the Domain Confirmation Dialog. This screen is part of Microsoft's security hardening efforts and requires users to confirm the domain of the tenant they're signing into.

What You Need to Do

When you see the Domain Confirmation Dialog:

- **Check the domain:** Verify the domain name on the screen matches the organization you intend to sign in to, such as `contoso.com`.
 - **If you recognize the domain**, select **Confirm** to proceed.
 - **If you don't recognize the domain**, cancel the sign-in process and contact your IT Admin for assistance.

Components of the Domain Confirmation Dialog

The following screenshot shows an example of what the domain confirmation dialog could look like for you:



← kelly@contoso.com

Do you trust contoso.com?

You're about to sign-in with your contoso.com account. If you do not recognize this account, contact your admin.

[Why am I seeing this?](#)

Cancel

Confirm

The identifier at the top of the dialog, `kelly@contoso.com`, represents the identifier used to sign-in. The tenant domain listed in the dialog's header and subheader shows the domain of the account's home tenant.

This dialog might not appear for every instance of auto-acceleration or smart links. Frequent domain confirmation dialogs might occur if your organization clears cookies due to browser policies. The Domain Confirmation Dialog shouldn't cause application breakages as Microsoft Entra ID manages the auto-acceleration sign-in flow.

Domain Hints

Domain hints are directives in authentication requests from applications that can accelerate users to their federated IdP sign-in page. Multitenant applications can use them to direct users to the branded Microsoft Entra sign-in page for their tenant.

For example, "largeapp.com" might allow access via a custom URL "contoso.largeapp.com" and include a domain hint to contoso.com in the authentication request.

Domain hint syntax varies by protocol:

- **WS-Federation:** `whr` query string parameter, for example, `whr=contoso.com`.

- **SAML:** SAML authentication request with a domain hint or `whr=contoso.com`.
- **OpenID Connect:** `domain_hint` query string parameter, for example, `domain_hint=contoso.com`.

Microsoft Entra ID redirects sign-in to the configured IDP for a domain if **both** of the following cases are true:

- A domain hint is included in the authentication request.
- The tenant is federated with that domain.

If the domain hint doesn't refer to a verified federated domain, it can be ignored.

Note

A domain hint in an authentication request overrides auto-acceleration set for the application in HRD policy.

HRD Policy for Auto-acceleration

Some applications don't allow configuration of authentication requests. In such cases, it's not possible to use domain hints to control auto-acceleration. Use [Home Realm Discovery](#) policy to configure auto-acceleration.

HRD Policy to Prevent Auto-acceleration

Some Microsoft and SaaS applications automatically include domain hints, which can disrupt managed credential rollouts like FIDO. Use [Home Realm Discovery policy to ignore domain hints](#) from certain apps or domains during managed credential rollouts.

Enable direct ROPC authentication of federated users for legacy applications

Best practice is for applications to use Microsoft Entra libraries and interactive sign-in for user authentication. Legacy applications using Resource Owner Password Credentials (ROPC) grants might submit credentials directly to Microsoft Entra ID without understanding federation. They don't perform HRD or interact with the correct federated endpoint. You can use [Home Realm Discovery policy to enable specific legacy applications](#) to authenticate directly with Microsoft Entra ID. This option works, provided Password Hash Sync is enabled.

Important

Only enable direct authentication if Password Hash Sync is active and it's acceptable to authenticate the application without on-premises IdP policies. If Password Hash Sync or Directory Synchronization with AD Connect is disabled, remove this policy to prevent direct authentication with stale password hashes.

Set HRD Policy

To set an HRD policy on an application for federated sign-in auto-acceleration or direct cloud-based applications:

- Create an HRD policy.
- Locate the service principal to attach the policy.
- Attach the policy to the service principal.

Policies take effect for a specific application when attached to a service principal. Only one HRD policy can be active on a service principal at a time. Use [Microsoft Graph PowerShell cmdlets](#) to create and manage HRD policy.

Example HRD policy definition:

JSON

```
{  
  "HomeRealmDiscoveryPolicy": {  
    "AccelerateToFederatedDomain": true,  
    "PreferredDomain": "federated.example.edu",  
    "AllowCloudPasswordValidation": false  
  }  
}
```

- **AccelerateToFederatedDomain**: Optional. If false, the policy doesn't affect auto-acceleration. If true and there's one verified federated domain, users are directed to the federated IdP. If multiple domains exist, specify **PreferredDomain**.
- **PreferredDomain**: Optional. Indicates a domain for acceleration. Omit if only one federated domain exists. If omitted with multiple domains, the policy has no effect.
- **AllowCloudPasswordValidation**: Optional. If true, allows federated user authentication via username/password credentials directly to Microsoft Entra token endpoint, requiring Password Hash Sync.

Additional tenant-level HRD options:

- **AlternateIdLogin**: Optional. Enables [AlternateLoginID](#) for email sign-in instead of UPN at the Microsoft Entra sign-in page. Relies on users not being auto-accelerated to a federated IDP.
- **DomainHintPolicy**: Optional complex object that [prevents domain hints from auto-accelerating users](#) to federated domains. Ensures applications sending domain hints don't prevent cloud-managed credential sign-ins.

Priority and Evaluation of HRD Policies

HRD policies can be assigned to organizations and service principals, allowing multiple policies to apply to an application. Microsoft Entra ID determines precedence using these rules:

- If a domain hint is present, the tenant's HRD policy checks if domain hints should be ignored. If allowed, the domain hint behavior is used.
- If a policy is explicitly assigned to the service principal, it's enforced.
- If no domain hint or service principal policy exists, a policy assigned to the parent organization is enforced.
- If no domain hint or policies are assigned, default HRD behavior applies.

Next Steps

- [Configure sign-in behavior for an application using a Home Realm Discovery policy](#)
- [Disable auto-acceleration to a federated IDP during user sign-in with Home Realm Discovery policy](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

My Apps portal overview

Article • 10/31/2024

My Apps is a web-based portal that is used for managing and launching applications in Microsoft Entra ID. To work with applications in My Apps, use an organizational account in Microsoft Entra ID and obtain access granted by the Microsoft Entra administrator.

My Apps is separate from the Microsoft Entra admin center and doesn't require users to have an Azure subscription or Microsoft 365 subscription.

Users access the My Apps portal to:

- Discover applications to which they have access
- Request new applications that the organization supports for self-service
- Create personal collections of applications
- Manage access to applications

The following conditions determine whether an application in the enterprise applications list in the Microsoft Entra admin center appears to a user or group in the My Apps portal:

- The application is set to be visible in its properties
- The application is assigned to the user or group

ⓘ Note

The **Users can only see Office 365 apps in the Office 365 portal** property in the Microsoft Entra admin center can affect whether users can only see Office 365 applications in the Office 365 portal. If this setting is set to **No**, then users will be able to see Office 365 applications in both the My Apps portal and the Office 365 portal. This setting can be found under **Manage in Enterprise applications > User settings**.

Administrators can configure:

- Consent experiences including terms of service
- Self-service application discovery and access requests
- Collections of applications
- Company and application branding

Understand application properties

Properties that are defined for an application can affect how the user interacts with it in the My Apps portal.

- **Enabled for users to sign in?** – If this property is set to **Yes**, then assigned users are able to sign into the application from the My Apps portal.
- **Name** - The name of the application that users see on the My Apps portal. Administrators see the name when they manage access to the application.
- **Homepage URL** -The URL that is launched when the application is selected in the My Apps portal.
- **Logo** - The application logo that users see on the My Apps portal.
- **Visible to users** - Makes the application visible in the My Apps portal. When this value is set to **Yes**, applications still don't appear in the My Apps portal if they don't yet have users or groups assigned to it. Only assigned users are able to see the application in the My Apps portal.

For more information, see [Properties of an enterprise application](#).

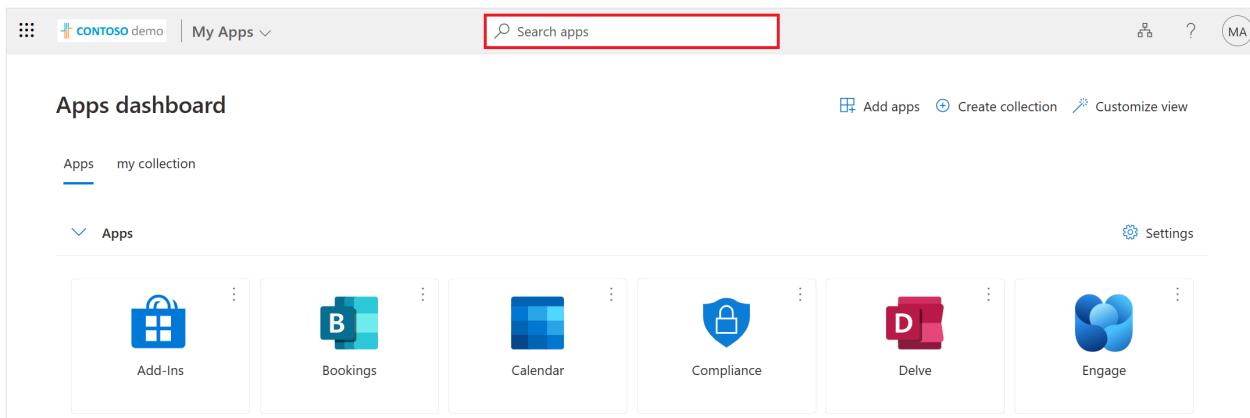
Discover applications

When signed in to the [My Apps](#) portal, the applications that are made visible are shown. For an application to be visible in the My Apps portal, set the appropriate properties in the [Microsoft Entra admin center](#). Also in the Microsoft Entra admin center, assign a user or group with the appropriate members.

In the My Apps portal, to search for an application, enter an application name in the search box at the top of the page to find an application. The applications that are listed can be formatted in **List view** or a **Grid view**.

Note

End users are no longer be able to add password SSO apps in My Apps. If you need to add a password SSO app for your end users, you can do so in the Microsoft Entra admin center. For more information, see [Add an application for password-based single sign-on](#).



The screenshot shows the Microsoft 365 Apps dashboard. At the top, there's a header with a search bar labeled "Search apps". Below the header, the title "Apps dashboard" is displayed. There are two tabs: "Apps" (which is selected) and "my collection". On the right side of the dashboard, there are several application tiles: "Add-Ins" (blue shopping bag icon), "Bookings" (blue "B" icon), "Calendar" (blue calendar icon), "Compliance" (blue shield icon), "Delve" (blue "D" icon), and "Engage" (blue circular icon). A "Settings" gear icon is also present.

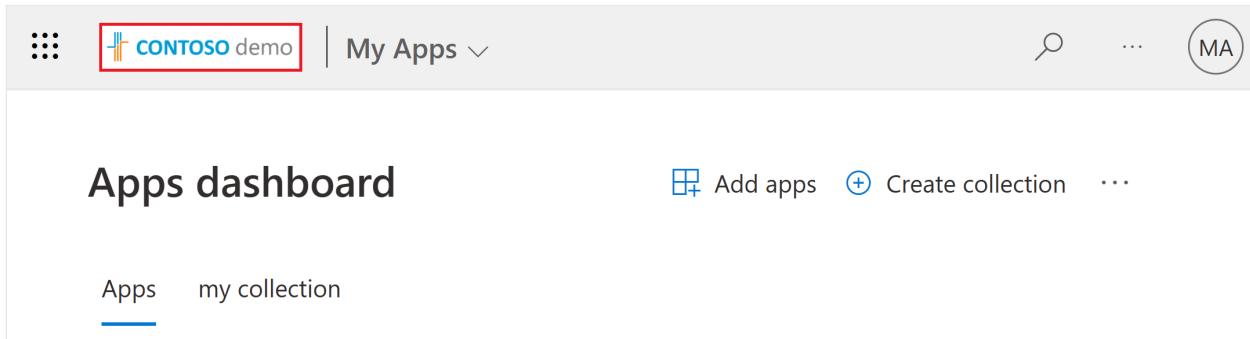
ⓘ Important

It can take several minutes for an application to appear in the My Apps portal after it has been added to the tenant in the Microsoft Entra admin center. There may also be a delay in how soon users can access the application after it has been added.

Applications can be hidden. For more information, see [Hide an Enterprise application](#).

Assign company branding

In the Microsoft Entra admin center, define the logo and name for the application to represent company branding in the My Apps portal. The banner logo appears at the top of the page, such as the following Contoso demo logo.



The screenshot shows the Microsoft 365 Apps dashboard with the "Contoso demo" logo prominently displayed in the top banner. The rest of the interface is identical to the one shown in the first screenshot, including the "Apps" tab selected, the "Add-Ins", "Bookings", "Calendar", "Compliance", "Delve", and "Engage" tiles, and the "Settings" gear icon.

For more information, see [Add branding to your organization's sign-in page](#).

Manage access to applications

Multiple factors affect how and whether an application is accessed by users. Permissions that are assigned to the application can affect what can be done with it. Applications can be configured to allow self-service access, or access can be only granted by an administrator of the tenant.

My Apps Secure Sign-in Extension

Install the My Apps secure sign-in extension to sign in to some applications. The extension is required for sign-in to password-based SSO applications, or to applications that are accessed by Microsoft Entra application proxy. Users are prompted to install the extension when they first launch the password-based single sign-on or an Application Proxy application.

To integrate these applications, define a mechanism to deploy the extension at scale with supported browsers. Options include:

- User-driven download and configuration for Chrome, Microsoft Edge, or IE
- Configuration Manager for Internet Explorer

For applications that use password-based SSO or accessed by using Microsoft Entra application proxy, use Microsoft Edge mobile. For other applications, any mobile browser can be used. Be sure to enable password-based SSO in the mobile settings, which can be off by default. For example, **Settings > Privacy and Security > Microsoft Entra Password SSO**.

To download and install the extension:

- **Microsoft Edge** - From the Microsoft Store, go to the [My Apps Secure Sign-in Extension](#) feature, and then select **Get to get the extension for Microsoft Edge legacy browser**.
- **Google Chrome** - From the Chrome Web Store, go to the [My Apps Secure Sign-in Extension](#) feature, and then select **Add to Chrome**.

An icon is added to the right of the address bar, which enables sign in and customization of the extension.

Note

Sign-in into the extension is currently not supported for Guest B2B Microsoft Accounts (MSA).

Permissions

Permissions that are granted to an application can be reviewed by selecting the upper right corner of the tile that represents the application and then selecting **Manage your application**.

The permissions that are shown are consented to by an administrator or are consented to by the user. Permissions consented to by the user can be revoked by the user.

Self-service access

Access can be granted on a tenant level, assigned to specific users, or from self-service access. Before users can self-discover applications from the My Apps portal, enable self-service application access in the Microsoft Entra admin center. This feature is available for applications when added using these methods:

- The Microsoft Entra application gallery
- Microsoft Entra application proxy
- Using user or admin consent

Enable users to discover and request access to applications by using the My Apps portal. To do so, complete the following tasks in the Microsoft Entra admin center:

- Enable self-service group management
- Enable the application for single sign-on
- Create a group for application access

When users request access, they request access to the underlying group, and group owners can be delegated permission to manage the group membership and application access. Approval workflows are available for explicit approval to access applications. Users who are approvers receive notifications within the My Apps portal when there are pending requests for access to the application.

For more information, see [Enable self-service application assignment](#)

Single sign-on

Enable single sign-on (SSO) in the Microsoft Entra admin center for all applications that are made available in the My Apps portal whenever possible. If SSO is set up, users have a seamless experience without the need to enter their credentials. To learn more, see [Single sign-on options in Microsoft Entra ID](#).

Applications can be added by using the Linked SSO option. Configure an application tile that links to the URL of the existing web application. Linked SSO allows the direction of users to the My Apps portal without migrating all the applications to Microsoft Entra SSO. Gradually move to Microsoft Entra SSO-configured applications to prevent disrupting the users' experience.

For more information, see [Add linked single sign-on to an application](#).

Create collections

By default, all applications are listed together on a single page. Collections can be used to group together related applications and present them on a separate tab, making them easier to find. For example, use collections to create logical groupings of applications for specific job roles, tasks, projects, and so on. Every application to which a user has access appears in the default Apps collection, but a user can remove applications from the collection.

Users can also customize their experience by:

- Creating their own application collections
- Hiding and reordering application collections

Applications can be hidden from the My Apps portal by a user or administrator. A hidden application can still be accessed from other locations, such as the Microsoft 365 portal. Only 950 applications to which a user has access can be accessed through the My Apps portal.

For more information, see [Create collections on the My Apps portal](#).

Important

In case there is domain federation and for the user to be redirected to an external federation endpoint for authentication, the request to My Apps must contain a domain hint URL parameter such as "domain_hint=example.com".

Next steps

Learn more about application management in [What is enterprise application management?](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Overview of user and admin consent

Article • 03/18/2025

Consent is a process where users can grant permission for an application to access a protected resource. To indicate the level of access required, an application requests the API permissions it requires. For example, an application can request the permission to see a signed-in user's profile and read the contents of the user's mailbox.

Consent can be initiated in various ways. For example, users can be prompted for consent when they attempt to sign in to an application for the first time. Depending on the permissions they require, some applications might require an administrator to be the one who grants consent.

In this article, you learn the foundational concepts and scenarios around user and admin consent in Microsoft Entra ID.

User consent

A user can authorize an application to access some data at the protected resource, while acting as that user. The permissions that allow this type of access are called "delegated permissions."

User consent is initiated when a user signs in to an application. After the user provides sign-in credentials, they're checked to determine whether consent is already granted. If no previous record of user or admin consent for the required permissions exists, the user is directed to the consent prompt window to grant the application the requested permissions.

User consent by nonadministrators is possible only in organizations where user consent is allowed for the application and for the set of permissions the application requires. If user consent is disabled, or users aren't allowed to consent for the requested permissions, they aren't prompted for consent. If users are allowed to consent and they accept the requested permissions, the consent is recorded. The users usually don't have to consent again on future sign-ins to the same application.

User consent settings

Users are in control of their data. A Privileged Administrator can configure whether nonadministrator users are allowed to grant user consent to an application. This setting can take into account aspects of the application and the application's publisher, and the

permissions being requested. For step by step instructions on how to configure user consent, see [Configure user consent settings](#).

As an administrator, you can choose whether user consent is allowed. If you choose to allow user consent, you can also choose what conditions must be met before a user can consent to an application.

By choosing which application consent policies apply for all users, you can set limits on when users are allowed to grant consent to applications. The consent policies also inform when users are required to request administrator review and approval. The Microsoft Entra admin center provides the following built-in options:

- *You can disable user consent.* Users can't grant permissions to applications. Users continue to sign in to applications they already consented to or to applications that administrators grant consent to on their behalf. However, they're not allowed to consent to new permissions to applications on their own. Only users who are granted a directory role that includes the permission to grant consent can consent to new applications.
- *Users can consent to applications from verified publishers or your organization, but only for permissions you select.* All users can consent only to applications published by a [verified publisher](#) and applications that are registered in your tenant. Users can consent only to the permissions classified as *low impact*. You must [classify permissions](#) to select which permissions users are allowed to consent to.
- *Users can consent to all applications.* This option allows all users to consent to any permissions that don't require admin consent, for any application.

For most organizations, one of the built-in options is appropriate. Some advanced customers might want more control over the conditions that govern when users are allowed to consent. These customers can [create custom app consent policy](#) and configure those policies to apply to user consent.

Admin consent

During admin consent, a Privileged Administrator might grant an application access on behalf of other users (usually, on behalf of the entire organization). Also during admin consent, applications or services provide direct access to an API, which is used by the application if there's no signed-in user. The specific role needed to grant admin consent differs based on the permissions requested, which are outlined in the [grant admin consent](#) article.

When your organization purchases a license or subscription for a new application, you might proactively want to set up the application so that all users in the organization can use it. To avoid the need for user consent, an administrator can grant consent for the application on behalf of all users in the organization.

After an administrator grants admin consent on behalf of the organization, users aren't prompted for consent for that application. In certain cases, a user might be prompted for consent even after an administrator grants consent. An example might be if an application requests another permission that the administrator hasn't granted.

Granting admin consent on behalf of an organization is a sensitive operation, potentially allowing the application's publisher access to significant portions of the organization's data, or the permission to do highly privileged operations. Examples of such operations might be role management, full access to all mailboxes or all sites, and full user impersonation.

Before you grant tenant-wide admin consent, ensure that you trust the application and the application publisher, for the level of access you're granting. If you aren't confident that you understand who controls the application and why the application is requesting the permissions, don't grant consent.

For guidance on evaluating admin consent requests, see [Evaluating a request for tenant-wide admin consent](#).

For step-by-step instructions for granting tenant-wide admin consent from the Microsoft Entra admin center, see [Grant tenant-wide admin consent to an application](#).

Grant consent on behalf of a specific user

Instead of granting consent for an entire organization, an admin can also use the [Microsoft Graph API](#) to grant consent to delegated permissions on behalf of a single user. For a detailed example that uses Microsoft Graph PowerShell, see [Grant consent on behalf of a single user by using PowerShell](#).

Limit user access to an application

User access to applications can still be limited, even when tenant-wide admin consent is already granted. Configure the application's properties to require user assignment to limit user access to the application. For more information, see [Methods for assigning users and groups](#).

For a broader overview, including how to handle other complex scenarios, see [Use Microsoft Entra ID for application access management](#).

Admin consent workflow

The admin consent workflow gives users a way to request admin consent for applications when they aren't allowed to consent themselves. When the admin consent workflow is enabled, users are presented with an "Approval required" window for requesting admin approval for access to the application.

After users submit the admin consent request, the admins who are designated as reviewers receive a notification. The users are notified after a reviewer acts on their request. For step-by-step instructions for configuring the admin consent workflow by using the Microsoft Entra admin center, see [configure the admin consent workflow](#).

How users request admin consent

After the admin consent workflow is enabled, users can request admin approval for an application that they're unauthorized to consent to. Here are the steps in the process:

- A user attempts to sign in to the application.
- An **Approval required** message appears. The user types a justification for needing access to the application and then selects "Request approval."
- A **Request sent** message confirms that the request was submitted to the admin. If the user sends several requests, only the first request is submitted to the admin.
- The user receives an email notification when the request is approved, denied, or blocked.

Related content

- [Configure user consent settings](#)
- [Configure the admin consent workflow](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

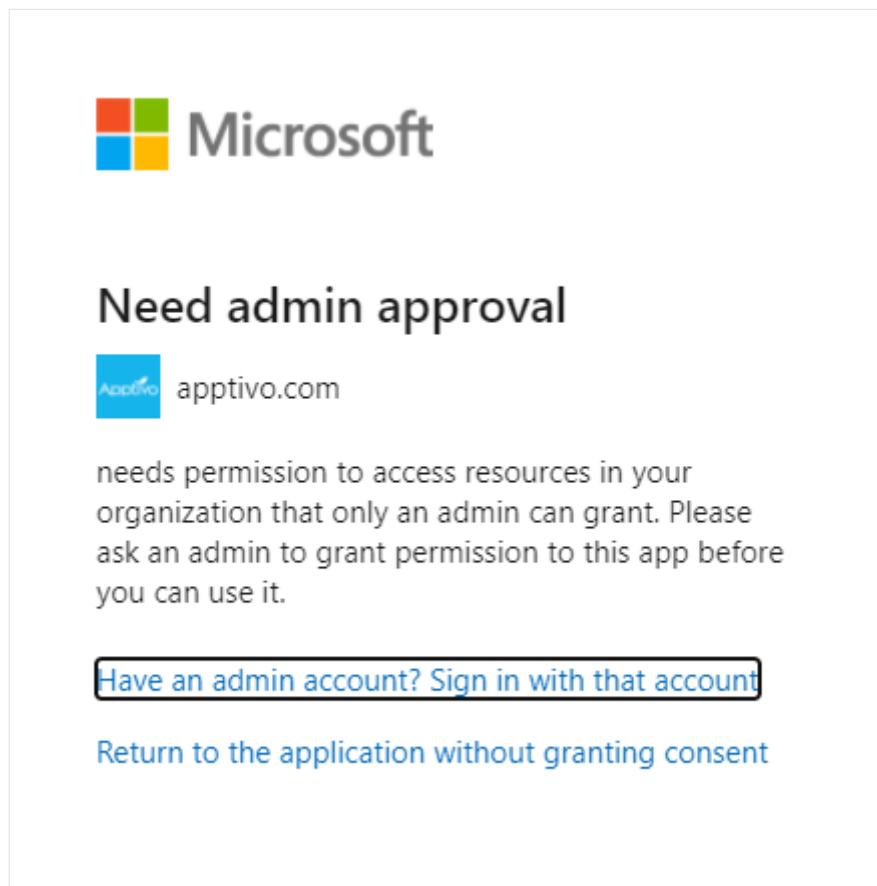
Overview of admin consent workflow

Article • 11/29/2024

There might be situations where your end-users need to consent to permissions for applications that they're creating or using with their work accounts. However, nonadmin users aren't allowed to consent to permissions that require admin consent. Also, users can't consent to applications when [user consent](#) is disabled in the user's tenant.

In such situations where user consent is disabled, an admin can grant users the ability to make requests for gaining access to applications by enabling the admin consent workflow. In this article, you learn about the user and admin experience when the admin consent workflow is on vs when it's off.

When attempting to sign in, users might see a consent prompt like the one in the following screenshot:



If the user doesn't know who to contact to grant them access, they might be unable to use the application. This situation also requires administrators to create a separate workflow to track requests for applications if they're open to receiving them. As an admin, the following options exist for you to determine how users consent to applications:

- Disable user consent. For example, a high school might want to turn off user consent so that the school IT administration has full control over all the applications in their tenant.
- Allow users to consent to the required permissions. The best practice is to keep user consent open if you have sensitive data in your tenant.
- If you still want to retain admin-only consent for certain permissions but want to assist your end-users in onboarding their application, you can use the admin consent workflow to evaluate and respond to admin consent requests. This way, you can have a queue of all the requests for admin consent for your tenant and can track and respond to them directly through the Microsoft Entra admin center. To learn how to configure the admin consent workflow, see [Configure the admin consent workflow](#).

How the admin consent workflow works

When you configure the admin consent workflow, your end users can request for consent directly through the prompt. The users might see a consent prompt like the one in the following screenshot:



Approval required



apptivo.com

This app requires your admin's approval to:

- ✓ Maintain access to data you have given it access to
- ✓ Read user contacts
- ✓ Have full access to user contacts
- ✓ Have full access to user calendars
- ✓ Read user calendars
- ✓ Read all users' basic profiles
- ✓ Sign in and read user profile
- ✓ Read user and shared tasks

Enter justification for requesting this app

[Sign in with another account](#)

Does this app look suspicious? [Report it here](#)

[Cancel](#)

[Request approval](#)

When an administrator responds to a request, the user receives an email alert informing them that the request is processed.

When the user submits a consent request, the request shows up in the admin consent request page in the Microsoft Entra admin center. Administrators and designated reviewers sign in to [view and act on the new requests](#). Reviewers only see consent requests that were created after they were designated as reviewers. Requests show up in the following two tabs in the admin consent requests pane:

- My pending: This tab shows any active requests that have the signed-in user designated as a reviewer. Although reviewers can block or deny requests, only people with the correct RBAC permissions to consent to the requested permissions can do so.
- All(Preview): All requests, active or expired, that exist in the tenant. Each request includes information about the application and the users requesting the

application.

Email notifications

If configured, all reviewers receive email notifications when:

- A new request is created
- A request expires
- A request is nearing the expiration date.

Requestors receive email notifications when:

- They submit a new request for access
- Their request expires
- Their request is denied or blocked
- Their request is approved

Audit logs

The following table outlines the scenarios and audit values available for the admin consent workflow.

 Expand table

Scenario	Audit Service	Audit Category	Audit Activity	Audit Actor	Audit log limitations
Admin enabling the consent request workflow	Access Reviews	UserManagement	Create governance policy template	App context	Currently you can't find the user context
Admin disabling the consent request workflow	Access Reviews	UserManagement	Delete governance policy template	App context	Currently you can't find the user context
Admin updating the consent workflow configurations	Access Reviews	UserManagement	Update governance policy template	App context	Currently you can't find the user context
End user creating an admin consent	Access Reviews	Policy	Create request	App context	Currently you can't find the user context

Scenario	Audit Service	Audit Category	Audit Activity	Audit Actor	Audit log limitations
request for an app					
Reviewers approving an admin consent request	Access Reviews	UserManagement	Approve all requests in business flow	App context	Currently you can't find the user context or the app ID that was granted admin consent.
Reviewers denying an admin consent request	Access Reviews	UserManagement	Approve all requests in business flow	App context	Currently you can't find the user context of the actor that denied an admin consent request

Next steps

- Enable the admin consent request workflow
- Review admin consent request
- Manage consent requests

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Manage consent to applications and evaluate consent requests

Article • 06/27/2024

Microsoft recommends that you [restrict user consent](#) to allow users to consent only for apps from verified publishers, and only for permissions that you select. For apps that don't meet these criteria, the decision-making process is centralized with your organization's security and identity administrator team.

After disabling or restricting user consent, you have several important steps to take to help keep your organization secure as you continue to allow business-critical applications to be used. These steps are crucial to minimize impact on your organization's support team and IT administrators, and to help prevent the use of unmanaged accounts in non-Microsoft applications.

This article provides guidance on managing consent to applications and evaluating consent requests in Microsoft's recommendations, including restricting user consent to verified publishers and selected permissions. It covers concepts such as process changes, education for administrators, auditing and monitoring, and managing tenant-wide admin consent.

Process changes and education

- Consider enabling the [admin consent workflow](#) to allow users to request administrator approval directly from the consent screen.
- Ensure that all administrators understand the:
 - [Permissions and consent framework](#)
 - How the [consent experience and prompts](#) work.
 - How to [evaluate a request for tenant-wide admin consent](#).
- Review your organization's existing processes for users to request administrator approval for an application, and update them if necessary. If processes are changed:
 - Update the relevant documentation, monitoring, automation, and so on.
 - Communicate process changes to all affected users, developers, support teams, and IT administrators.

Auditing and monitoring

- Audit apps and granted permissions in your organization to ensure that no unwarranted or suspicious applications are already granted access to data.
- Review the [Detect and Remediate Illicit Consent Grants in Office 365](#) article for more best practices and safeguards against suspicious applications that request OAuth consent.
- If your organization has the appropriate license:
 - Use other [OAuth application auditing features in Microsoft Defender for Cloud Apps](#).
 - Use [Azure Monitor Workbooks](#) to monitor permissions and consent-related activity. The *Consent Insights* workbook provides a view of apps by number of failed consent requests. This information can help you prioritize applications for administrators to review and decide whether to grant them admin consent.

Other considerations for reducing friction

To minimize impact on trusted, business-critical applications that are already in use, consider proactively granting administrator consent to applications that have a high number of user consent grants:

- Take an inventory of the apps already added to your organization with high usage, based on sign-in logs or consent grant activity. You can use a [PowerShell script](#) to quickly and easily discover applications with a large number of user consent grants.
- Evaluate the top applications to grant admin consent.

Important

Carefully evaluate an application before granting tenant-wide admin consent, even if many users in the organization have already consented for themselves.

- For each approved application, grant tenant-wide admin consent and consider restricting user access by [requiring user assignment](#).

Evaluate a request for tenant-wide admin consent

Granting tenant-wide admin consent is a sensitive operation. Permissions are granted on behalf of the entire organization, and they can include permissions to attempt highly

privileged operations. Examples of such operations are role management, full access to all mailboxes or all sites, and full user impersonation.

Before you grant tenant-wide admin consent, it's important to ensure that you trust the application, and the application publisher for the level of access you're granting. If you aren't confident that you understand who controls the application and why the application is requesting the permissions, don't grant consent.

When you're evaluating a request to grant admin consent, here are some recommendations to consider:

- Understand the [permissions and consent framework](#) in the Microsoft identity platform.
- Understand the difference between [delegated permissions and application permissions](#).

Application permissions allow the application to access the data for the entire organization, without any user interaction. Delegated permissions allow the application to act on behalf of a user who was signed into the application at some point.

- Understand the permissions that are being requested.

The permissions requested by the application are listed in the [consent prompt](#).

Expanding the permission title displays the permission's description. The description for application permissions generally ends in "without a signed-in user." The description for delegated permissions generally end with "on behalf of the signed-in user." Permissions for the Microsoft Graph API are described in [Microsoft Graph Permissions Reference](#). Refer to the documentation for other APIs to understand the permissions they expose.

If you don't understand a permission that's being requested, don't grant consent.

- Understand which application is requesting permissions and who published the application.

Be wary of malicious applications that try to look like other applications.

If you doubt the legitimacy of an application or its publisher, don't grant consent. Instead, seek confirmation (for example, directly from the application publisher).

- Ensure that the requested permissions are aligned with the features you expect from the application.

For example, an application that offers SharePoint site management might require delegated access to read all site collections, but it wouldn't necessarily need full access to all mailboxes, or full impersonation privileges in the directory.

If you suspect that the application is requesting more permissions than it needs, don't grant consent. Contact the application publisher to obtain more details.

Grant tenant-wide admin consent

For step-by-step instructions for granting tenant-wide admin consent from the Microsoft Entra admin center, see [Grant tenant-wide admin consent to an application](#).

Revoke tenant wide admin consent

To revoke tenant-wide admin consent, you can review and revoke the permissions previously granted to the application. For more information, see [review permissions granted to applications](#). You can also remove user's access to the application by [disabling user sign-in to application](#) or by [hiding the application](#) so that it doesn't appear in the My apps portal.

Grant consent on behalf of a specific user

Instead of granting consent for the entire organization, an administrator can also use the [Microsoft Graph API](#) to grant consent to delegated permissions on behalf of a single user. For a detailed example that uses Microsoft Graph PowerShell, see [Grant consent on behalf of a single user by using PowerShell](#).

Limit user access to applications

User access to applications can still be limited even when tenant-wide admin consent is granted. To limit user access, require user assignment to an application. For more information, see [Methods for assigning users and groups](#). Administrators can also limit user access to applications by disabling all future user consent operations to any application.

For a broader overview, including how to handle more complex scenarios, see [Use Microsoft Entra ID for application access management](#).

Next steps

- Configure the admin consent workflow
 - Configure how users consent to applications
-

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Protect against consent phishing

Article • 01/08/2025

Productivity is no longer confined to private networks, and work is shifting dramatically toward cloud services. While cloud applications enable employees to be productive remotely, attackers can also use application-based attacks to gain access to valuable organization data. You might be familiar with attacks focused on users, such as email phishing or credential compromise. Consent phishing is another threat vector to be aware of.

This article explores what consent phishing is, what Microsoft does to protect an organization, and what steps organizations can take to stay safe.

What is consent phishing?

Consent phishing attacks trick users into granting permissions to malicious cloud applications. These malicious applications can then gain access to legitimate cloud services and data of users. Unlike credential compromise, *threat actors* who perform consent phishing target users who can grant access to their personal or organizational data directly. The consent screen displays all permissions the application receives. Because a legitimate provider (such as the Microsoft identity platform) hosts the application, unsuspecting users accept the terms. This action grants a malicious application the requested permissions to the data. The following image shows an example of an OAuth app that is requesting access to a wide variety of permissions.



testadmin@fourthcoffeetest.onmicrosoft.com

Permissions requested



Best Practices Demo
microsoftidentity.dev

This application is not published by Microsoft or your organization.

This app would like to:

- Read all groups
- Maintain access to data you have given it access to
- View your basic profile
- Consent on behalf of your organization

Accepting these permissions means that you allow this app to use your data as specified in their [terms of service](#) and [privacy statement](#). You can change these permissions at <https://myapps.microsoft.com>. [Show details](#)

Does this app look suspicious? [Report it here](#)

[Cancel](#)

[Accept](#)

Mitigating consent phishing attacks

Administrators, users, or Microsoft security researchers might flag OAuth applications that appear to behave suspiciously. Microsoft reviews a flagged application to determine whether it violates the terms of service. If a violation is confirmed, Microsoft Entra ID disables the application and prevents further use across all Microsoft services.

When Microsoft Entra ID disables an OAuth application, the following actions occur:

- The malicious application and related service principals are placed into a fully disabled state. Any new token requests or requests for refresh tokens are denied, but existing access tokens are still valid until their expiration.
- These applications show `DisabledDueToViolationOfServicesAgreement` on the `disabledByMicrosoftStatus` property on the related [application](#) and [service principal](#) resource types in Microsoft Graph. To prevent them from being instantiated in your organization again in the future, you can't delete these objects.
- An email is sent to a Privileged Role Administrator when a user in an organization consented to an application before it was disabled. The email specifies the action taken and recommended steps they can do to investigate and improve their security posture.

Recommended response and remediation

If a Microsoft disabled application impacts the organization, the organization should take the following steps to keep the environment secure:

1. Investigate the application activity for the disabled application, including:
 - The delegated permissions or application permissions requested by the application.
 - The Microsoft Entra audit logs for activity by the application and sign-in activity for users authorized to use the application.
2. Review and use the [guidance for defending against illicit consent grants](#). The guidance includes auditing permissions and consent for disabled and suspicious applications found during review.
3. Implement best practices for hardening against consent phishing, described in the following section.

Best practices for hardening against consent phishing attacks

Administrators should be in control of application use by providing the right insights and capabilities to control how applications are allowed and used within organizations. While attackers never rest, there are steps organizations can take to improve the security posture. Some best practices to follow include:

- Educate your organization on how our permissions and consent framework works:
 - Understand the data and the permissions an application is asking for and understand how [permissions and consent](#) works within the platform.
 - Make sure that administrators know how to [manage and evaluate consent requests](#).
 - Routinely [audit applications and consented permissions](#) in the organization to make sure that applications are accessing only the data they need and are adhering to the principles of least privilege.
- Know how to spot and block common consent phishing tactics:
 - Check for poor spelling and grammar. If an email message or the consent screen of the application has spelling and grammatical errors, it's likely a suspicious application. In that case, report it directly on the [consent prompt](#) with the [Report it here](#) link and Microsoft investigates if it's a malicious application and disable it, if confirmed.

- Don't rely on application names and domain URLs as a source of authenticity. Attackers like to spoof application names and domains that make it appear to come from a legitimate service or company to drive consent to a malicious application. Instead, validate the source of the domain URL and use applications from [verified publishers](#) when possible.
 - Block [consent phishing emails with Microsoft Defender for Office 365](#) by protecting against phishing campaigns where an attacker is impersonating a known user in the organization.
 - Configure Microsoft Defender for Cloud Apps policies to help manage abnormal application activity in the organization. For example, [activity policies](#), [anomaly detection](#), and [OAuth app policies](#).
 - Investigate and hunt for consent phishing attacks by following the guidance on [advanced hunting with Microsoft 365 Defender](#).
- Allow access to trusted applications that meet certain criteria and protect against those applications that don't:
 - [Configure user consent settings](#) to allow users to only consent to applications that meet certain criteria. Such applications include applications developed by your organization or from verified publishers and only for low risk permissions you select.
 - Use applications that are publisher verified. [Publisher verification](#) helps administrators and users understand the authenticity of application developers through a Microsoft supported vetting process. Even if an application does have a verified publisher, it's still important to review the consent prompt to understand and evaluate the request. For example, reviewing the permissions being requested to ensure they align with the scenario the app is requesting them to enable, other app and publisher details on the consent prompt, and so on.
 - Create proactive [application governance](#) policies to monitor third-party application behavior on the Microsoft 365 platform to address common suspicious application behaviors.
 - With [Microsoft Security Copilot](#), you can use natural language prompts to get insights from your Microsoft Entra data. This helps you identify and understand risks related to applications or workload identities. Learn more about how to [Assess application risks using Microsoft Security Copilot in Microsoft Entra](#).

Next steps

- [Application consent grant investigation](#)
- [Managing access to applications](#)
- [Restrict user consent operations in Microsoft Entra ID](#)

- Compromised and malicious applications investigation
-

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Understand the stages of migrating application authentication from AD FS to Microsoft Entra ID

Article • 04/29/2025

Microsoft Entra ID offers a universal identity platform that provides your people, partners, and customers a single identity to access applications and collaborate from any platform and device. Microsoft Entra ID has a full suite of identity management capabilities. Standardizing your application authentication and authorization to Microsoft Entra ID provides these benefits.

Types of apps to migrate

Your applications might use modern or legacy protocols for authentication. When you plan your migration to Microsoft Entra ID, consider migrating the apps that use modern authentication protocols (such as SAML and OpenID Connect) first.

These apps can be reconfigured to authenticate with Microsoft Entra ID either via a built-in connector from the Azure App Gallery. They can also be reconfigured by registering the custom application in Microsoft Entra ID.

Apps that use older protocols can be integrated using [Application Proxy](#) or any of our [Secure Hybrid Access \(SHA\) partners](#).

For more information, see:

- [Using Microsoft Entra application proxy to publish on-premises apps for remote users](#).
- [What is application management?](#)
- [AD FS application activity report to migrate applications to Microsoft Entra ID](#).
- [Monitor AD FS using Microsoft Entra Connect Health](#).

The migration process

During the process of moving your app authentication to Microsoft Entra ID, test your apps and configuration. We recommend that you continue to use existing test environments for migration testing before you move to the production environment. If a test environment isn't currently available, you can set one up using [Azure App Service](#) or [Azure Virtual Machines](#), depending on the architecture of the application.

You might choose to set up a separate test Microsoft Entra tenant on which to develop your app configurations.

Your migration process might look like this:

Stage 1 – Current state: The production app authenticates with AD FS



Stage 2 – (Optional) Point a test instance of the app to the test Microsoft Entra tenant

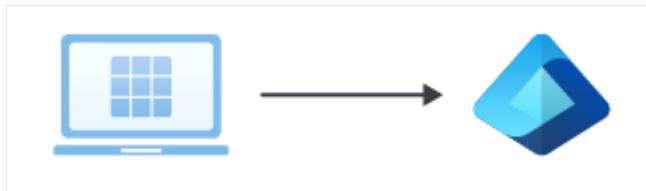
Update the configuration to point your test instance of the app to a test Microsoft Entra tenant, and make any required changes. The app can be tested with users in the test Microsoft Entra tenant. During the development process, you can use tools such as [Fiddler](#) to compare and verify requests and responses.

If it isn't feasible to set up a separate test tenant, skip this stage and point a test instance of the app to your production Microsoft Entra tenant as described in Stage 3 below.



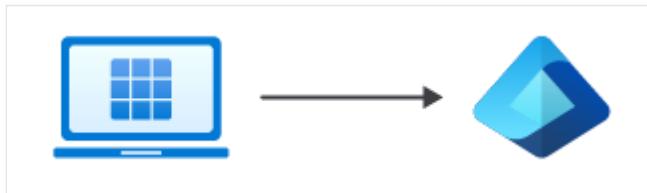
Stage 3 – Point a test instance of the app to the production Microsoft Entra tenant

Update the configuration to point your test instance of the app to your production Microsoft Entra tenant. You can now test with users in your production tenant. If necessary, review the section of this article on transitioning users.



Stage 4 – Point the production app to the production Microsoft Entra tenant

Update the configuration of your production app to point to your production Microsoft Entra tenant.



Apps that authenticate with AD FS can use Active Directory groups for permissions. Use [Microsoft Entra Connect Sync](#) to sync identity data between your on-premises environment and Microsoft Entra ID before you begin migration. Verify those groups and membership before migration so that you can grant access to the same users when the application is migrated.

Line of business apps

Your line-of-business apps are apps that your organization developed or apps that are a standard packaged product.

Line-of-business apps that use OAuth 2.0, OpenID Connect, or WS-Federation can be integrated with Microsoft Entra ID as [app registrations](#). Integrate custom apps that use SAML 2.0 or WS-Federation as [non-gallery applications](#) on the enterprise applications page in the [Microsoft Entra admin center](#).

Related content

[Configure SAML-based single sign-on.](#)

SAML-based single sign-on: Configuration and Limitations

Article • 08/20/2024

In this article, you learn how to configure an application for SAML-based single sign-on (SSO) with Microsoft Entra ID. It focuses on configuring SAML SSO for apps that are migrated from Active Directory Federation Services (ADFS) to Microsoft Entra ID.

The concepts covered include mapping users to specific application roles based on rules, and limitations to keep in mind when mapping attributes. It also covers SAML signing certificates, SAML token encryption, SAML request signature verification, and custom claims providers.

Apps that use SAML 2.0 for authentication can be configured for [SAML-based single sign-on \(SSO\)](#). With SAML-based SSO, you can map users to specific application roles based on rules that you define in your SAML claims.

To configure a SaaS application for SAML-based SSO, see [Quickstart: Set up SAML-based single sign-on](#).

The screenshot shows the Microsoft Entra admin center interface. The left sidebar navigation includes Home, Favorites, Identity, Overview, Users, Groups, Devices, Applications (selected), Enterprise applications, App registrations, Protection, Identity governance, External Identities, Show more, and Learn & support. The main content area is titled "Microsoft Entra SAML Toolkit 1 | SAML-based Sign-on" under "Enterprise Application". It displays two steps: Step 1, "Basic SAML Configuration", which lists fields: Identifier (Entity ID) [Required], Reply URL (Assertion Consumer Service URL) [Required], Sign on URL [Required], Relay State (Optional) [Optional], and Logout Url (Optional) [Optional]; Step 2, "Attributes & Claims", which lists attributes and their mappings: givenname to user.givenname, surname to user.surname, emailaddress to user.mail, name to user.userprincipalname, and Unique User Identifier to user.userprincipalname. A note at the top right says "Set up Single Sign-On with SAML" and provides a link to the configuration guide.

Many SaaS applications have an [application-specific tutorial](#) that steps you through the configuration for SAML-based SSO.

Some apps can be migrated easily. Apps with more complex requirements, such as custom claims, might require extra configuration in Microsoft Entra ID and/or [Microsoft Entra Connect Health](#). For information about supported claims mappings, see [How to: Customize claims emitted in tokens for a specific app in a tenant \(Preview\)](#).

Keep in mind the following limitations when mapping attributes:

- Not all attributes that can be issued in AD FS show up in Microsoft Entra ID as attributes to emit to SAML tokens, even if those attributes are synced. When you edit the attribute, the Value dropdown list shows you the different attributes that are available in Microsoft Entra ID. Check [Microsoft Entra Connect Sync articles](#) configuration to ensure that a required attribute—for example, `samAccountName`—is synced to Microsoft Entra ID. You can use the extension attributes to emit any claim that isn't part of the standard user schema in Microsoft Entra ID.
- In the most common scenarios, only the `NameID` claim and other common user identifier claims are required for an app. To determine if any extra claims are required, examine what claims you're issuing from AD FS.
- Not all claims can be issued, as some claims are protected in Microsoft Entra ID.
- The ability to use encrypted SAML tokens is now in preview. See [How to: customize claims issued in the SAML token for enterprise applications](#).

Software as a service (SaaS) apps

If your users sign in to SaaS apps such as Salesforce, ServiceNow, or Workday, and are integrated with AD FS, you're using federated sign-on for SaaS apps.

Most SaaS applications can be configured in Microsoft Entra ID. Microsoft has many preconfigured connections to SaaS apps in the [Microsoft Entra app gallery](#), which makes your transition easier. SAML 2.0 applications can be integrated with Microsoft Entra ID via the Microsoft Entra app gallery or as [non-gallery applications](#).

Apps that use OAuth 2.0 or OpenID Connect can be similarly integrated with Microsoft Entra ID as [app registrations](#). Apps that use legacy protocols can use [Microsoft Entra application proxy](#) to authenticate with Microsoft Entra ID.

SAML signing certificates for SSO

Signing certificates are an important part of any SSO deployment. Microsoft Entra ID creates the signing certificates to establish SAML-based federated SSO to your SaaS applications. Once you add either gallery or non-gallery applications, you configure the added application using the federated SSO option. See [Manage certificates for federated single sign-on in Microsoft Entra ID](#).

SAML token encryption

Both AD FS and Microsoft Entra ID provide token encryption—the ability to encrypt the SAML security assertions that go to applications. The assertions are encrypted with a public key, and decrypted by the receiving application with the matching private key. When you configure token encryption, you upload X.509 certificate files to provide the public keys.

For information about Microsoft Entra SAML token encryption and how to configure it, see [How to: Configure Microsoft Entra SAML token encryption](#).

 **Note**

Token encryption is a Microsoft Entra ID P1 or P2 feature. To learn more about Microsoft Entra editions, features, and pricing, see [Microsoft Entra pricing](#).

SAML request signature verification

This functionality validates the signature of signed authentication requests. An App Admin enables and disables the enforcement of signed requests and uploads the public keys that should be used to do the validation. For more information, see [How to enforce signed SAML authentication requests](#).

Custom claims providers (preview)

To migrate data from legacy systems such as ADFS, or data stores such as LDAP, your apps are dependent on certain data in the tokens. You can use custom claims providers to add claims into the token. For more information, see [Custom claims provider overview](#).

Apps and configurations that can be moved today

Apps that you can move easily today include SAML 2.0 apps that use the standard set of configuration elements and claims. These standard items are:

- User Principal Name
- Email address
- Given name
- Surname
- Alternate attribute as SAML **NameID**, including the Microsoft Entra ID mail attribute, mail prefix, employee ID, extension attributes 1-15, or on-premises **SamAccountName** attribute. For more information, see [Editing the NamelIdentifier claim](#).
- Custom claims.

The following require more configuration steps to migrate to Microsoft Entra ID:

- Custom authorization or multifactor authentication (MFA) rules in AD FS. You configure them using the [Microsoft Entra Conditional Access](#) feature.
- Apps with multiple Reply URL endpoints. You configure them in Microsoft Entra ID using PowerShell or the Microsoft Entra admin center interface.
- WS-Federation apps such as SharePoint apps that require SAML version 1.1 tokens. You can configure them manually using PowerShell. You can also add a preintegrated generic template for SharePoint and SAML 1.1 applications from the gallery. We support the SAML 2.0 protocol.
- Complex claims issuance transforms rules. For information about supported claims mappings, see:
 - [Claims mapping in Microsoft Entra ID](#).
 - [Customizing claims issued in the SAML token for enterprise applications in Microsoft Entra ID](#).

Apps and configurations not supported in Microsoft Entra today

Apps that require certain capabilities can't be migrated today.

Protocol capabilities

Apps that require the following protocol capabilities can't be migrated today:

- Support for the WS-Trust ActAs pattern
- SAML artifact resolution

Map app settings from AD FS to Microsoft Entra ID

Migration requires assessing how the application is configured on-premises, and then mapping that configuration to Microsoft Entra ID. AD FS and Microsoft Entra ID work similarly, so the concepts of configuring trust, sign-on and sign-out URLs, and identifiers apply in both cases. Document the AD FS configuration settings of your applications so that you can easily configure them in Microsoft Entra ID.

Map app configuration settings

The following table describes some of the most common mapping of settings between an AD FS Relying Party Trust to Microsoft Entra Enterprise Application:

- AD FS—Find the setting in the AD FS Relying Party Trust for the app. Right-click the relying party and select Properties.
- Microsoft Entra ID—The setting is configured within [Microsoft Entra admin center](#) in each application's SSO properties.

 Expand table

Configuration setting	AD FS	How to configure in Microsoft Entra ID	SAML Token
App sign-on URL	N/A The URL for the user to sign in to the app in a SAML flow initiated by a Service Provider (SP).	Open Basic SAML Configuration from SAML based sign-on	N/A
App reply URL	Select the Endpoints tab The URL of the app from the perspective of the identity provider (IdP). The IdP sends the user and token here after the user signs in to the IdP. Also known as SAML assertion consumer endpoint .	Open Basic SAML Configuration from SAML based sign-on	Destination element in the SAML token. Example value: <code>https://contoso.my.salesforce.com</code>

Configuration setting	AD FS	How to configure in Microsoft Entra ID	SAML Token
App sign-out URL The URL to which sign-out cleanup requests are sent when a user signs out from an app. The IdP sends the request to sign out the user from all other apps as well.	Select the Endpoints tab	Open Basic SAML Configuration from SAML based sign-on	N/A
App identifier The app identifier from the IdP's perspective. The sign-on URL value is often used for the identifier (but not always). Sometimes the app calls it the <i>entity ID</i> .	Select the Identifiers tab	Open Basic SAML Configuration from SAML based sign-on	Maps to the Audience element in the SAML token.
App federation metadata The location of the app's federation metadata. The IdP uses it to automatically update specific configuration settings, such as endpoints or encryption certificates.	Select the Monitoring tab	N/A. Microsoft Entra ID doesn't support consuming application federation metadata directly. You can manually import the federation metadata.	N/A
User Identifier/ Name ID Attribute that is used to uniquely indicate the user identity from Microsoft Entra ID or AD FS to your app. This attribute is typically either the UPN or the email address of the user.	Claim rules. In most cases, the claim rule issues a claim with a type that ends with the NameIdentifier .	You can find the identifier under the header User Attributes and Claims . By default, the UPN is applied	Maps to the NameID element in the SAML token.
Other claims Examples of other claim information that is commonly sent from the IdP to the app include first name, last name, email address, and group membership.	In AD FS, you can find this as other claim rules on the relying party.	You can find the identifier under the header User Attributes & Claims . Select View and edit all other user attributes .	N/A

Map Identity Provider (IdP) settings

Configure your applications to point to Microsoft Entra ID versus AD FS for SSO. Here, we're focusing on SaaS apps that use the SAML protocol. However, this concept extends to custom line-of-business apps as well.

Note

The configuration values for Microsoft Entra ID follows the pattern where your Azure Tenant ID replaces `{tenant-id}` and the Application ID replaces `{application-id}`. You find this information in the [Microsoft Entra admin center](#) under **Microsoft Entra ID > Properties**:

- Select Directory ID to see your Tenant ID.
- Select Application ID to see your Application ID.

At a high-level, map the following key SaaS apps configuration elements to Microsoft Entra ID.

 Expand table

Element	Configuration Value
Identity provider issuer	<code>https://sts.windows.net/{tenant-id}/</code>

Element	Configuration Value
Identity provider sign-in URL	<code>https://login.microsoftonline.com/{tenant-id}/saml2</code>
Identity provider sign-out URL	<code>https://login.microsoftonline.com/{tenant-id}/saml2</code>
Federation metadata location	<code>https://login.windows.net/{tenant-id}/federationmetadata/2007-06/federationmetadata.xml?appid={application-id}</code>

Map SSO settings for SaaS apps

SaaS apps need to know where to send authentication requests and how to validate the received tokens. The following table describes the elements to configure SSO settings in the app, and their values or locations within AD FS and Microsoft Entra ID.

[Expand table](#)

Configuration setting	AD FS	How to configure in Microsoft Entra ID
IdP Sign-on URL	The AD FS sign-on URL is the AD FS federation service name followed by <code>/adfs/ls/</code> .	Replace <code>{tenant-id}</code> with your tenant ID. For apps that use the SAML-P protocol: <code>https://login.microsoftonline.com/{tenant-id}/saml2</code>
Sign-on URL of the IdP from the app's perspective (where the user is redirected for sign-in).	For example: <code>https://fs.contoso.com/adfs/ls/</code>	For apps that use the WS-Federation protocol: <code>https://login.microsoftonline.com/{tenant-id}/wsfed</code>
IdP sign-out URL	The sign-out URL is either the same as the sign-on URL, or the same URL with <code>wa=wsignin1.0</code> appended. For example: <code>https://fs.contoso.com/adfs/ls/?wa=wsignin1.0</code>	Replace <code>{tenant-id}</code> with your tenant ID. For apps that use the SAML-P protocol: <code>https://login.microsoftonline.com/{tenant-id}/saml2</code>
Sign-out URL of the IdP from the app's perspective (where the user is redirected when they choose to sign out of the app).		For apps that use the WS-Federation protocol: <code>https://login.microsoftonline.com/common/wsfed?wa=wsignout1.0</code>
Token signing certificate	Find the AD FS token signing certificate in AD FS Management under Certificates.	Find it in the Microsoft Entra admin center in the application's Single sign-on properties under the header SAML Signing Certificate . There, you can download the certificate for upload to the app. If the application has more than one certificate, you can find all certificates in the federation metadata XML file.
The IdP uses the private key of the certificate to sign issued tokens. It verifies that the token came from the same IdP that the app is configured to trust.		
Identifier/ "issuer"	The identifier for AD FS is usually the federation service identifier in AD FS Management under Service > Edit Federation Service Properties . For	Replace <code>{tenant-id}</code> with your tenant ID. <code>https://sts.windows.net/{tenant-id}/</code>

Configuration AD FS setting	How to configure in Microsoft Entra ID
<p>Identifier of the IdP from the app's perspective (sometimes called the "issuer ID").</p> <p>In the SAML token, the value appears as the Issuer element.</p>	<p>example: <code>http://fs.contoso.com/adfs/services/trust</code></p>
<p>IdP federation metadata Find the AD FS federation metadata URL in AD FS Management under Service > Endpoints > Metadata > Type: Federation Metadata. For example: <code>https://fs.contoso.com/FederationMetadata/2007-06/FederationMetadata.xml</code></p> <p>Location of the IdP's publicly available federation metadata. (Some apps use federation metadata as an alternative to the administrator configuring URLs, identifier, and token signing certificate individually.)</p>	<p>The corresponding value for Microsoft Entra ID follows the pattern <code>https://login.microsoftonline.com/{TenantDomainName}/FederationMetadata/2007-06/FederationMetadata.xml</code>. Replace {TenantDomainName} with your tenant's name in the format <code>contoso.onmicrosoft.com</code>.</p> <p>For more information, see Federation metadata.</p>

Next steps

- Represent AD FS security policies in Microsoft Entra ID.

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Represent AD FS security policies in Microsoft Entra ID: Mappings and examples

Article • 05/06/2024

In this article, you'll learn how to map authorization and multifactor authentication rules from AD FS to Microsoft Entra ID when moving your app authentication. Find out how to meet your app owner's security requirements while making the app migration process easier with mappings for each rule.

When moving your app authentication to Microsoft Entra ID, create mappings from existing security policies to their equivalent or alternative variants available in Microsoft Entra ID. Ensuring that these mappings can be done while meeting security standards required by your app owners makes the rest of the app migration easier.

For each rule example, we show what the rule looks like in AD FS, the AD FS rule language equivalent code, and how this maps to Microsoft Entra ID.

Map authorization rules

The following are examples of various types of authorization rules in AD FS, and how you map them to Microsoft Entra ID.

Example 1: Permit access to all users

Permit Access to All Users in AD FS:

Edit Access Control Policy for My SaaS application

X

Issuance Authorization Rules

The following authorization rules specify the users that will be permitted access to the relying party. When the list does not contain a rule, all users will be denied access.

Order	Rule Name	Issued Claims
1	Permit Access to All Users	Permit



Add Rule...

Edit Rule...

Remove Rule...

[Use access control policy](#)

OK

Cancel

Apply

This maps to Microsoft Entra ID in one of the following ways:

1. Set Assignment required to No.

The screenshot shows the Microsoft Entra admin center interface. On the left, there's a navigation sidebar with various sections like Home, Favorites, Identity, Applications, Protection, and Learn & support. The main area is titled 'My SaaS App in Microsoft Entra | Properties' for an 'Enterprise Application'. It includes tabs for Manage (Properties, Owners, Roles and administrators, Users and groups, Single sign-on, Provisioning, Application proxy, Self-service, Custom security attributes), Security (Conditional Access, Permissions, Token encryption), and Activity (Sign-in logs, Usage & insights, Audit logs, Provisioning logs). The 'Properties' tab is selected. At the top right, there are Save, Discard, Delete, and Got feedback? buttons. Below them, a note says 'View and manage application settings for your organization. Editing properties like display information, user sign-in settings, and user visibility settings requires Global Administrator, Cloud Application Administrator, Application Administrator roles. [Learn more](#)'. A section for managing users and groups follows, with a note: 'If this application resides in your tenant, you can manage additional properties on the [application registration](#)'. There are fields for Name (set to 'My SaaS App in Microsoft Entra'), Homepage URL (set to 'https://account.activedirectory.windowsazure.com:444/applications/de...'), Logo (a green square with 'MS'), User access URL (set to 'https://launcher.myapps.microsoft.com/api/signin/a509d6aa-146a-4e1...'), Application ID (set to 'a509d6aa-146a-4e1e-bad2-089ef9b6581a'), Object ID (set to 'cc077aff-592f-4feb-9326-814ffd13ed3d'), Terms of Service Url (set to 'Publisher did not provide this information'), Privacy Statement Url (set to 'Publisher did not provide this information'), and Reply URL (set to 'Publisher did not provide this information'). A red box highlights the 'Assignment required?' switch, which is set to 'No'. Below it, there's a note: 'Visible to users? Yes No'. At the bottom right of the main area, there are 'Save', 'Discard', 'Delete', and 'Got feedback?' buttons.

! Note

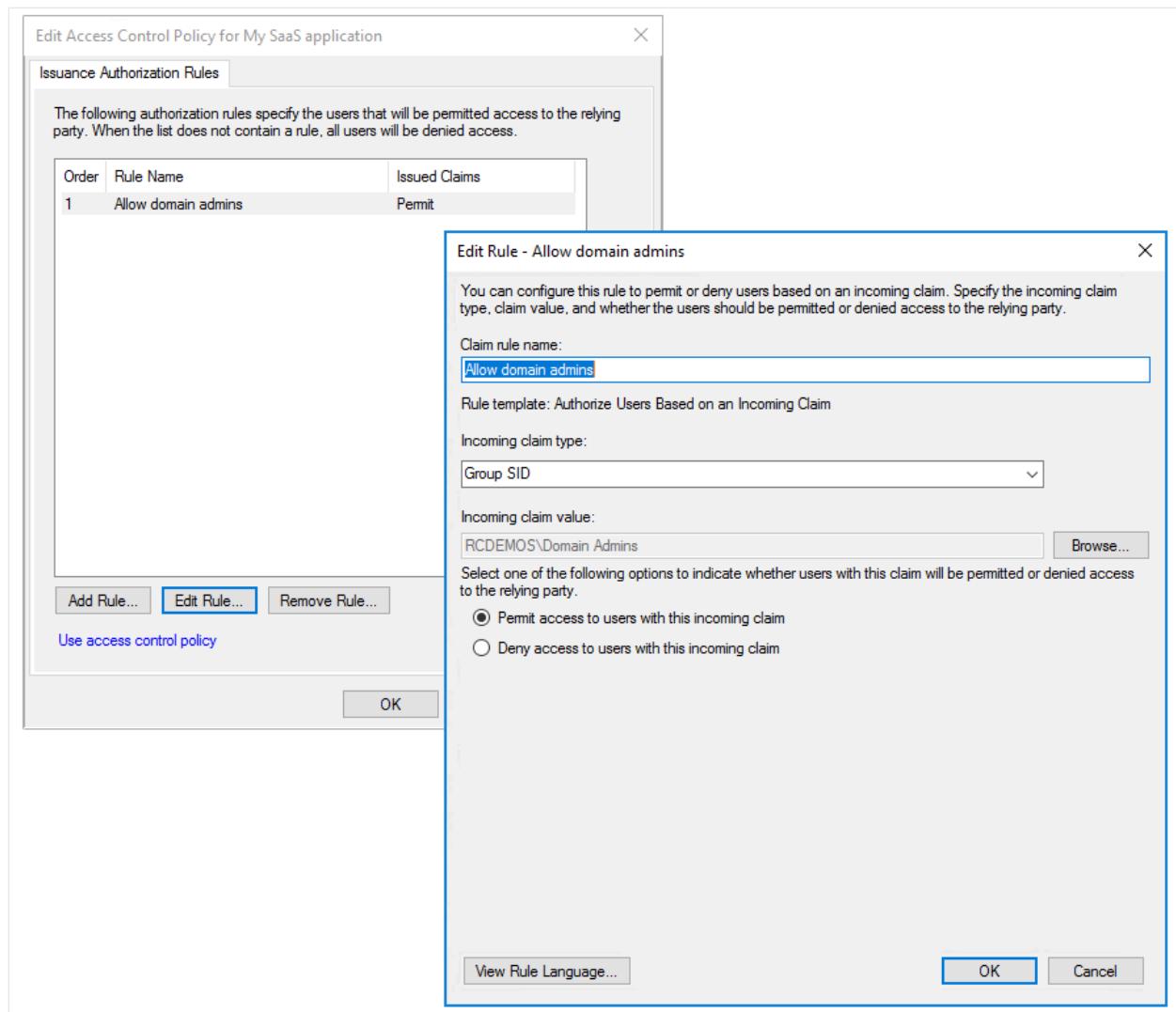
Setting **Assignment required** to **Yes** requires that users are assigned to the application to gain access. When set to **No**, all users have access. This switch doesn't control what users see in the **My Apps** experience.

2. In the **Users and groups** tab, assign your application to the **All Users** automatic group. You must [enable Dynamic Groups](#) in your Microsoft Entra tenant for the default **All Users** group to be available.

The screenshot shows the 'Users and groups' tab for the 'My SaaS App in Microsoft Entra' application. The left sidebar is identical to the previous screenshot. The main area is titled 'My SaaS App in Microsoft Entra | Users and groups'. It includes buttons for Add user/group, Edit assignment, Remove, Update credentials, Columns, and Got feedback?. A note says: 'The application will appear for assigned users within My Apps. Set visible to users? to no in properties to prevent this.' Below is a table titled 'Assign users and groups to app-roles for your application here. To create new app-roles for this application, use the application registration.' The table has columns for Display Name, Object Type, and Role assigned. One row is shown: 'AU' (checkbox checked), 'All Users' (display name), 'Group' (object type), and 'Default Access' (role assigned). A red box highlights this row. At the bottom right of the main area, there are 'Save', 'Discard', 'Delete', and 'Got feedback?' buttons.

Example 2: Allow a group explicitly

Explicit group authorization in AD FS:



To map this rule to Microsoft Entra ID:

1. In the [Microsoft Entra admin center](#), create a user group that corresponds to the group of users from AD FS.
2. Assign app permissions to the group:

Users and groups

X

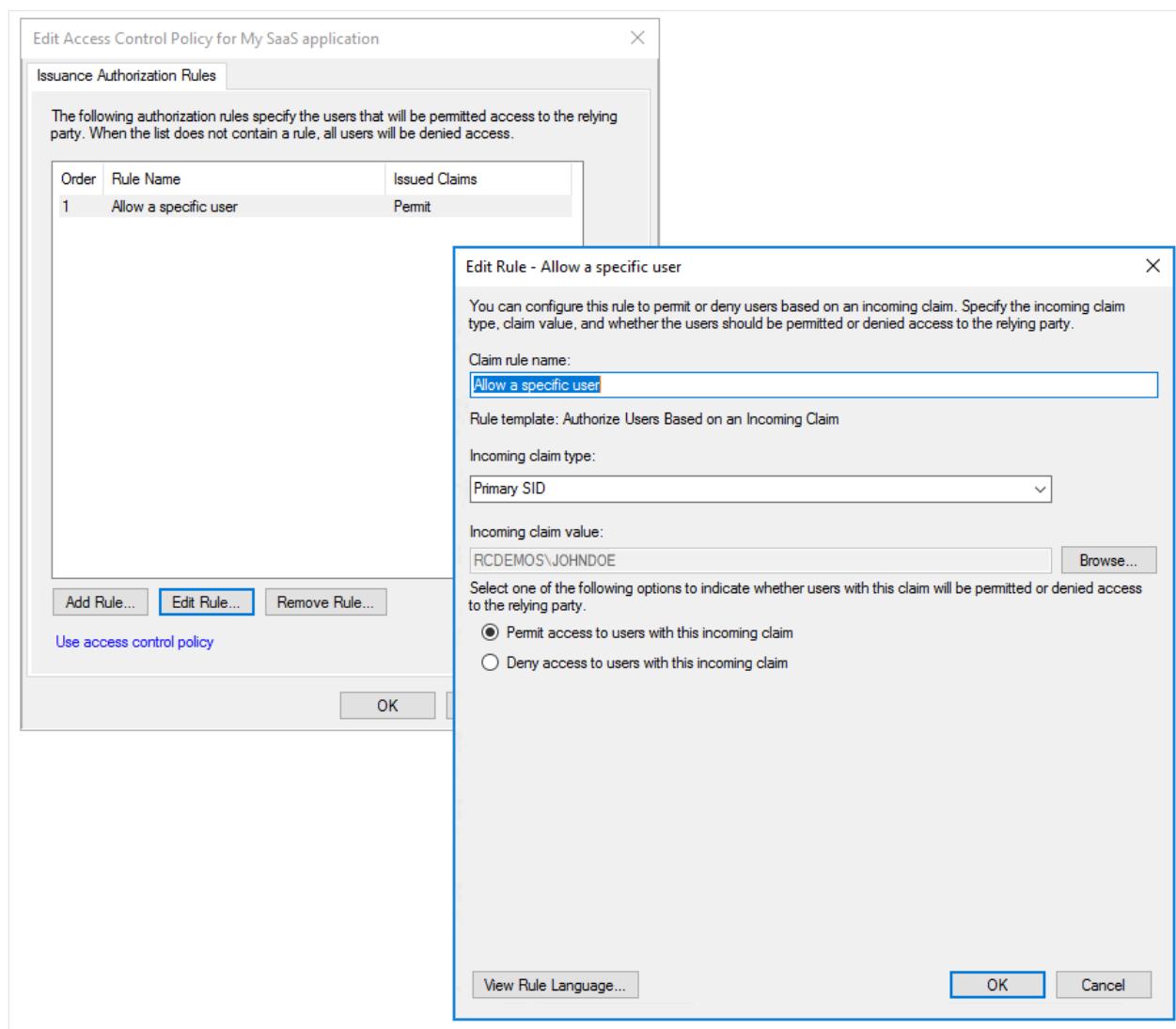
 Domain Admins X

DA

Domain Admins

Example 3: Authorize a specific user

Explicit user authorization in AD FS:



To map this rule to Microsoft Entra ID:

- In the [Microsoft Entra admin center](#), add a user to the app through the Add Assignment tab of the app as shown below:

The screenshot shows the Microsoft Entra admin center interface. On the left, there's a navigation sidebar with sections like Home, Favorites, Identity (selected), Overview, Users, Groups, Devices, Applications, Enterprise applications, App registrations, Protection, and Identity governance. The main area is titled 'My SaaS App in Microsoft Entra | Users and groups' and shows a list of users assigned to the app. The list includes 'DA - Domain Admins' (Group, Default Access) and 'JD - John Doe' (User, Default Access). A red box highlights the user 'JD - John Doe'.

Map multifactor authentication rules

An on-premises deployment of [Multifactor Authentication \(MFA\)](#) and AD FS still works after the migration because you're federated with AD FS. However, consider migrating to Azure's built-in MFA capabilities that are tied into Microsoft Entra Conditional Access policies.

The following are examples of types of MFA rules in AD FS, and how you can map them to Microsoft Entra ID based on different conditions.

MFA rule settings in AD FS:

[Primary](#) [Multi-factor](#)

Configure multi-factor authentication (MFA) settings.

Users/Groups

MFA is required for the following users and groups:

[Add...](#)
[Remove](#)

Devices

MFA is required for the following devices:

Unregistered devices
 Registered devices

Locations

MFA is required when accessing applications from the following locations:

Extranet
 Intranet

Select additional authentication methods. You must select at least one of the following methods to enable MFA:

Certificate Authentication

[What is multi-factor authentication?](#)

[OK](#) [Cancel](#) [Apply](#)

Example 1: Enforce MFA based on users/groups

The users/groups selector is a rule that allows you to enforce MFA on a per-group (Group SID) or per-user (Primary SID) basis. Apart from the users/groups assignments, all other checkboxes in the AD FS MFA configuration UI function as extra rules that are evaluated after the users/groups rule is enforced.

[Common Conditional Access policy: Require MFA for all users](#)

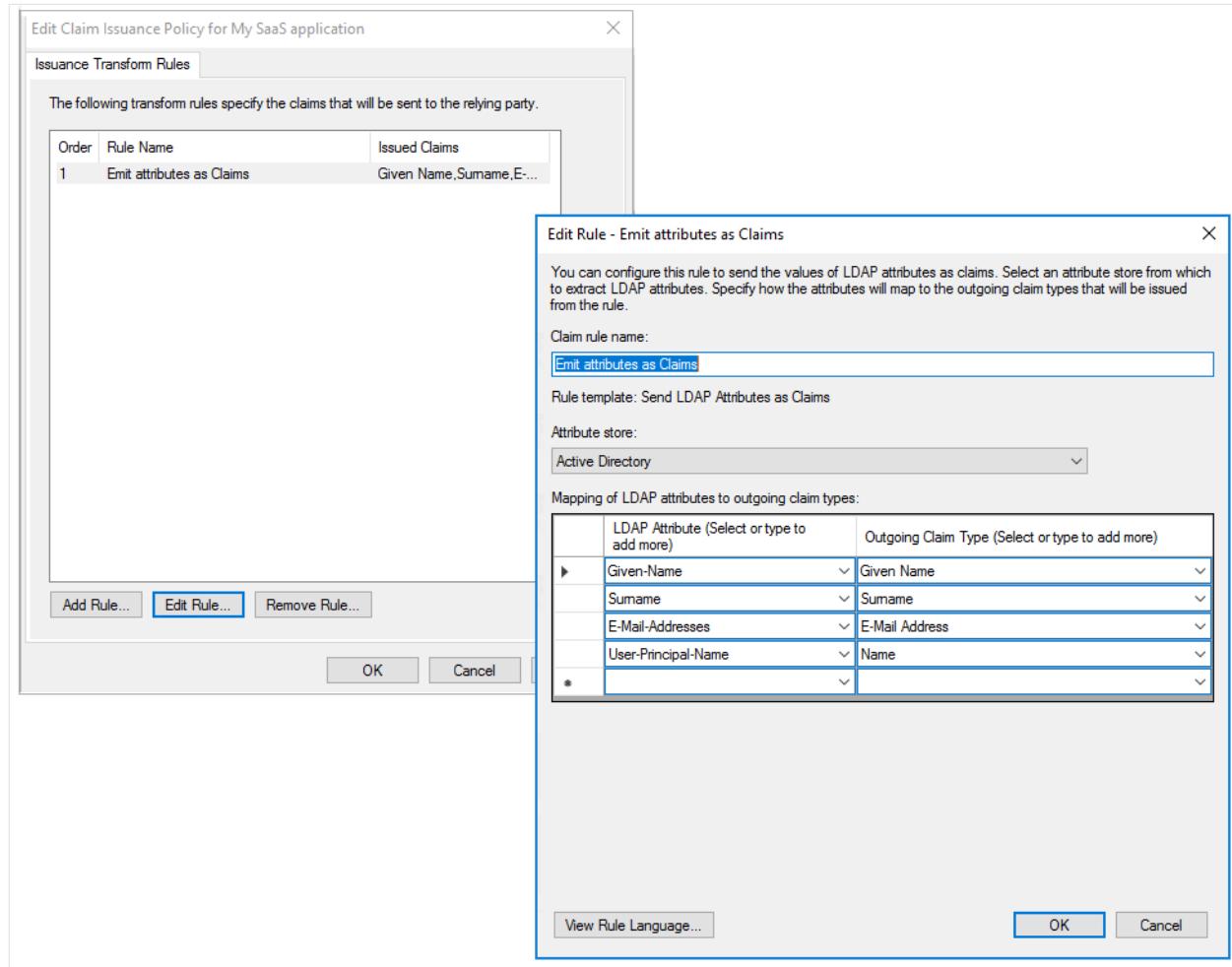
Example 2: Enforce MFA for unregistered devices

Specify MFA rules for unregistered devices in Microsoft Entra:

Common Conditional Access policy: Require a compliant device, Microsoft Entra hybrid joined device, or multifactor authentication for all users

Map Emit attributes as Claims rule

Emit attributes as Claims rule in AD FS:



To map the rule to Microsoft Entra ID:

1. In the [Microsoft Entra admin center](#), select **Enterprise Applications** and then **Single sign-on** to view the SAML-based sign-on configuration:

My SaaS App in Microsoft Entra | SAML-based Sign-on

Basic SAML Configuration

- Identifier (Entity ID): http://adapplicationregistry.onmicrosoft.com/customapps/so/primary
- Reply URL (Assertion Consumer Service URL): https://reply1.mysaasapp.com
- Sign on URL: Optional
- Relay State (Optional): Optional
- Logout Url (Optional): Optional

Attributes & Claims

Attribute Name	Description
givenname	user.givenname
surname	user.surname
emailaddress	user.mail
name	user.userprincipalname
Unique User Identifier	user.userprincipalname

SAML Certificates

Token signing certificate

Attribute	Description
Status	Active
Thumbprint	7E997565DF2F173BCD9374C5E1833680963B881D3
Expiration	12/19/2026, 4:11:47 PM
Notification Email	admin@MSdx427733.onmicrosoft.com

2. Select **Edit** (highlighted) to modify the attributes:

Attributes & Claims

Required claim

Claim name	Type	Value
Unique User Identifier (Name ID)	SAML	user.userprincipalname...

Additional claims

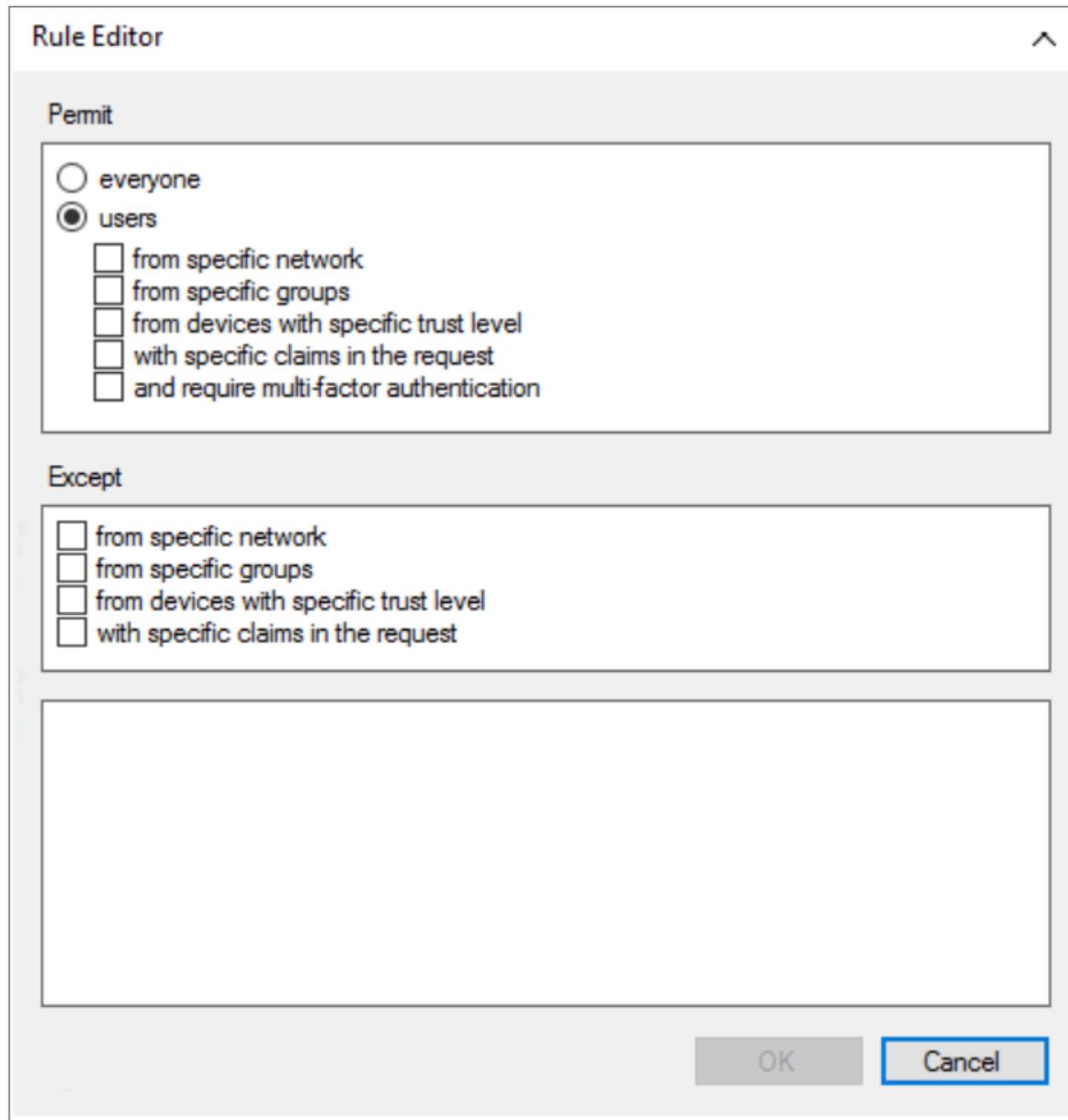
Claim name	Type	Value
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailadd...	SAML	user.mail
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	SAML	user.givenname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	SAML	user.userprincipalname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	SAML	user.surname

Map built-in access control policies

Built-in access control policies in AD FS 2016:

Access Control Policies				
Name	Built-in	Parameters	Usage	Modified
Permit everyone and require MFA, allow automatic device registration	Yes	No	Not in use	2/22/2018 7:11 PM
Permit everyone and require MFA	Yes	No	Not in use	2/22/2018 7:11 PM
Permit everyone and require MFA for specific group	Yes	Yes	Not in use	2/22/2018 7:11 PM
Permit everyone and require MFA from extranet access	Yes	No	Not in use	2/22/2018 7:11 PM
Permit everyone	Yes	No	In use (1)	2/22/2018 7:11 PM
Permit specific group	Yes	Yes	Not in use	2/22/2018 7:11 PM
Permit everyone for intranet access	Yes	No	Not in use	2/22/2018 7:11 PM
Permit everyone and require MFA from unauthenticated devices	Yes	No	Not in use	2/22/2018 7:11 PM

To implement built-in policies in Microsoft Entra ID, use a [new Conditional Access policy](#) and configure the access controls, or use the custom policy designer in AD FS 2016 to configure access control policies. The Rule Editor has an exhaustive list of Permit and Except options that can help you make all kinds of permutations.



In this table, we've listed some useful Permit and Except options and how they map to Microsoft Entra ID.

[\[+\] Expand table](#)

Option	How to configure Permit option in Microsoft Entra ID?	How to configure Except option in Microsoft Entra ID?
From specific network	Maps to Named Location in Microsoft Entra	Use the Exclude option for trusted locations

Option	How to configure Permit option in Microsoft Entra ID?	How to configure Except option in Microsoft Entra ID?
From specific groups	Set a User/Groups Assignment	Use the Exclude option in Users and Groups
From Devices with Specific Trust Level	Set this from the Device State control under Assignments -> Conditions	Use the Exclude option under Device State Condition and Include All devices
With Specific Claims in the Request	This setting can't be migrated	This setting can't be migrated

Here's an example of how to configure the Exclude option for trusted locations in the Microsoft Entra admin center:

Transition users from AD FS to Microsoft Entra ID

Sync AD FS groups in Microsoft Entra ID

When you map authorization rules, apps that authenticate with AD FS may use Active Directory groups for permissions. In such a case, use [Microsoft Entra Connect](#) to sync

these groups with Microsoft Entra ID before migrating the applications. Make sure that you verify those groups and membership before migration so that you can grant access to the same users when the application is migrated.

For more information, see [Prerequisites for using Group attributes synchronized from Active Directory](#).

Set up user self-provisioning

Some SaaS applications support the ability to Just-in-Time (JIT) provision users when they first sign in to the application. In Microsoft Entra ID, app provisioning refers to automatically creating user identities and roles in the cloud ([SaaS ↗](#)) applications that users need to access. Users that are migrated already have an account in the SaaS application. Any new users added after the migration need to be provisioned. Test [SaaS app provisioning](#) once the application is migrated.

Sync external users in Microsoft Entra ID

Your existing external users can be set up in these two ways in AD FS:

- **External users with a local account within your organization**—You continue to use these accounts in the same way that your internal user accounts work. These external user accounts have a principle name within your organization, although the account's email may point externally.

As you progress with your migration, you can take advantage of the benefits that [Microsoft Entra B2B](#) offers by migrating these users to use their own corporate identity when such an identity is available. This streamlines the process of signing in for those users, as they're often signed in with their own corporate sign-in. Your organization's administration is easier as well, by not having to manage accounts for external users.

- **Federated external Identities**—If you're currently federating with an external organization, you have a few approaches to take:
 - [Add Microsoft Entra B2B collaboration users in the Microsoft Entra admin center](#). You can proactively send B2B collaboration invitations from the Microsoft Entra administrative portal to the partner organization for individual members to continue using the apps and assets they're used to.
 - [Create a self-service B2B sign-up workflow](#) that generates a request for individual users at your partner organization using the B2B invitation API.

No matter how your existing external users are configured, they likely have permissions that are associated with their account, either in group membership or specific

permissions. Evaluate whether these permissions need to be migrated or cleaned up.

Accounts within your organization that represent an external user need to be disabled once the user has been migrated to an external identity. The migration process should be discussed with your business partners, as there may be an interruption in their ability to connect to your resources.

Next steps

- Read [Migrating application authentication to Microsoft Entra ID](#).
- Set up [Conditional Access](#) and [MFA](#).
- Try a step-wise code sample:[AD FS to Microsoft Entra application migration playbook for developers](#).

Overview of AD FS application migration

Article • 06/10/2024

In this article, you learn about the capabilities of the AD FS application migration wizard and the migration status available on its dashboard. You also learn the various validation tests that the application migration generates for each of the applications that you want to migrate from AD FS to Microsoft Entra ID.

The AD FS application migration wizard lets you quickly identify which of your applications are capable of being migrated to Microsoft Entra ID. It assesses all AD FS applications for compatibility with Microsoft Entra ID. It also checks for any issues, gives guidance on preparing individual applications for migration, and configuring new Microsoft Entra application using one-click experience.

With the AD FS application migration wizard, you can:

- **Discover AD FS applications and scope your migration** - The AD FS application migration wizard lists all AD FS applications in your organization that have had an active user sign-in in the last 30 days. The report indicates an app's readiness for migration to Microsoft Entra ID. The report doesn't display Microsoft related relying parties in AD FS such as Office 365. For example, relying parties with name `urn:federation:MicrosoftOnline`.
- **Prioritize applications for migration** - Get the number of unique users signed in to the application in the past 1, 7, or 30 days to help determine the criticality or risk of migrating the application.
- **Run migration tests and fix issues** - The reporting service automatically runs tests to determine if an application is ready to migrate. The results are displayed in the AD FS application migration dashboard as a migration status. If the AD FS configuration isn't compatible with a Microsoft Entra configuration, you get specific guidance on how to address the configuration in Microsoft Entra ID.
- **Use one-click application configuration experience to configure new Microsoft Entra application** - This provides a guided experience to migrate on-premises relying party applications to cloud. The migration experience uses the relying party application's metadata that is directly imported from your on-premises environment. Also the experience provides a one-click configuration of SAML application on Microsoft Entra platform with some basic SAML settings, claims configurations, and groups assignments.

Note

AD FS application migration only supports SAML-based applications. It doesn't support applications that use protocols such as OpenID Connect, WS-Fed and OAuth 2.0. If you want to migrate applications that use these protocols, see [Use the AD FS application activity report](#) to identify the applications that you want to migrate. Once you've identified the apps you want to migrate, you can configure them manually in Microsoft Entra ID. For more information on how to get started on manual migration, see [Migrate and test your application](#).

AD FS application migration status

The Microsoft Entra Connect and Microsoft Entra Connect Health agents for AD FS reads your on-premises relying party application configurations and sign-in audit logs. This data about each AD FS application is analyzed to determine if it can be migrated as-is, or if additional review is needed. Based on the result of this analysis, migration status for the given application is determined.

Applications are categorized into following migration statuses:

- **Ready to migrate** means the AD FS application configuration is fully supported in Microsoft Entra ID and can be migrated as-is.
- **Needs review** means some of the application's settings can be migrated to Microsoft Entra ID, but you need to review the settings that can't be migrated as-is.
- **Additional steps required** means Microsoft Entra ID doesn't support some of the application's settings, so the application can't be migrated in its current state.

AD FS application migration validation tests

Application readiness is evaluated based on following predefined AD FS application configuration tests. The tests are run automatically and the results are displayed in the AD FS application migration dashboard as a **Migration status**. If the AD FS configuration isn't compatible with a Microsoft Entra configuration, you get specific guidance on how to address the configuration in Microsoft Entra ID.

AD FS application migration insights status updates

When the application is updated, internal agents sync the updates within a few minutes. However, AD FS migration insights jobs are responsible for evaluating the updates and compute a new migration status. Those jobs are scheduled to run every 24 hours, which means that the data will be computed only once in a day, at around 00:00 Coordinated Universal Time (UTC).

[+] [Expand table](#)

Result	Pass/Warning/Fail	Description
Test-ADFSRPAdditionalAuthenticationRules At least one nonmigratable rule was detected for AdditionalAuthentication.	Pass/Warning	<p>The relying party has rules to prompt for multifactor authentication. To move to Microsoft Entra ID, translate those rules into Conditional Access policies. If you're using an on-premises MFA, we recommend that you move to Microsoft Entra multifactor authentication.</p> <p>Learn more about Conditional Access.</p>
Test-ADFSRPAdditionalWSFedEndpoint Relying party has AdditionalWSFedEndpoint set to true.	Pass/Fail	<p>The relying party in AD FS allows multiple WS-Fed assertion endpoints. Currently, Microsoft Entra only supports one. If you have a scenario where this result is blocking migration, let us know.</p>
Test-ADFSRPAllowedAuthenticationClassReferences Relying Party has set AllowedAuthenticationClassReferences.	Pass/Fail	<p>This setting in AD FS lets you specify whether the application is configured to only allow certain authentication types. We recommend using Conditional Access to achieve this capability. If you have a scenario where this result is blocking migration, let us know.</p> <p>Learn more about Conditional Access.</p>

Result	Pass/Warning/Fail	Description
Test-ADFSRPAlwaysRequireAuthentication AlwaysRequireAuthenticationCheckResult	Pass/Fail	<p>This setting in AD FS lets you specify whether the application is configured to ignore SSO cookies and Always Prompt for Authentication. In Microsoft Entra ID, you can manage the authentication session using Conditional Access policies to achieve similar behavior. Learn more about configuring authentication session management with Conditional Access.</p>
Test-ADFSRPAutoUpdateEnabled Relying Party has AutoUpdateEnabled set to true	Pass/Warning	<p>This setting in AD FS lets you specify whether AD FS is configured to automatically update the application based on changes within the federation metadata. Microsoft Entra ID doesn't support this today but shouldn't block the migration of the application to Microsoft Entra ID.</p>
Test-ADFSRPCClaimsProviderName Relying Party has multiple ClaimsProviders enabled	Pass/Fail	<p>This setting in AD FS calls out the identity providers from which the relying party is accepting claims. In Microsoft Entra ID, you can enable external collaboration using Microsoft Entra B2B. Learn more about Microsoft Entra B2B.</p>
Test-ADFSRPDelegationAuthorizationRules	Pass/Fail	<p>The application has custom delegation authorization rules defined. This is a WS-Trust concept that Microsoft Entra ID supports by using</p>

Result	Pass/Warning/Fail	Description
		<p>modern authentication protocols, such as OpenID Connect and OAuth 2.0.</p> <p>Learn more about the Microsoft identity platform.</p>
Test-ADFSRPImpersonationAuthorizationRules	Pass/Warning	<p>The application has custom impersonation authorization rules defined. This is a WS-Trust concept that Microsoft Entra ID supports by using modern authentication protocols, such as OpenID Connect and OAuth 2.0.</p> <p>Learn more about the Microsoft identity platform.</p>
<p>Test-ADFSRPIssuanceAuthorizationRules</p> <p>At least one nonmigratable rule was detected for IssuanceAuthorization.</p>	Pass/Warning	<p>The application has custom issuance authorization rules defined in AD FS.</p> <p>Microsoft Entra ID supports this functionality with Microsoft Entra Conditional Access. Learn more about Conditional Access.</p> <p>You can also restrict access to an application by user or groups assigned to the application. Learn more about assigning users and groups to access applications.</p>
<p>Test-ADFSRPIssuanceTransformRules</p> <p>At least one nonmigratable rule was detected for IssuanceTransform.</p>	Pass/Warning	<p>The application has custom issuance transform rules defined in AD FS. Microsoft Entra ID supports customizing the claims issued in the token.</p> <p>To learn more, see Customize claims issued</p>

Result	Pass/Warning/Fail	Description
		in the SAML token for enterprise applications.
Test-ADFSRPMonitoringEnabled Relying Party has MonitoringEnabled set to true.	Pass/Warning	This setting in AD FS lets you specify whether AD FS is configured to automatically update the application based on changes within the federation metadata. Microsoft Entra doesn't support this today but shouldn't block the migration of the application to Microsoft Entra ID.
Test-ADFSRPNotBeforeSkew NotBeforeSkewCheckResult	Pass/Warning	AD FS allows a time skew based on the NotBefore and NotOnOrAfter times in the SAML token. Microsoft Entra ID automatically handles this by default.
Test-ADFSRPRRequestMFAFromClaimsProviders Relying Party has RequestMFAFromClaimsProviders set to true.	Pass/Warning	This setting in AD FS determines the behavior for MFA when the user comes from a different claims provider. In Microsoft Entra ID, you can enable external collaboration using Microsoft Entra B2B. Then, you can apply Conditional Access policies to protect guest access. Learn more about Microsoft Entra B2B and Conditional Access .
Test-ADFSRPSignedSamlRequestsRequired Relying Party has SignedSamlRequestsRequired set to true	Pass/Fail	The application is configured in AD FS to verify the signature in the SAML request. Microsoft Entra ID accepts a signed SAML request; however, it will not verify the signature. Microsoft Entra

Result	Pass/Warning/Fail	Description
		ID has different methods to protect against malicious calls. For example, Microsoft Entra ID uses the reply URLs configured in the application to validate the SAML request. Microsoft Entra ID will only send a token to reply URLs configured for the application. If you have a scenario where this result is blocking migration, let us know .
Test-ADFSRPTokenLifetimeTokenLifetimeCheckResult	Pass/Warning	The application is configured for a custom token lifetime. The AD FS default is one hour. Microsoft Entra ID supports this functionality using Conditional Access. To learn more, see Configure authentication session management with Conditional Access .
Relying Party is set to encrypt claims. This is supported by Microsoft Entra ID	Pass	With Microsoft Entra ID, you can encrypt the token sent to the application. To learn more, see Configure Microsoft Entra SAML token encryption .
EncryptedNameIdRequiredCheckResult	Pass/Fail	The application is configured to encrypt the nameID claim in the SAML token. With Microsoft Entra ID, you can encrypt the entire token sent to the application. Encryption of specific claims isn't yet supported. To learn more, see Configure Microsoft Entra SAML token encryption .

Next steps

- Use AD FS application migration wizard to migrate apps from AD FS to Microsoft Entra ID
-

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Plan application migration to Microsoft Entra ID

Article • 12/19/2023

In this article, you learn about the benefits of Microsoft Entra ID and how to plan for migrating your application authentication. This article gives an overview of the planning and exit criteria to help you plan your migration strategy and understand how Microsoft Entra authentication can support your organizational goals.

The process is broken into four phases. Each phase contains detailed planning and exit criteria that help you plan your migration strategy and understand how Microsoft Entra authentication supports your organizational goals.

<https://www.youtube-nocookie.com/embed/8Wmquuuualk>

Introduction

Today, your organization requires numerous applications for users to get work done. You likely continue to add, develop, or retire apps every day. Users access these applications from a vast range of corporate and personal devices, and locations. They open apps in many ways, including:

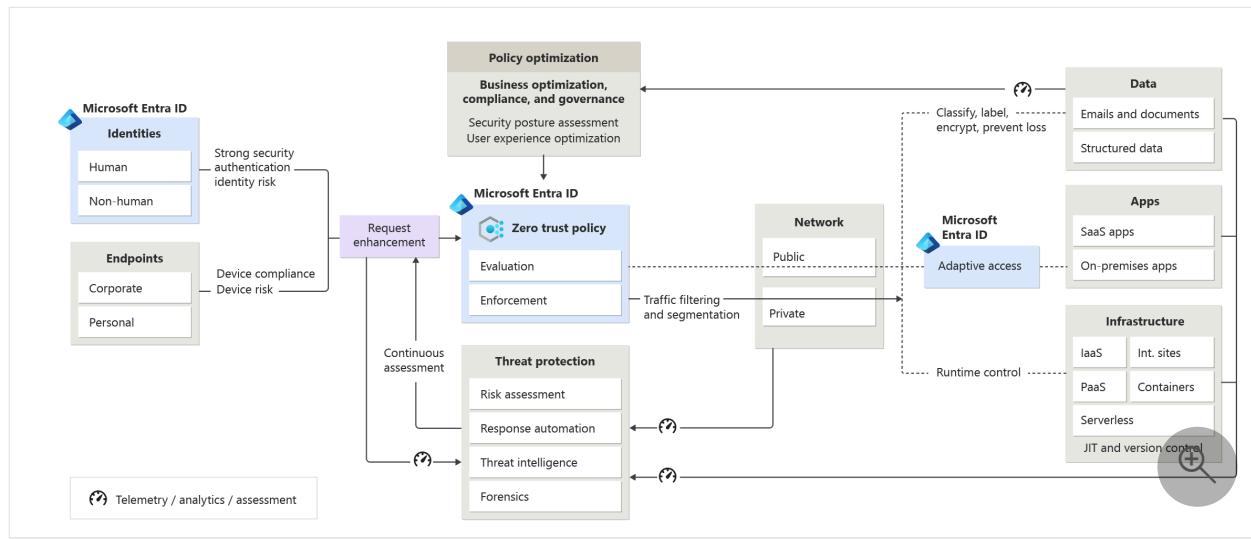
- Through a company homepage, or portal
- By bookmarking or adding favorites on their browsers
- Through a vendor's URL for software as a service (SaaS) apps
- Links pushed directly to user's desktops or mobile devices via a mobile device/application management (MDM/ MAM) solution

Your applications are likely using the following types of authentication:

- Security Assertion Markup Language (SAML) or OpenID Connect (OIDC) via an on-premises or cloud-hosted Identity and Access Management (IAM) solutions federation solution (such as Active Directory Federation Services (ADFS), Okta, or Ping)
- Kerberos or NTLM via Active Directory
- Header-based authentication via Ping Access

To ensure that the users can easily and securely access applications, your goal is to have a single set of access controls and policies across your on-premises and cloud environments.

Microsoft Entra ID offers a universal identity platform that provides your employees, partners, and customers a single identity to access the applications they want. The platform boosts collaboration from any platform and device.



Microsoft Entra ID has a [full suite of identity management capabilities](#). Standardizing your app authentication and authorization to Microsoft Entra ID gets you the benefits that these capabilities provide.

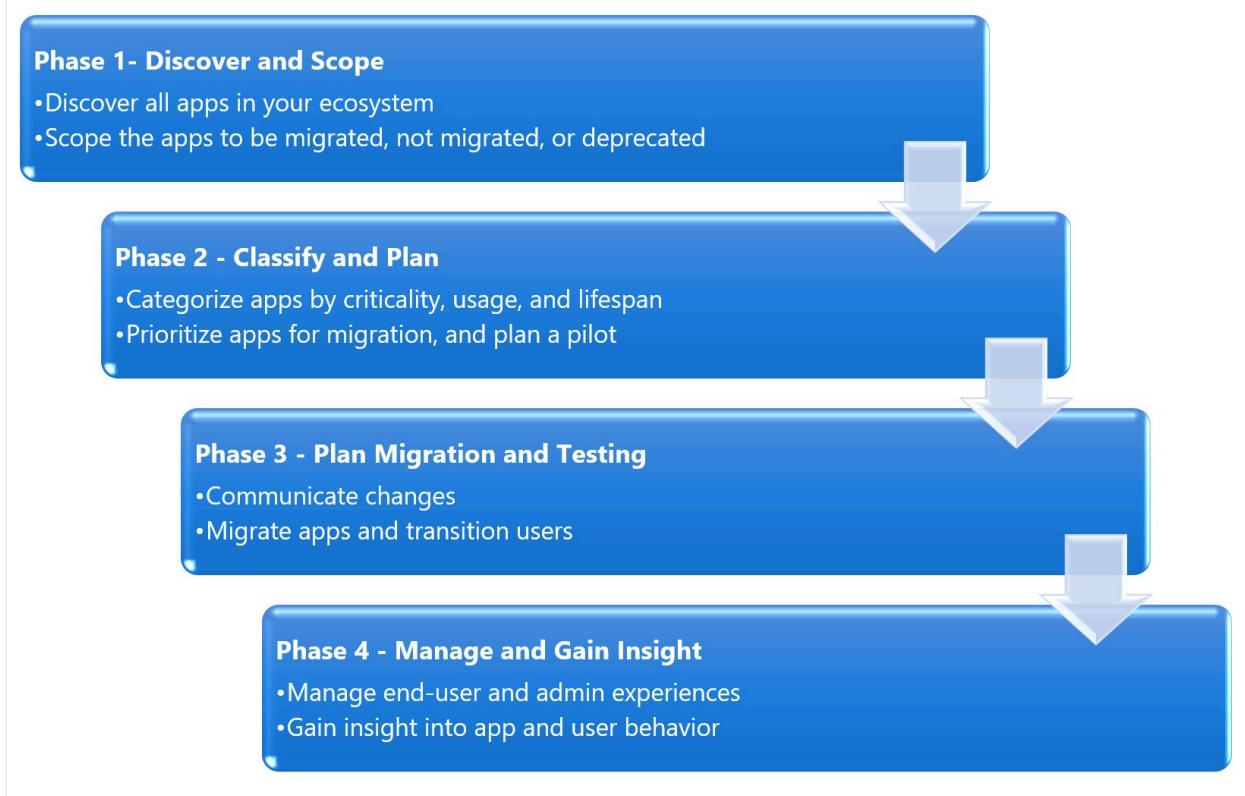
You can find more migration resources at <https://aka.ms/migrateapps>

Plan your migration phases and project strategy

When technology projects fail, it's often due to mismatched expectations, the right stakeholders not being involved, or a lack of communication. Ensure your success by planning the project itself.

The phases of migration

Before we get into the tools, you should understand how to think through the migration process. Through several direct-to-customer workshops, we recommend the following four phases:



Assemble the project team

Application migration is a team effort, and you need to ensure that you have all the vital positions filled. Support from senior business leaders is important. Ensure that you involve the right set of executive sponsors, business decision-makers, and subject matter experts (SMEs.)

During the migration project, one person might fulfill multiple roles, or multiple people fulfill each role, depending on your organization's size and structure. You might also have a dependency on other teams that play a key role in your security landscape.

The following table includes the key roles and their contributions:

[\[+\] Expand table](#)

Role	Contributions
Project Manager	<p>Project coach accountable for guiding the project, including:</p> <ul style="list-style-type: none"> - gain executive support - bring in stakeholders - manage schedules, documentation, and communications
Identity Architect / Microsoft Entra App Administrator	<p>Responsible for the following tasks:</p> <ul style="list-style-type: none"> - design the solution in cooperation with stakeholders - document the solution design and operational procedures for handoff to the operations team - manage the preproduction and production environments

Role	Contributions
On premises AD operations team	<p>The organization that manages the different on-premises identity sources such as AD forests, LDAP directories, HR systems, and so on.</p> <ul style="list-style-type: none"> - perform any remediation tasks needed before synchronizing - Provide the service accounts required for synchronization - provide access to configure federation to Microsoft Entra ID
IT Support Manager	A representative from the IT support organization who can provide input on the supportability of this change from a helpdesk perspective.
Security Owner	A representative from the security team that can ensure that the plan meets the security requirements of your organization.
Application technical owners	Includes technical owners of the apps and services that integrate with Microsoft Entra ID. They provide the applications' identity attributes that should include in the synchronization process. They usually have a relationship with CSV representatives.
Application business Owners	Representative colleagues who can provide input on the user experience and usefulness of this change from a user's perspective. This representative also owns the overall business aspect of the application, which might include managing access.
Pilot group of users	Users who test as a part of their daily work, the pilot experience, and provide feedback to guide the rest of the deployments.

Plan communications

Effective business engagement and communication are the keys to success. It's important to give stakeholders and end-users an avenue to get information and keep informed of schedule updates. Educate everyone about the value of the migration, what the expected timelines are, and how to plan for any temporary business disruption. Use multiple avenues such as briefing sessions, emails, one-to-one meetings, banners, and town halls.

Based on the communication strategy that you choose for the app you might want to remind users of the pending downtime. You should also verify that there are no recent changes or business impacts that would require to postpone the deployment.

In the following table, you find the minimum suggested communication to keep your stakeholders informed:

Plan phases and project strategy

[\[+\] Expand table](#)

Communication	Audience
Awareness and business / technical value of project	All except end users
Solicitation for pilot apps	<ul style="list-style-type: none">- App business owners- App technical owners- Architects and Identity team

Phase 1- Discover and Scope:

[\[+\] Expand table](#)

Communication	Audience
<ul style="list-style-type: none">- Solicitation for application information- Outcome of scoping exercise	<ul style="list-style-type: none">- App technical owners- App business owners

Phase 2- Classify apps and plan pilot:

[\[+\] Expand table](#)

Communication	Audience
<ul style="list-style-type: none">- Outcome of classifications and what that means for migration schedule- Preliminary migration schedule	<ul style="list-style-type: none">- App technical owners- App business owners

Phase 3 – Plan migration and testing:

[\[+\] Expand table](#)

Communication	Audience
<ul style="list-style-type: none">- Outcome of application migration testing	<ul style="list-style-type: none">- App technical owners- App business owners
<ul style="list-style-type: none">- Notification that migration is coming and explanation of resultant end-user experiences.- Downtimes coming and complete communications, including what they should now do, feedback, and how to get help	<ul style="list-style-type: none">- End users (and all others)

Phase 4 – Manage and gain insights:

[\[+\] Expand table](#)

Communication	Audience
Available analytics and how to access	- App technical owners - App business owners

Migration states communication dashboard

Communicating the overall state of the migration project is crucial, as it shows progress, and helps app owners whose apps are coming up for migration to prepare for the move. You can put together a simple dashboard using Power BI or other reporting tools to provide visibility into the status of applications during the migration.

The migration states you might consider using are as follows:

[\[+\] Expand table](#)

Migration states	Action plan
Initial Request	Find the app and contact the owner for more information
Assessment Complete	App owner evaluates the app requirements and returns the app questionnaire
Configuration in Progress	Develop the changes necessary to manage authentication against Microsoft Entra ID
Test Configuration Successful	Evaluate the changes and authenticate the app against the test Microsoft Entra tenant in the test environment
Production Configuration Successful	Change the configurations to work against the production AD tenant and assess the app authentication in the test environment
Complete / Sign Off	Deploy the changes for the app to the production environment and execute against the production Microsoft Entra tenant

This phase ensures app owners know what the app migration and testing schedule are when their apps are up for migration. They also know what the results are from other migrated apps. You might also consider providing links to your bug tracker database for owners to be able to file and view issues for apps that are being migrated.

Next steps

- Phase 1 - Discover and Scope.

Phase 1: Discover and scope apps

Article • 01/31/2025

Application discovery and analysis are a fundamental exercise to give you a good start. You may not know everything so be prepared to accommodate the unknown apps.

Find your apps

The first decision in the migration process is which apps to migrate, which if any should remain, and which apps to deprecate. There's always an opportunity to deprecate the apps that you won't use in your organization. There are several ways to find apps in your organization. While discovering apps, ensure you include in-development and planned apps. Use Microsoft Entra ID for authentication in all future apps.

Discover applications using ADFS:

- **Use Microsoft Entra Connect Health for ADFS:** If you have a Microsoft Entra ID P1 or P2 license, we recommend deploying [Microsoft Entra Connect Health](#) to analyze the app usage in your on-premises environment. You can use the [ADFS application report](#) to discover ADFS applications that can be migrated and evaluate the readiness of the application to be migrated.
- If you don't have Microsoft Entra ID P1 or P2 licenses, we recommend using the ADFS to Microsoft Entra app migration tools based on [PowerShell](#). Refer to [solution guide](#):

<https://www.youtube-nocookie.com/embed/PxLlacDpHh4>

ⓘ Note

This video covers both phase 1 and 2 of the migration process.

Using other identity providers (IdPs)

If you're using other identity providers, you can use the following approaches to discover applications:

- If you're currently using Okta, refer to our [Okta to Microsoft Entra migration guide](#).
- If you're currently using Ping Federate, then consider using the [Ping Administrative API](#)

- If the applications are integrated with Active Directory, search for service principals or service accounts that may be used for applications.

Using cloud discovery tools

In the cloud environment, you need rich visibility, control over data travel, and sophisticated analytics to find and combat cyber threats across all your cloud services. You can gather your cloud app inventory using the following tools:

- **Cloud Access Security Broker (CASB)** – A [CASB](#) typically works alongside your firewall to provide visibility into your employees' cloud application usage and helps you protect your corporate data from cybersecurity threats. The CASB report can help you determine the most used apps in your organization, and the early targets to migrate to Microsoft Entra ID.
- **Cloud Discovery** - By configuring [Microsoft Defender for Cloud Apps](#), you gain visibility into the cloud app usage, and can discover unsanctioned or Shadow IT apps.
- **Azure Hosted Applications** - For apps connected to Azure infrastructure, you can use the APIs and tools on those systems to begin to take an inventory of hosted apps. In the Azure environment:
 - Use the [Get-AzureWebsite](#) cmdlet to get information about Azure websites.
 - Use the [Get-AzWebApp](#) cmdlet to get information about your Azure Web Apps.
 - Query Microsoft Entra ID looking for [Applications](#) and [Service Principals](#).

Manual discovery process

Once you've taken the automated approaches described in this article, you have a good handle on your applications. However, you might consider doing the following to ensure you have good coverage across all user access areas:

- Contact the various business owners in your organization to find the applications in use in your organization.
- Run an HTTP inspection tool on your proxy server, or analyze proxy logs, to see where traffic is commonly routed.
- Review weblogs from popular company portal sites to see what links users access the most.
- Reach out to executives or other key business members to ensure that you've covered the business-critical apps.

Type of apps to migrate

Once you find your apps, you identify these types of apps in your organization:

- Apps that use modern authentication protocols such as [Security Assertion Markup Language \(SAML\)](#) or [OpenID Connect \(OIDC\)](#).
- Apps that use legacy authentication such as [Kerberos](#) or NT LAN Manager (NTLM) that you choose to modernize.
- Apps that use legacy authentication protocols that you choose NOT to modernize
- New Line of Business (LoB) apps

Apps that use modern authentication already

The already modernized apps are the most likely to be moved to Microsoft Entra ID. These apps already use modern authentication protocols such as SAML or OIDC and can be reconfigured to authenticate with Microsoft Entra ID.

We recommend you search and add applications from the [Microsoft Entra app gallery](#). If you don't find them in the gallery, you can still onboard a custom application.

Legacy apps that you choose to modernize

For legacy apps that you want to modernize, moving to Microsoft Entra ID for core authentication and authorization unlocks all the power and data-richness that the [Microsoft Graph](#) and [Intelligent Security Graph](#) have to offer.

We recommend updating the authentication stack code for these applications from the legacy protocol (such as Windows-Integrated Authentication, Kerberos, HTTP Headers-based authentication) to a modern protocol (such as SAML or OpenID Connect).

Legacy apps that you choose NOT to modernize

For certain apps using legacy authentication protocols, sometimes modernizing their authentication isn't the right thing to do for business reasons. These include the following types of apps:

- Apps kept on-premises for compliance or control reasons.
- Apps connected to an on-premises identity or federation provider that you don't want to change.
- Apps developed using on-premises authentication standards that you have no plans to move

Microsoft Entra ID can bring great benefits to these legacy apps. You can enable modern Microsoft Entra security and governance features like [Multi-Factor Authentication](#),

[Conditional Access](#), [Microsoft Entra ID Protection](#), [Delegated Application Access](#), and [Access Reviews](#) against these apps without touching the app at all!

- Start by extending these apps into the cloud with [Microsoft Entra application proxy](#).
- Or explore using one of our [Secure Hybrid Access \(SHA\) partner integrations](#) that you might have deployed already.

New Line of Business (LoB) apps

You usually develop LoB apps for your organization's in-house use. If you have new apps in the pipeline, we recommend using the [Microsoft identity platform](#) to implement OIDC.

Apps to deprecate

Apps without clear owners and clear maintenance and monitoring present a security risk for your organization. Consider deprecating applications when:

- Their **functionality is highly redundant** with other systems
- There's **no business owner**
- There's clearly **no usage**

We recommend that you **do not deprecate high impact, business-critical applications**. In those cases, work with business owners to determine the right strategy.

Exit criteria

You're successful in this phase with:

- A good understanding of the applications in scope for migration, those that require modernization, those that should stay as-is, or those you've marked for deprecation.

Next steps

- Phase 2 - [Classify apps and plan pilot](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Phase 2: Classify apps and plan pilot

Article • 08/25/2024

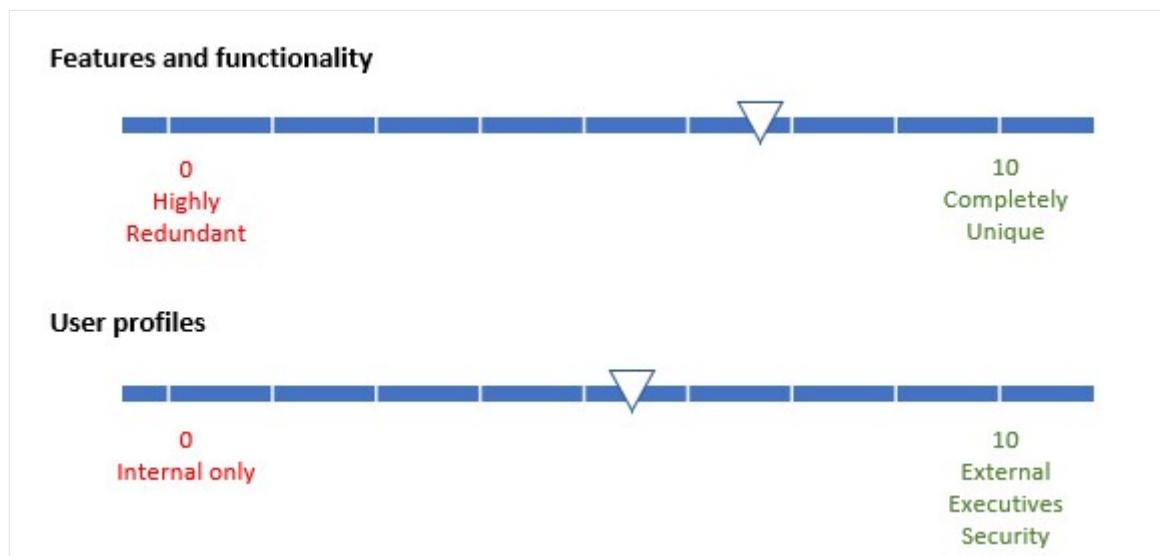
Classifying the migration of your apps is an important exercise. Not every app needs to be migrated and transitioned at the same time. Once you collect information about each of the apps, you can rationalize which apps should be migrated first and which might take added time.

Classify in-scope apps

One way to think about this aspect is along the axes of business criticality, usage, and lifespan, each of which is dependent on multiple factors.

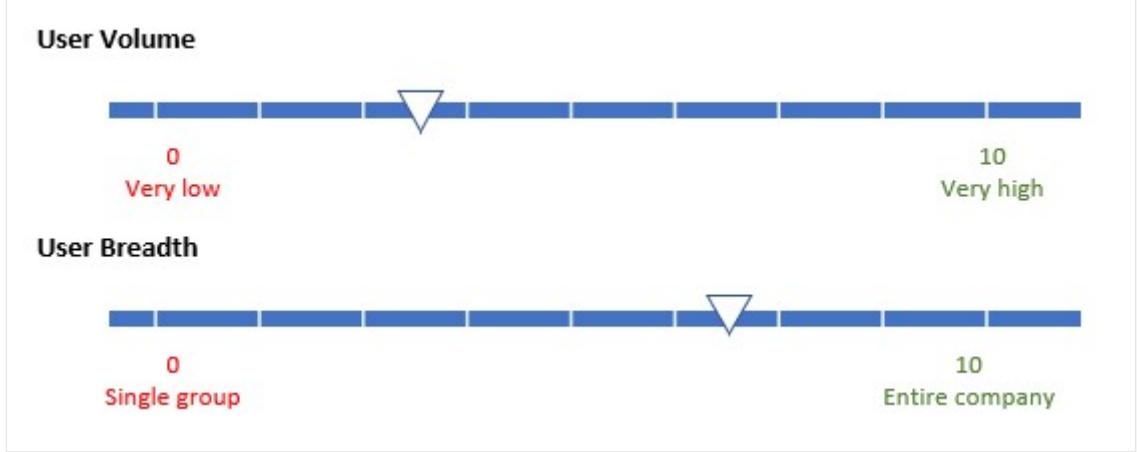
Business criticality

Business criticality takes on different dimensions for each business, but the two measures that you should consider are **features and functionality** and **user profiles**. Assign apps with unique functionality a higher point value than apps with redundant or obsolete functionality.

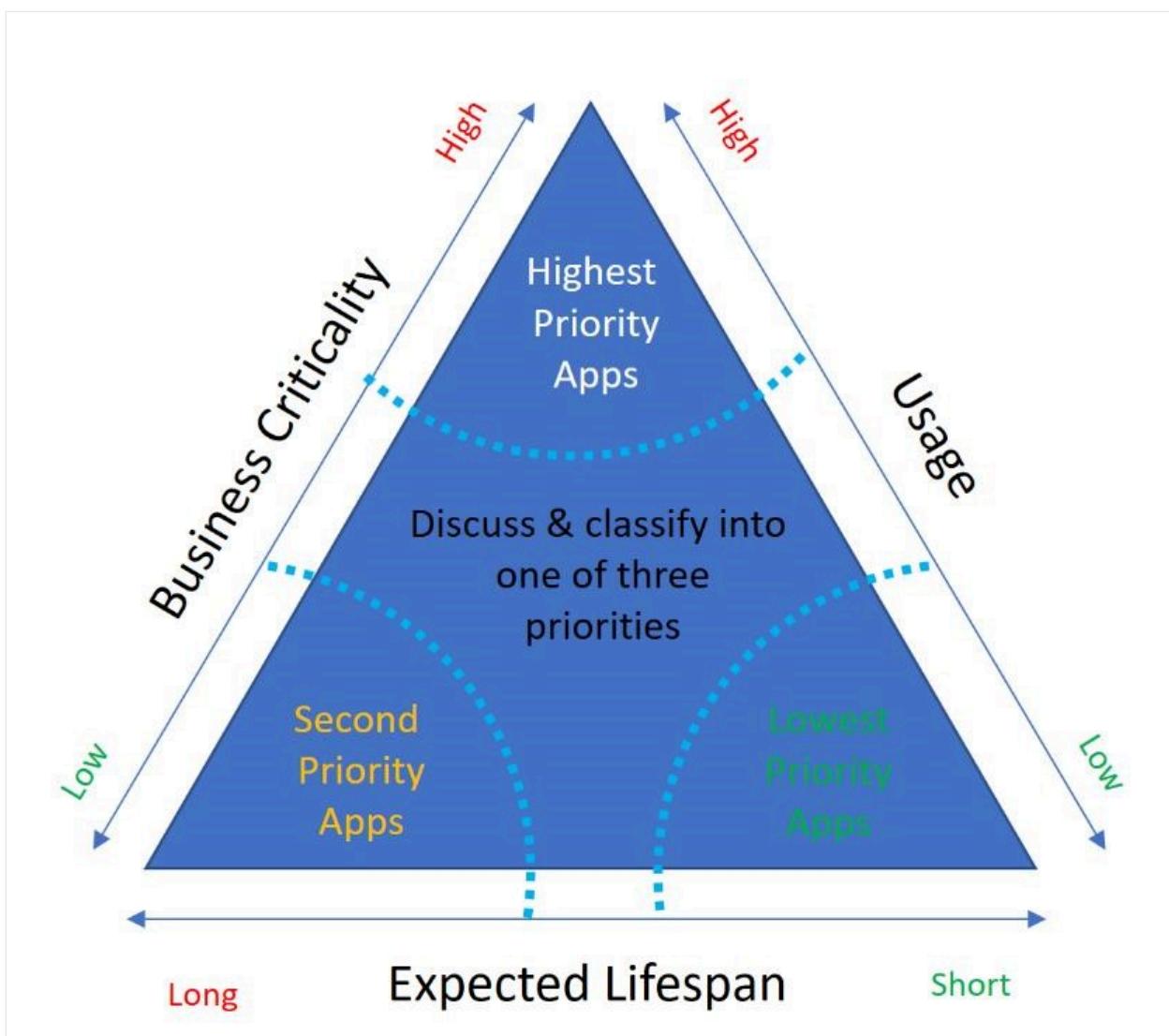


Usage

Applications with **high usage numbers** should receive a higher value than apps with low usage. Assign a higher value to apps with external, executive, or security team users. For each app in your migration portfolio, complete these assessments.



Once you determined values for business criticality and usage, you can then determine the **application lifespan**, and create a matrix of priority. The diagram shows the matrix.



[https://www.youtube-nocookie.com/embed/PxLiacDpHh4 ↗](https://www.youtube-nocookie.com/embed/PxLiacDpHh4)

ⓘ Note

This video covers both phase 1 and 2 of the migration process.

Prioritize apps for migration

You can choose to begin the app migration with either the lowest priority apps or the highest priority apps based on your organization's needs.

In a scenario where you might not have experience using Microsoft Entra ID and Identity services, consider moving your **lowest priority apps** to Microsoft Entra first. This option minimizes your business impact, and you can build momentum. Once you successfully move these apps and gain the stakeholder's confidence, you can continue to migrate the other apps.

If there's no clear priority, you should consider moving the apps that are in the [Microsoft Entra Gallery](#) first and support multiple identity providers because they're easier to integrate. It's likely that these apps are the **highest-priority apps** in your organization. To help integrate your SaaS applications with Microsoft Entra ID, There's a collection of [tutorials](#) that walk you through configuration.

When you have a deadline to migrate the apps, these highest priority apps bucket takes the major workload. You can eventually select the lower priority apps as they don't change the cost even if you move the deadline.

In addition to this classification and depending on the urgency of your migration, you should publish a **migration schedule** within which app owners must engage to have their apps migrated. At the end of this process, you should have a list of all applications in prioritized buckets for migration.

Document your apps

First, start by gathering key details about your applications. The [Application Discovery Worksheet](#) helps you to make your migration decisions quickly and get a recommendation out to your business group in no time at all.

Information that is important to making your migration decision includes:

- **App name** – what is this app known as to the business?
- **App type** – is it a third-party SaaS app? A custom line-of-business web app? An API?
- **Business criticality** – is its high criticality? Low? Or somewhere in between?
- **User access volume** – does everyone access this app or just a few people?
- **User access type**: who needs to access the application – Employees, business partners, or customers or perhaps all?

- **Planned lifespan** – how long it's availability? Less than six months? More than two years?
- **Current identity provider** – what is the primary IdP for this app? AD FS, Active Directory, or Ping Federate?
- **Security requirements** - does the application require MFA or that users be on the corporate network to access the application?
- **Method of authentication** – does the app authenticate using open standards?
- **Whether you plan to update the app code** – is the app under planned or active development?
- **Whether you plan to keep the app on-premises** – do you want to keep the app in your datacenter long term?
- **Whether the app depends on other apps or APIs** – does the app currently call into other apps or APIs?
- **Whether the app is in the Microsoft Entra gallery** – is the app currently already integrated with the [Microsoft Entra Gallery](#)?

Other data that helps you later, but that you don't need to make an immediate migration decision includes:

- **App URL** – where do users go to access the app?
- **Application Logo:** If migrating an application to Microsoft Entra ID that isn't in the Microsoft Entra app gallery, we recommend you provide a descriptive logo
- **App description** – what is a brief description of what the app does?
- **App owner** – who in the business is the main POC for the app?
- **General comments or notes** – any other general information about the app or business ownership

Once you classify your application and documented the details, then be sure to gain business owner buy-in to your planned migration strategy.

Application users

There are two main categories of users of your apps and resources that Microsoft Entra ID supports:

- **Internal:** Employees, contractors, and vendors that have accounts within your identity provider. This category might need further pivots with different rules for managers or leadership versus other employees.
- **External:** Vendors, suppliers, distributors, or other business partners that interact with your organization in the regular course of business with [Microsoft Entra B2B collaboration](#).

You can define groups for these users and populate these groups in diverse ways. You might choose that an administrator must manually add members into a group, or you can enable self-service dynamic membership groups. Rules can be established that automatically add members into groups based on the specified criteria using [dynamic membership groups](#).

External users might also refer to customers. [Azure AD B2C](#), a separate product supports customer authentication. However, it is outside the scope of this paper.

Plan a pilot

The app(s) you select for the pilot should represent the key identity and security requirements of your organization, and you must have clear buy-in from the application owners. Pilots typically run in a separate test environment.

Don't forget about your external partners. Make sure that they participate in migration schedules and testing. Finally, ensure they have a way to access your helpdesk if there were breaking issues.

Plan for limitations

While some apps are easy to migrate, others might take longer due to multiple servers or instances. For example, SharePoint migration might take longer due to custom sign-in pages.

Many SaaS app vendors might not provide a self-service means to reconfigure the application and might charge for changing the SSO connection. Check with them and plan for limitation.

App owner approval

Business critical and universally used applications might need a group of pilot users to test the app in the pilot stage. Once you test an app in the preproduction or pilot environment, ensure that app business owners give approval on performance prior to the migration of the app. You should also ensure that all users migrate to production use of Microsoft Entra ID for authentication.

Plan the security posture

Before you initiate the migration process, take time to fully consider the security posture you wish to develop for your corporate identity system. This aspect is based on gathering these valuable sets of information: **Identities, devices, and locations that are accessing your applications and data.**

Identities and data

Most organizations have specific requirements about identities and data protection that vary by industry segment and by job functions within organizations. Refer to [identity and device access configurations](#) for our recommendations. The recommendations include a prescribed set of [Conditional Access policies](#) and related capabilities.

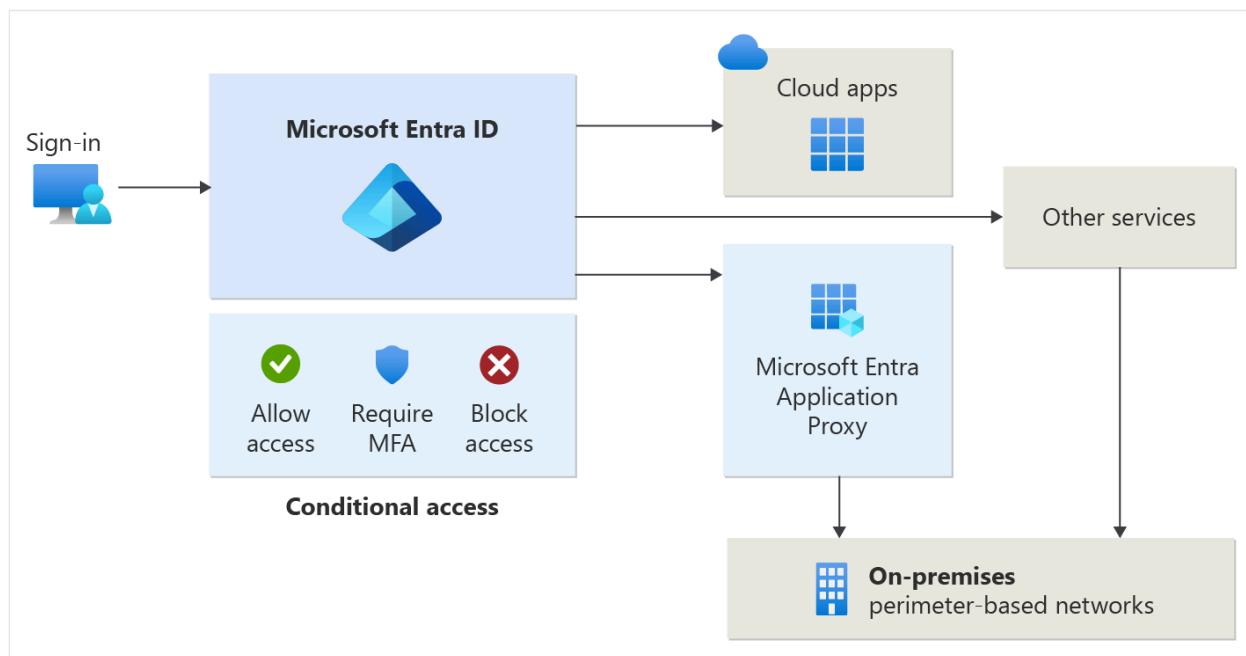
You can use this information to protect access to all services integrated with Microsoft Entra ID. These recommendations are aligned with Microsoft Secure Score and the [identity score in Microsoft Entra ID](#). The score helps you to:

- Objectively measure your identity security posture
- Plan identity security improvements
- Review the success of your improvements

The Microsoft Secure Score also helps you implement the [five steps to securing your identity infrastructure](#). Use the guidance as a starting point for your organization and adjust the policies to meet your organization's specific requirements.

Device/location used to access data

The device and location that a user uses to access an app are also important. Devices physically connected to your corporate network are more secure. Connections from outside the network over VPN might need scrutiny.



With these aspects of resource, user, and device in mind, you might choose to use [Microsoft Entra Conditional Access](#) capabilities. Conditional Access goes beyond user permissions. It depends on a combination of factors such as:

- The identity of a user or group
- The network that the user is connected to
- The device and application the user is using
- The type of data the user is trying to access.

The access granted to the user adapts to this broader set of conditions.

Exit criteria

You're successful in this phase when you have:

- Fully documented the apps you intend to migrate
- Prioritized apps based on business criticality, usage volume, and lifespan
- Selected apps that represent your requirements for a pilot
- Business-owner buy-in to your prioritization and strategy
- Understanding of your security posture needs and how to implement them

Next steps

- Phase 3 - Plan migration and testing

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Phase 3: Plan migration and testing

Article • 10/23/2023

Once you've gained business buy-in, the next step is to start migrating these apps to Microsoft Entra authentication.

Migration tools and guidance

Use the tools and guidance provided to follow the precise steps needed to migrate your applications to Microsoft Entra ID:

- **General migration guidance** – Use the whitepaper, tools, email templates, and applications questionnaire in the [Microsoft Entra apps migration toolkit](#) to discover, classify, and migrate your apps.
- **SaaS applications** – See our list of [SaaS app tutorials](#) and the [Microsoft Entra SSO deployment plan](#) to walk through the end-to-end process.
- **Applications running on-premises** – Learn all about the [Microsoft Entra application proxy](#) and use the complete [Microsoft Entra application proxy deployment plan](#) to get going quickly or consider our [Secure Hybrid Access partners](#), which you may already own.
- **Apps you're developing** – Read our step-by-step [integration](#) and [registration](#) guidance.

https://www.youtube-nocookie.com/embed/PvI4Q4P_HfU

Plan testing

During the process of the migration, your app may already have a test environment used during regular deployments. You can continue to use this environment for migration testing. If a test environment isn't currently available, you may be able to set one up using Azure App Service or Azure Virtual Machines, depending on the architecture of the application.

You may choose to set up a separate test Microsoft Entra tenant to use as you develop your app configurations. This tenant starts in a clean state and won't be configured to sync with any system.

Depending on how you configure your app, verify that SSO works properly.

[+] Expand table

Authentication type	Testing
OAuth / OpenID Connect	Select Enterprise applications > Permissions and ensure you've consented to the application to be used in your organization in the user settings for your app.
SAML-based SSO	Use the Test SAML Settings button found under Single Sign-On .
Password-Based SSO	Download and install the MyApps Secure Sign-in Extension . This extension helps you start any of your organization's cloud apps that require you to use an SSO process.
Application Proxy	Ensure your connector is running and assigned to your application. Visit the Application Proxy troubleshooting guide for further assistance.

You can test each app by logging in with a test user and make sure all functionality is the same as prior to the migration. If you determine during testing that users need to update their [MFA](#) or [SSPR](#) settings, or you're adding this functionality during the migration, be sure to add that to your end-user communication plan. See [MFA](#) and [SSPR](#) end-user communication templates.

Troubleshoot

If you run into problems, check out our [apps troubleshooting guide](#) and [Secure Hybrid Access partner integration article](#) to get help. You can also check out our troubleshooting articles, see [Problems signing in to SAML-based single sign-on configured apps](#).

Plan rollback

If the migration fails, we recommend that you leave the existing Relying Parties on the AD FS servers and remove access to the Relying Parties. This allows for a quick fallback if needed during the deployment.

Consider the following suggestions for actions you can take to mitigate migration issues:

- **Take screenshots** of the existing configuration of your app. You can look back if you must reconfigure the app once again.
- You might also consider **providing links for the application to use alternative authentication options (legacy or local authentication)**, in case there are issues with cloud authentication.

- Before you complete your migration, **do not change your existing configuration** with the existing identity provider.
- Be aware of the **apps that support multiple IdPs** since they provide an easier rollback plan.
- Ensure that your app experience has a **Feedback button** or pointers to your **helpdesk** issues.

Employee communication

While the planned outage window itself can be minimal, you should still plan on communicating these timeframes proactively to employees while switching from AD FS to Microsoft Entra ID. Ensure that your app experience has a feedback button, or pointers to your helpdesk for issues.

Once deployment is complete, you can inform users of the successful deployment and remind them of any steps that they need to take.

- Instruct users to use [My Apps](#) to access all the migrated applications.
- Remind users they might need to update their MFA settings.
- If Self-Service Password Reset is deployed, users might need to update or verify their authentication methods. See [MFA](#) and [SSPR](#) end-user communication templates.

External user communication

This group of users is usually the most critically impacted in case of any issues. This is especially true if your security posture dictates a different set of Conditional Access rules or risk profiles for external partners. Ensure that external partners are aware of the cloud migration schedule and have a timeframe during which they're encouraged to participate in a pilot deployment that tests out all flows unique to external collaboration. Finally, ensure they have a way to access your helpdesk in case there are problems.

Exit criteria

You're successful in this phase when you have:

- Reviewed the migration tools
- Planned your testing including test environments and groups
- Planned rollback

Next steps

- Phase 4 - Manage and gain insights

Phase 4: Plan management and insights

Article • 10/23/2023

Once apps are migrated, you must ensure that:

- Users can securely access and manage
- You can gain the appropriate insights into usage and app health

We recommend taking the following actions as appropriate to your organization.

Manage your users' app access

Once you've migrated the apps, consider applying the following suggestions to enrich your user's experience:

- Make apps discoverable by publishing them to the [Microsoft MyApplications portal](#).
- Add [app collections](#) so users can locate application based on business function.
- Add their own application bookmarks to the [MyApplications portal](#).
- Enable [self-service application access](#) to an app and [let users add apps that you curate](#).
- Optionally [hide applications from end-users](#).
- Users can go to [Office.com](#) to [search for their apps and have their most-recently-used apps appear](#) for them right from where they do work.
- Users can download the MyApps secure sign-in extension in Chrome, or Microsoft Edge so they can launch applications directly from their browser without having to first navigate to MyApplications.
- Users can access the MyApps portal with Intune-managed browser on their [iOS 7.0 or later](#) or [Android](#) devices.
 - For [Android devices](#), from the [Google play store](#)
 - For [Apple devices](#), from the [Apple App Store](#) or they can download the My Apps mobile app for [iOS](#).

<https://www.youtube-nocookie.com/embed/8aUluOXeDxw>

Secure app access

Microsoft Entra ID provides a centralized access location to manage your migrated apps. Sign in to the [Microsoft Entra admin center](#) and enable the following capabilities:

- **Secure user access to apps.** Enable [Conditional Access policies](#) or [Identity Protection](#) to secure user access to applications based on device state, location, and more.
- **Automatic provisioning.** Set up [automatic provisioning of users](#) with various third-party SaaS apps that users need to access. In addition to creating user identities, it includes the maintenance and removal of user identities as status or roles change.
- **Delegate user access management.** As appropriate, enable self-service application access to your apps and *assign a business approver to approve access to those apps*. Use [Self-Service Group Management](#) for groups assigned to collections of apps.
- **Delegate admin access using Directory Role** to assign an admin role (such as Application administrator, Cloud Application administrator, or Application developer) to your user.
- **Add applications to Access Packages** to provide governance and attestation.

Audit and gain insights of your apps

You can also use the [Microsoft Entra admin center](#) to audit all your apps from a centralized location,

- **Audit your app** using [Enterprise Applications](#), [Audit](#), or access the same information from the [Microsoft Entra reporting API](#) to integrate into your favorite tools.
- **View the permissions for an app** using [Enterprise Applications](#), [Permissions for apps](#) using OAuth/OpenID Connect.
- **Get sign-in insights** using [Enterprise Applications](#), [Sign-Ins](#). Access the same information from the [Microsoft Entra reporting API](#).
- **Visualize your app's usage** from the [Microsoft Entra ID Power BI content pack](#)

Exit criteria

You're successful in this phase when you:

- Provide secure app access to your users
- Manage to audit and gain insights of the migrated apps

Do even more with deployment plans

Deployment plans walk you through the business value, planning, implementation steps, and management of Microsoft Entra solutions, including app migration scenarios. They bring together everything that you need to start deploying and getting value out of Microsoft Entra capabilities. The deployment guides include content such as Microsoft recommended best practices, end-user communications, planning guides, implementation steps, test cases, and more.

Many [deployment plans](#) are available for your use, and we're always making more!

Contact support

Visit the following support links to create or track support ticket and monitor health.

- **Azure Support:** You can call [Microsoft Support](#) and open a ticket for any Azure Identity deployment issue depending on your Enterprise Agreement with Microsoft.
- **FastTrack:** If you've purchased Enterprise Mobility and Security (EMS) or Microsoft Entra ID P1 or P2 licenses, you're eligible to receive deployment assistance from the [FastTrack program](#).
- **Engage the Product Engineering team:** If you're working on a major customer deployment with millions of users, you're entitled to support from the Microsoft account team or your Cloud Solutions Architect. Based on the project's deployment complexity, you can work directly with the [Azure Identity Product Engineering team](#).

Next steps

- [Migration process](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) | [Get help at Microsoft Q&A](#)

Articles for integrating applications with Microsoft Entra ID

Article • 05/20/2025

To help integrate your cloud-enabled [software as a service \(SaaS\)](#) and on-premises applications with Microsoft Entra ID, we have developed a collection of articles that walk you through configuration.

For a list of all SaaS apps that have been preintegrated into Microsoft Entra ID, see the [Microsoft Entra Marketplace](#).

Use the [application network portal](#) to request a SCIM enabled application to be added to the gallery for automatic provisioning or a SAML / OIDC enabled application to be added to the gallery for SSO.

Quick links

 Expand table

Logo	Application article for single sign-on	Application article for user provisioning
	Atlassian Cloud	Atlassian Cloud - User Provisioning
	ServiceNow	ServiceNow - User Provisioning
	Slack	Slack - User Provisioning
	SuccessFactors	SuccessFactors - User Provisioning
	Workday	Workday - User Provisioning

To find more articles, use the table of contents on the left.

Cloud Integrations

 Expand table

Logo	Application article for single sign-on	Application article for user provisioning
	Amazon Web Services (AWS) Console	Amazon Web Services (AWS) Console - Role Provisioning
	Alibaba Cloud Service (Role based SSO)	
	Google Cloud Platform	Google Cloud Platform - User Provisioning
	Salesforce	Salesforce - User Provisioning
	SAP Cloud Identity Services	SAP Cloud Identity Services - Provisioning

Related content

To learn more about application management, see [What is application management](#).

Assign enterprise application owners

Article • 04/25/2025

An [owner of an enterprise application](#) in Microsoft Entra ID can manage the organization-specific configuration of the application, such as single sign-on, provisioning, and user assignments. An owner can also add or remove other owners. Unlike other Application Administrators, owners can manage only the enterprise applications they own. In this article, you learn how to assign an owner of an application.

Prerequisites

To add an enterprise application to your Microsoft Entra tenant, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free ↗](#).
- One of the following roles: Cloud Application Administrator, or Application Administrator.

Assign an owner

To assign an owner to an enterprise application:

1. Sign in to the [Microsoft Entra admin center ↗](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Entra ID > Enterprise apps > All applications**.
3. Select the application that you want to add an owner to.
4. Select **Owners**, and then select **Add** to get a list of user accounts that you can choose an owner from.
5. Search for and select the user account that you want to be an owner of the application.
6. Select **Select** to add the user account that you chose as an owner of the application.

ⓘ Note

If the user setting **Restrict access to Microsoft Entra administration portal** is set to **Yes**, non-admin users aren't able to use the Microsoft Entra admin center to manage the applications they own. For more information about the actions that can be performed on owned enterprise applications, see [Owned enterprise applications](#).

Next steps

- [Delegate app registration permissions in Microsoft Entra ID](#)

Configure enterprise application properties

Article • 04/08/2025

This article shows you where you can configure the properties of an enterprise application in your Microsoft Entra tenant. For more information about the properties that you can configure, see [Properties of an enterprise application](#).

Prerequisites

To configure the properties of an enterprise application, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free ↗](#).
- One of the following roles:
 - Cloud Application Administrator
 - Application Administrator, or owner of the service principal.

Configure application properties

Application properties control how the application is represented and how the application is accessed.

To configure the application properties:

1. Sign in to the [Microsoft Entra admin center ↗](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. Search for and select the application that you want to use.
4. In the **Manage** section, select **Properties** to open the **Properties** pane for editing.
5. On the **Properties** pane, you might want to configure the following properties for your application.
 - Logo
 - User sign in options
 - App visibility to users
 - Set available URL options
 - Choose whether app assignment is required
6. After you configure the properties according to your apps needs, select **Save**.

Note

Managed identities are distinct from Microsoft Entra App Registrations. Managed identities only have a service principal object and do not possess an application object, which is typically used for granting app permissions. As a result, global admins cannot change the settings of a managed identity, as the security boundary is the resource itself.

Use Microsoft Graph to configure advanced app properties

You can also configure other advanced properties of both app registrations and enterprise applications (service principals) through Microsoft Graph. These properties include permissions and role assignments. For more information, see [Create and manage a Microsoft Entra application using Microsoft Graph](#).

Related content

- [What is application management in Microsoft Entra ID?](#)

ⓘ Note: The author created this article with assistance from AI. [Learn more](#)

Create an enterprise application from a multitenant application in Microsoft Entra ID

Article • 07/19/2024

In this article, you'll learn how to create an enterprise application in your tenant using the client ID for a multitenant application. An enterprise application refers to a service principal within a tenant. The service principal discussed in this article is the local representation, or application instance, of a global application object in a single tenant or directory.

Before you proceed to add the application using any of these options, check whether the enterprise application is already in your tenant by attempting to sign in to the application. If the sign-in is successful, the enterprise application already exists in your tenant.

If you have verified that the application isn't in your tenant, proceed with any of the following ways to add the enterprise application to your tenant.

Prerequisites

To add an enterprise application to your Microsoft Entra tenant, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free](#).
- One of the following roles: Cloud Application Administrator, or Application Administrator.
- The client ID (also called appId in Microsoft Graph) of the multitenant application.

Create an enterprise application

1. Run `connect-MgGraph -Scopes "Application.ReadWrite.All"` and sign in with at least a Cloud Application Administrator role.
2. Run the following command to create the enterprise application:

PowerShell

```
New-MgServicePrincipal -AppId 00001111-aaaa-2222-bbbb-3333cccc4444
```

3. To delete the enterprise application you created, run the command:

```
PowerShell
```

```
Remove-MgServicePrincipal  
-ServicePrincipalId bbbbbbbb-1111-2222-3333-cccccccccccc
```

Next steps

- Add RBAC role to the enterprise application
- Assign users to your application

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Enable SAML single sign-on for an enterprise application

Article • 02/14/2025

In this article, you use the Microsoft Entra admin center to enable single sign-on (SSO) for an enterprise application that you added to your Microsoft Entra tenant. After you configure SSO, your users can sign in by using their Microsoft Entra credentials.

Microsoft Entra ID has a gallery that contains thousands of preintegrated applications that use SSO. This article uses an enterprise application named **Microsoft Entra SAML Toolkit 1** as an example, but the concepts apply for most preconfigured enterprise applications in the Microsoft Entra application gallery.

If your application will not integrate directly with Microsoft Entra for single sign-on, and instead tokens are provided to the application by a relying party Security Token Service (STS), then see the article [Enable single sign-on for an enterprise application with a relying party security token service](#).

We recommend that you use a nonproduction environment to test the steps in this article.

Prerequisites

To configure SSO, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free](#).
- One of the following roles: Cloud Application Administrator, Application Administrator, or owner of the service principal.
- Completion of the steps in [Quickstart: Create and assign a user account](#).

Enable single sign-on

To enable SSO for an application:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.

3. Enter the name of the existing application in the search box, and then select the application from the search results. For example, **Microsoft Entra SAML Toolkit 1**.
4. In the **Manage** section of the left menu, select **Single sign-on** to open the **Single sign-on** pane for editing.
5. Select **SAML** to open the SSO configuration page. After the application is configured, users can sign in to it by using their credentials from the Microsoft Entra tenant.
6. The process of configuring an application to use Microsoft Entra ID for SAML-based SSO varies depending on the application. For any of the enterprise applications in the gallery, use the **configuration guide** link to find information about the steps needed to configure the application. The steps for the **Microsoft Entra SAML Toolkit 1** are listed in this article.

The screenshot shows the Microsoft Entra SAML Toolkit 1 configuration page. The left sidebar has a 'Manage' section with 'Single sign-on' selected. The main area is titled 'Set up Single Sign-On with SAML'. It contains two numbered steps: Step 1, 'Basic SAML Configuration', which lists fields: Identifier (Entity ID) [Required], Reply URL (Assertion Consumer Service URL) [Required], Sign on URL [Required], Relay State (Optional), and Logout Url (Optional); Step 2, 'Attributes & Claims', which shows mappings: givenname to user.givenname, surname to user.surname, and emailaddress to user.mail. A note says 'Fill out required fields in Step 1'.

7. In the **Set up Microsoft Entra SAML Toolkit 1** section, record the values of the **Login URL**, **Microsoft Entra Identifier**, and **Logout URL** properties to be used later.

Configure single sign-on in the tenant

You add sign-in and reply URL values, and you download a certificate to begin the configuration of SSO in Microsoft Entra ID.

To configure SSO in Microsoft Entra ID:

1. In the Microsoft Entra admin center, select **Edit** in the **Basic SAML Configuration** section on the **Set up Single Sign-On with SAML** pane.
2. For **Reply URL (Assertion Consumer Service URL)**, enter
`https://samltoolkit.azurewebsites.net/SAML/Consume`.
3. For **Sign on URL**, enter `https://samltoolkit.azurewebsites.net/`. The **Identifier (Entity ID)** is typically a URL specific to the application you're integrating with. For the **Microsoft Entra SAML Toolkit 1** application in this example, the value is automatically generated once you input the **Sign on URL** and **Reply URL** values. Follow the specific configuration guide for the application you're integrating with to determine the correct value.
4. Select **Save**.
5. In the **SAML Certificates** section, select **Download for Certificate (Raw)** to download the SAML signing certificate and save it to be used later.

Configure single sign-on in the application

Using single sign-on in the application requires you to register the user account with the application and to add the SAML configuration values that you previously recorded.

Register the user account

To register a user account with the application:

1. Open a new browser window and browse to the sign-in URL for the application.
For the **Microsoft Entra SAML Toolkit** application, the address is
`https://samltoolkit.azurewebsites.net`.
2. Select **Register** in the upper right corner of the page.
3. For **Email**, enter the email address of the user that can access the application.
Ensure that the user account is already assigned to the application.
4. Enter a **Password** and confirm it.
5. Select **Register**.

Configure SAML settings

To configure SAML settings for the application:

1. On the application's sign-in page, sign in with the credentials of the user account that you already assigned to the application, select **SAML Configuration** at the

upper-left corner of the page.

2. Select **Create** in the middle of the page.
3. For **Login URL**, **Microsoft Entra Identifier**, and **Logout URL**, enter the values that you recorded earlier.
4. Select **Choose file** to upload the certificate that you previously downloaded.
5. Select **Create**.
6. Copy the values of the **SP Initiated Login URL** and the **Assertion Consumer Service (ACS) URL** to be used later.

Update single sign-on values

Use the values that you recorded for **SP Initiated Login URL** and **Assertion Consumer Service (ACS) URL** to update the single sign-on values in your tenant.

To update the single sign-on values:

1. In the Microsoft Entra admin center, select **Edit** in the **Basic SAML Configuration** section on the **Set up single sign-on** pane.
2. For **Reply URL (Assertion Consumer Service URL)**, enter the **Assertion Consumer Service (ACS) URL** value that you previously recorded.
3. For **Sign on URL**, enter the **SP Initiated Login URL** value that you previously recorded.
4. Select **Save**.

Test single sign-on

You can test the single sign-on configuration from the **Set up single sign-on** pane.

To test SSO:

1. In the **Test single sign-on with Microsoft Entra SAML Toolkit 1** section, on the **Set up single sign-on with SAML** pane, select **Test**.
2. Sign in to the application using the Microsoft Entra credentials of the user account that you assigned to the application.

Next steps

- [Manage self service access](#)
- [Configure user consent](#)
- [Grant tenant-wide admin consent](#)

Feedback

Was this page helpful?

 Yes

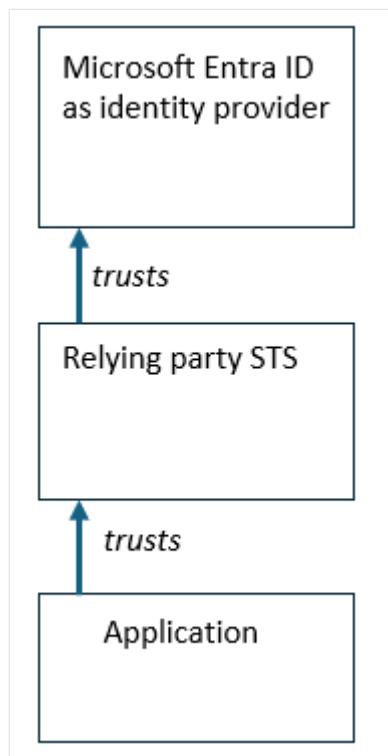
 No

[Provide product feedback ↗](#)

Enable single sign-on for an enterprise application with a relying party STS

Article • 02/14/2025

In this article, you use the Microsoft Entra admin center to enable single sign-on (SSO) for an enterprise application which is dependent upon a relying party security token service (STS). The relying party STS supports the Security Assertion Markup Language (SAML), and can be integrated with Microsoft Entra as an enterprise application. After you configure SSO, your users can sign in to the application by using their Microsoft Entra credentials.



If your application will integrate directly with Microsoft Entra for single sign-on, and does not require a relying party Security Token Service (STS), then see the article [Enable single sign-on for an enterprise application](#).

We recommend that you use a nonproduction environment to test the steps in this article, before configuring an application in a production tenant.

Prerequisites

To configure SSO, you need:

- A relying party STS, such as Active Directory Federation Services (AD FS) or PingFederate, with HTTPS endpoints

1. You'll need the entity identifier (entity ID) of the relying party STS. This must be unique across all relying party STS and applications configured in a Microsoft Entra tenant. There cannot be two applications in a single Microsoft Entra tenant with the same entity identifier. For example, if Active Directory Federation Services (AD FS) is the relying party STS, then the identifier may be a URL of the form `http://{hostname.domain}/adfs/services/trust`.
 2. You'll also need the assertion consumer service URL, or reply URL, of the relying party STS. This URL must be a `HTTPS` URL to securely transfer SAML tokens from Microsoft Entra to the relying party STS as part of single sign-on to an application. For example, if AD FS is the relying party STS, then the URL may be of the form `https://{hostname.domain}/adfs/ls/`.
- An application, which has already been integrated with that relying party STS
 - One of the following roles in Microsoft Entra: Cloud Application Administrator, Application Administrator
 - A test user in Microsoft Entra who can sign into the application

Note

This tutorial assumes that there is one Microsoft Entra tenant, one relying party STS, and one application connected to the relying party STS. This tutorial shows how to configure Microsoft Entra to use the entity identifier provided by the relying party STS to determine the appropriate enterprise application, to send a SAML token in a response. If you had more than one application connected to a single relying party STS, then Microsoft Entra wouldn't be able to distinguish between those two applications when issuing SAML tokens. Configuring different entity identifiers is outside the scope of this tutorial.

Create an application in Microsoft Entra

First, create an enterprise application in Microsoft Entra, which enables Microsoft Entra to generate SAML tokens for the relying party STS to provide to the application.

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. If you have already configured an application representing the relying party STS, then enter the name of the existing application in the search box, select the application from the search results, and continue at the next section.
4. Select **New application**.
5. Select **Create your own application**.

- Type the name of the new application in the input name box, select **Integrate any other application you don't find in the gallery (Non-gallery)**, and select **Create**.

Configure single sign-on in the application

- In the **Manage** section of the left menu, select **Single sign-on** to open the **Single sign-on** pane for editing.
- Select **SAML** to open the SSO configuration page.
- In the **Basic SAML configuration** box, select **Edit**. The identifier and reply URL must be set before further SAML configuration changes can be made.

The screenshot shows the 'Basic SAML Configuration' dialog box. It has a title bar with 'Basic SAML Configuration' and an 'Edit' button with a pencil icon. Below the title, there are five configuration items with their status indicated:

Setting	Status
Identifier (Entity ID)	Required
Reply URL (Assertion Consumer Service URL)	Required
Sign on URL	Optional
Relay State (Optional)	Optional
Logout Url (Optional)	Optional

- In the Basic SAML configuration page, under **Identifier (Entity ID)**, if there's no identifier listed, select **Add identifier**. Type the identifier for the application as provided by the relying party STS. For example, the identifier may be a URL of the form `http://{hostname.domain}/adfs/services/trust`.
- In the Basic SAML configuration page, under **Reply URL (Assertion Consumer Service URL)**, select **Add reply URL**. Type the HTTPS URL of the relying party STS Assertion Consumer Service. For example, the URL may be of the form `https:// {hostname.domain}/adfs/ls/`.
- Optionally, configure the **sign on**, **relay state**, or **logout** URLs, if required by the relying party STS.
- Select **Save**.

Download metadata and certificates from Microsoft Entra

Your relying party STS may require the federation metadata from Microsoft Entra as the identity provider in order to complete the configuration. The federation metadata and associated certificates are provided in the **SAML Certificates** section of the **Basic SAML configuration** page. For more information, see [federation metadata](#).

SAML Certificates	
Token signing certificate	 Edit
Status	Active
Thumbprint	
Expiration	2028-02-06, 2:22:44 p.m.
Notification Email	
App Federation Metadata Url	https://login.microsoftonline.com/ ...
Certificate (Base64)	Download
Certificate (Raw)	Download
Federation Metadata XML	Download

- If your relying party STS can download federation metadata from an Internet endpoint, then copy the value next to the **App Federation Metadata Url**.
- If your relying party STS requires a local XML file containing the federation metadata, then select **Download** next to **Federation Metadata XML**.
- If your relying party STS requires the certificate of the identity provider, then select **Download** next to either the **Certificate (Base64)** or **Certificate (Raw)**.
- If your relying party STS does not support federation metadata, then copy the **Login URL** and **Microsoft Entra Identifier** to configure your relying party STS.

[Set up](#)

You'll need to configure the application to link with Microsoft Entra ID.

Login URL	https://login.microsoftonline.com/ ...
Microsoft Entra Identifier	https://sts.windows.net/ ...
Logout URL	https://login.microsoftonline.com/ ...

Configure claims issued by Microsoft Entra

By default, only a few attributes from Microsoft Entra users are included in the SAML token Microsoft Entra sends to the relying party STS. You can add additional claims which your applications require, and change the attribute provided in the SAML name identifier. For more information on standard claims, see [SAML token claims reference](#).

1. In the **Attributes & Claims** box, select **Edit**.
2. To change which Entra ID attribute is sent as the value of the Name identifier, select the row **Unique User Identifier (Name ID)**. You can change the source attribute to another Microsoft Entra built-in or extension attribute. Then select **Save**.
3. To change which Entra ID attribute is sent as the value of a claim already configured, select the row in the **Additional claims** section.
4. To add a new claim, select **Add new claim**.
5. When complete, select **SAML-based Sign-on** to close this screen.

Configure who can sign-in to the application

When testing the configuration, you should assign a designated test user to the application in Microsoft Entra, to validate that the user is able to sign on to the application via Microsoft Entra and the relying party STS.

1. In the **Manage** section of the left menu, select **Properties**.
2. Ensure that the value of **Enabled for users to sign-in** is set to **Yes**.
3. Ensure that the value of **Assignment required** is set to **Yes**.
4. If you made any changes, select **Save**.
5. In the **Manage** section of the left menu, select **Users and groups**.
6. Select **Add user/group**.
7. Select **None selected**.
8. In the search box, type the name of the test user, then pick the user and select **Select**.
9. Select **Assign** to assign the user to the default **User** role of the application.
10. In the **Security** section of the left menu, select **Conditional Access**.
11. Select **What if**.
12. Select **No user or service principal selected**, select **No user selected**, and select the user previously assigned to the application.
13. Select **Any cloud app**, and select the enterprise application.
14. Select **What if**. Validate that any policies that will apply allow the user to sign into the application.

Configure Microsoft Entra as an identity provider in your relying party STS

Next, import the federation metadata into your relying party STS. The following steps are shown using AD FS, but another relying party STS could be used instead.

1. In the claims provider trust list of your relying party STS, select **Add Claims Provider Trust**, and select **Start**.
2. Depending on whether you downloaded the federation metadata from Microsoft Entra, select **Import data about the claims provider published online or on a local network**, or **Import data about the claims provider from a file**.
3. You may need to also provide the certificate of Microsoft Entra to the relying party STS.
4. When your configuration of Microsoft Entra as an identity provider is complete, confirm that:

- The claims provider identifier is a URI of the form `https://sts.windows.net/{tenantid}`.
- If using the Microsoft Entra ID global service, the endpoints for SAML single sign-on are URI of the form `https://login.microsoftonline.com/{tenantid}/saml2`. For national clouds, see [Microsoft Entra authentication & national clouds](#).
- A certificate of Microsoft Entra is recognized by the relying party STS.
- No encryption is configured.
- The claims configured in Microsoft Entra are listed as available for claims rule mappings in your relying party STS. If you subsequently added additional claims, you may need to also add them to the configuration of the identity provider in your relying party STS.

Configure claims rules in your relying party STS

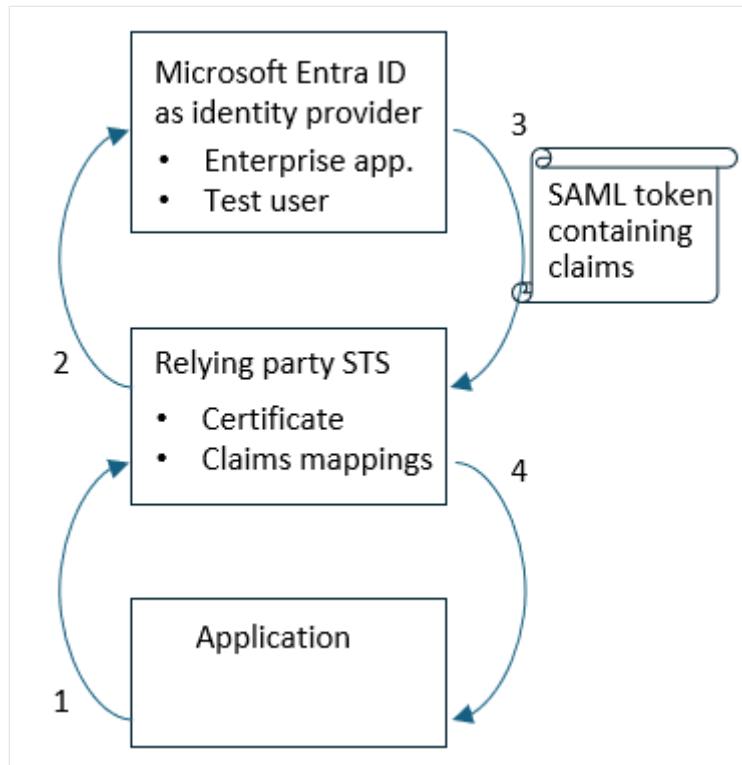
Once the claims that Microsoft Entra will send as the identity provider are known to the relying party STS, you'll need to map or transform those claims into the claims required by your application. The following steps are shown using AD FS, but another relying party STS could be used instead.

1. In the claims provider trust list of your relying party STS, select the claims provider trust for Microsoft Entra, and select **Edit Claims Rules**.
2. For each claim provided by Microsoft Entra and required by your application, select **Add Rule**. In each rule, select **Pass through or Filter an Incoming Claim**, or **Transform an Incoming Claim**, based on the requirements of your application.

Test the single sign-on to your application

After the application is configured in Microsoft Entra and your relying party STS, users can sign into it by authenticating to Microsoft Entra, and having a token provided by Microsoft Entra transformed by your relying party STS into the form and claims required by your application.

This tutorial illustrates testing the sign-in flow using a web-based application which implements the relying party initiated single sign-on pattern. For more information, see [Single sign-on SAML protocol](#).



1. In a web browser private browsing session, connect to the application and initiate the login process. The application redirects the web browser to the relying party STS, and the relying party STS determines the identity providers which can provide appropriate claims.
2. In the relying party STS, if prompted, select the Microsoft Entra identity provider. The relying party STS redirects the web browser to the Microsoft Entra login endpoint, <https://login.microsoftonline.com> if using the Microsoft Entra ID global service.
3. Sign in to Microsoft Entra using the identity of the test user, previously configured in the step [configure who can sign-in to the application](#). Microsoft Entra then locates the enterprise application based on the entity identifier, and redirect the web browser to the relying party STS reply URL endpoint, with the web browser transporting the SAML token.
4. The relying party STS validates the SAML token was issued by Microsoft Entra, then extract and transform the claims from the SAML token, and redirect the web browser to the application. Confirm that your application has received the required claims from Microsoft Entra via this process.

Complete configuration

1. After testing the initial sign-on configuration, you'll need to ensure that your relying party STS stays up to date as new certificates are added to Microsoft Entra. Some relying party STS may have a built-in process to monitor the federation metadata of the identity provider.

2. This tutorial illustrated configuring single sign-in. Your relying party STS may also support SAML single sign-out. For more information on this capability, see [Single Sign-Out SAML Protocol](#).
3. You can remove the assignment of the test user to the application. You can use other features such as dynamic groups or entitlement management to assign users to the application. For more information, see [Quickstart: Create and assign a user account](#).

Next steps

- [What is application management in Microsoft Entra ID?](#)
 - [Govern access for applications in your environment](#)
-

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Add linked single sign-on to an application

Article • 04/25/2025

This article shows you how to configure linked-based single sign-on (SSO) for your application in Microsoft Entra ID. Linked-based SSO enables Microsoft Entra ID to provide SSO to an application that is already configured for SSO in another service. The linked option lets you configure the target location when a user selects the application in your organization's My Apps or Microsoft 365 portal.

Linked-based SSO doesn't provide sign-on functionality through Microsoft Entra ID. The option simply sets the location that users are sent when they select the application on the My Apps or Microsoft 365 portal.

Some common scenarios where linked-based SSO is valuable include:

- Add a link to a custom web application that currently uses federation, such as Active Directory Federation Services (ADFS).
- Add deep links to specific web pages that you want to appear on your user's access pages.
- Add a link to an application that doesn't require authentication. The linked option doesn't provide sign-on functionality through Microsoft Entra credentials, but you can still use some of the other features of enterprise applications. For example, you can use audit logs and add a custom logo and application name.

Prerequisites

To configure linked-based SSO in your Microsoft Entra tenant, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free](#).
- One of the following roles: Cloud Application Administrator, Application Administrator, or owner of the service principal.
- An application that supports linked-based SSO.

Configure linked-based single sign-on

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Entra ID > Enterprise apps > All applications**.
3. Search for and select the application that you want to add linked SSO.
4. Select **Single sign-on** and then select **Linked**.

5. Enter the URL for the sign-in page of the application.

6. Select **Save**.

Next steps

- [Manage access to apps](#)

Add an OpenID Connect-based single sign-on application

Article • 02/27/2025

In this article, you use the Microsoft Entra admin center to add an enterprise application that uses the [OpenID Connect \(OIDC\)](#) standard for Single sign-on (SSO). After you configure SSO, your users can sign in by using their Microsoft Entra credentials.

We recommend you use a nonproduction environment to test the steps in this page.

Prerequisites

To configure OIDC-based SSO, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free ↗](#).
- One of the following roles:
 - Cloud Application Administrator
 - Application Administrator
 - owner of the service principal

Add the application from the Microsoft Entra app Gallery

When you add an enterprise application that uses the OIDC standard for SSO, you select a setup button. When you select the button, you complete the sign-up process for the application.

To configure OIDC-based SSO for an application:

1. Sign in to the [Microsoft Entra admin center ↗](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. In the **All applications** pane, select **New application**.
4. The **Browse Microsoft Entra Gallery** pane opens and displays tiles for cloud platforms, on-premises applications, and featured applications. Applications listed in the **Featured applications** section have icons indicating whether they support

federated SSO and provisioning. Search for and select the application. In this example, **SmartSheet** is being used.

5. Select **Sign-up**. Sign in with the user account credentials from Microsoft Entra ID. If you already have a subscription to the application, then user details and tenant information is validated. If the application isn't able to verify the user, then it redirects you to sign up for the application service.

The screenshot shows the Microsoft Entra admin center interface. On the left, the navigation pane is open with 'Enterprise applications' selected under 'Applications'. A search bar at the top right contains the text 'SmartSheet'. The main area displays search results for 'Showing 6 of 6 results', with the first result being 'Smartsheet' by Smartsheet.com, Inc. The 'Smartsheet' card includes a logo, a 'Name' field set to 'Smartsheet', a 'Publisher' field set to 'Smartsheet.com, Inc.', and a 'URL' field set to 'http://www.smartsheet.com'. A 'Sign up for Smartsheet' button is visible at the bottom of the card.

6. Select **Consent on behalf of your organization** and then select **Accept**. The application is added to your tenant and the application home page appears. To learn more about user and admin consent, see [Understand user and admin consent](#).

Related content

- [Configure linked single sign-on](#)
- [Configure password single sign-on](#)
- [Configure SAML-based single sign-on](#)

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Add password-based single sign-on to an application

Article • 01/31/2025

This article shows you how to set up password-based single sign-on (SSO) in Microsoft Entra ID. With password-based SSO, a user signs in to the application with a username and password the first time they sign in to it. After the first sign-on, Microsoft Entra ID sends the username and password to the application.

Password-based SSO uses the existing authentication process provided by the application. When you enable password-based SSO for an application, Microsoft Entra ID collects and securely stores usernames and passwords for the application. User credentials are stored in an encrypted state in the directory. Password-based SSO is supported for any cloud-based application that has an HTML-based sign-in page.

Choose password-based SSO when:

- An application doesn't support the Security Assertion Markup Language (SAML) SSO protocol.
- An application authenticates with a username and password instead of access tokens and headers.

The configuration page for password-based SSO is simple. It includes only the URL of the sign-on page that the application uses. This string must be the page that includes the username input field.

Prerequisites

To configure password-based SSO in your Microsoft Entra tenant, you need:

- An Azure account with an active subscription. If you don't already have one, you can [create an account for free](#)
- Application Administrator, Cloud Application Administrator, or owner of the service principal.
- An application that supports password-based SSO.

Configure password-based single sign-on

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).

2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. Enter the name of the existing application in the search box, and then select the application from the search results.
4. Select **Single sign-on** and then select **Password-based**.
5. Enter the URL for the sign-in page of the application.
6. Select **Save**.

Microsoft Entra ID parses the HTML of the sign-in page for username and password input fields. If the attempt succeeds, you're signed in. Your next step is to [Assign users or groups](#) to the application.

After assigning users and groups, you can provide credentials to be used for a user when they sign in to the application.

1. Select **Users and groups**, select the checkbox for the user's or group's row, and then select **Update Credentials**.
2. Enter the username and password to be used for the user or group. If you don't, users are prompted to enter the credentials themselves upon launch.

Manual configuration

If the parsing attempt by Microsoft Entra ID fails, you can configure sign-on manually.

1. Select **Configure {application name} Password Single Sign-on Settings** to display the **Configure sign-on** page.
2. Select **Manually detect sign-in fields**. More instructions that describe manual detection of sign-in fields appear.
3. Select **Capture sign-in fields**. A capture status page opens in a new tab, showing the message metadata capture is currently in progress.
4. If the **My Apps Extension Required** box appears in a new tab, select **Install Now** to install the My Apps Secure Sign-in Extension browser extension. (The browser extension requires Microsoft Edge or Chrome.) Then install, launch, and enable the extension, and refresh the capture status page. The browser extension then opens another tab that displays the entered URL.
5. In the tab with the entered URL, go through the sign-in process. Fill in the username and password fields, and try to sign in. (You don't have to provide the correct password.) A prompt asks you to save the captured sign-in fields.
6. Select **OK**. The browser extension updates the capture status page with the message **Metadata has been updated for the application**. The browser tab closes.
7. In the Microsoft Entra ID Configure sign-on page, select **Ok, I was able to sign-in to the app successfully**.
8. Select **OK**.

Limitations

For password-based SSO, the end user's browsers can be:

- Internet Explorer 8, 9, 10, 11--on Windows 7 or later (limited support)
- Microsoft Edge on Windows 10 Anniversary Edition or later
- Chrome--on Windows 7 or later, and on macOS X or later

Users may only have a maximum of [48 credentials](#) configured for applications utilizing password-based single sign-on.

Next steps

- [Manage access to apps](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Configure how users consent to applications

Article • 04/30/2025

In this article, you learn how to configure user consent settings in Microsoft Entra ID to control when and how users grant permissions to applications. This guidance helps IT admins reduce security risks by restricting or disabling user consent.

Before an application can access your organization's data, a user must grant the application permissions to do so. Different permissions allow different levels of access. By default, all users are allowed to consent to applications for permissions that don't require administrator consent. For example, by default, a user can consent to allow an app to access their mailbox but can't consent to allow an app unfettered access to read and write to all files in your organization.

To reduce the risk of malicious applications attempting to trick users into granting them access to your organization's data, we recommend that you allow user consent only for applications that have been published by a [verified publisher](#).

⚠ Note

Applications that require users to be assigned to the application must have their permissions consented by an administrator, even if the user consent policies for your directory would otherwise allow a user to consent on behalf of themselves.

Prerequisites

To configure user consent, you need:

- A user account. If you don't already have one, you can [create an account for free ↗](#).
- A [Privileged Role Administrator](#) role.

Configure user consent settings

You can configure user consent settings in Microsoft Entra ID using either the Microsoft Entra admin center, Microsoft Graph PowerShell, or Microsoft Graph API. The settings you configure apply to all users in your organization.

Configure user consent in Microsoft Entra admin center

To configure user consent settings through the Microsoft Entra admin center:

1. Sign in to the [Microsoft Entra admin center](#) as a **Privileged Role Administrator**.
2. Browse to **Entra ID > Enterprise apps > Consent and permissions > User consent settings**.
3. Under **User consent for applications**, select which consent setting you want to configure for all users.
4. Select **Save** to save your settings.

User consent for applications

Configure whether users are allowed to consent for applications to access your organization's data.

- Do not allow user consent
An administrator will be required for all apps.
- Allow user consent for apps from verified publishers, for selected permissions (Recommended)
All users can consent for permissions classified as "low impact", for apps from verified publishers or apps registered in this organization.
-  [7 permissions classified as low impact](#)
- Allow user consent for apps
All users can consent for any app to access the organization's data.

Tip

To allow users to request an administrator's review and approval of an application that the user isn't allowed to consent to, [enable the admin consent workflow](#). For example, you might do this when user consent has been disabled or when an application is requesting permissions that the user isn't allowed to grant.

Next steps

- [Manage app consent policies](#)
- [Configure the admin consent workflow](#)

Configure risk-based step-up consent using PowerShell

Article • 02/19/2024

In this article, you'll learn how to configure risk-based step-up consent in Microsoft Entra ID. Risk-based step-up consent helps reduce user exposure to malicious apps that make [illicit consent requests](#).

For example, consent requests for newly registered multitenant apps that are not [publisher verified](#) and require non-basic permissions are considered risky. If a risky user consent request is detected, the request requires a "step-up" to admin consent instead. This step-up capability is enabled by default, but it results in a behavior change only when user consent is enabled.

When a risky consent request is detected, the consent prompt displays a message that indicates that admin approval is needed. If the [admin consent request workflow](#) is enabled, the user can send the request to an admin for further review directly from the consent prompt. If the admin consent request workflow isn't enabled, the following message is displayed:

AADSTS90094: <clientAppDisplayName> needs permission to access resources in your organization that only an admin can grant. Request an admin to grant permission to this app before you can use it.

In this case, an audit event is also logged with a category of "ApplicationManagement," an activity type of "Consent to application," and a status reason of "Risky application detected."

Prerequisites

To configure risk-based step-up consent, you need:

- A user account. If you don't already have one, you can [create an account for free](#).
- A Global Administrator role or a Privileged Administrator role.

Disable or re-enable risk-based step-up consent

You can use the [Microsoft Graph PowerShell beta module](#) to disable the step-up to admin consent that's required in cases where a risk is detected, or to enable it if it was previously disabled.

ⓘ Important

Make sure you're using the Microsoft Graph PowerShell Beta cmdlets module.

1. Run the following command:

```
PowerShell  
  
Install-Module Microsoft.Graph.Beta
```

2. Connect to Microsoft Graph PowerShell:

```
PowerShell  
  
Connect-MgGraph -Scopes "Directory.ReadWrite.All"
```

3. Retrieve the current value for the **Consent Policy Settings** directory settings in your tenant. Doing so requires checking to see whether the directory settings for this feature have been created. If they haven't been created, use the values from the corresponding directory settings template.

```
PowerShell  
  
$consentSettingsTemplateId = "dfffd5d46-495d-40a9-8e21-954ff55e198a" #  
Consent Policy Settings  
$settings = Get-MgBetaDirectorySetting -All | Where-Object {  
  $_.TemplateId -eq $consentSettingsTemplateId }  
if (-not $settings) {  
  $params = @{  
    TemplateId = $consentSettingsTemplateId  
    Values = @(  
      @{  
        Name = "BlockUserConsentForRiskyApps"  
        Value = "True"  
      }  
      @{  
        Name =  
        "ConstrainGroupSpecificConsentToMembersOfGroupId"  
        Value = "<groupId>"  
      }  
      @{  
        Name = "EnableAdminConsentRequests"  
        Value = "True"  
      }  
    )  
  }  
  $settings = New-MgBetaDirectorySetting -TemplateId $consentSettingsTemplateId -Value $params  
}  
$settings
```

```

        }
        @{
            Name = "EnableGroupSpecificConsent"
            Value = "True"
        }
    )
}
$settings = New-MgBetaDirectorySetting -BodyParameter $params
}
$riskBasedConsentEnabledValue = $settings.Values | ? { $_.Name -eq
"BlockUserConsentForRiskyApps" }

```

4. Check the value:

PowerShell

```
$riskBasedConsentEnabledValue
```

Understand the settings value:

[+] Expand table

Setting	Type	Description
BlockUserConsentForRiskyApps	Boolean	A flag indicating whether user consent will be blocked when a risky request is detected.

5. To change the value of `BlockUserConsentForRiskyApps`, use the [Update-MgBetaDirectorySetting](#) cmdlet.

PowerShell

```

$params = @{
    TemplateId = $consentSettingsTemplateId
    Values = @(
        @{
            Name = "BlockUserConsentForRiskyApps"
            Value = "False"
        }
        @{
            Name = "ConstrainGroupSpecificConsentToMembersOfGroupId"
            Value = "<groupId>"
        }
        @{
            Name = "EnableAdminConsentRequests"
            Value = "True"
        }
        @{
            Name = "EnableGroupSpecificConsent"

```

```
        Value = "True"
    }
)
}
Update-MgBetaDirectorySetting -DirectorySettingId $settings.Id -
BodyParameter $params
```

Next steps

- Manage app consent policies
- Configure the admin consent workflow

Configure permission classifications

Article • 03/06/2025

In this article, you learn how to configure permissions classifications in Microsoft Entra ID. Permission classifications allow you to identify the impact that different permissions have based on your organization's policies and risk evaluations. For example, you can use permission classifications in consent policies to identify the set of permissions that users are allowed to consent to.

Three permission classifications are supported: "Low," "Medium" (preview), and "High" (preview). Currently, only delegated permissions that don't require admin consent can be classified.

The minimum permissions needed to do basic sign-in are `openid`, `profile`, `email`, and `offline_access`, which are all delegated permissions on the Microsoft Graph. With these permissions an app can read details of the signed-in user's profile, and can maintain this access even when the user is no longer using the app.

Prerequisites

To configure permission classifications, you need:

- An Azure account with an active subscription. [Create an account for free](#).
- One of the following roles: Application Administrator, or Cloud Application Administrator

Manage permission classifications

Follow these steps to classify permissions using the Microsoft Entra admin center:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > Consent and permissions > Permission classifications**.
3. Choose the tab for the permission classification you'd like to update.
4. Choose **Add permissions** to classify another permission.
5. Select the API and then select one or more delegated permissions.

In this example, we classify the minimum set of permission required for single sign-on:

Classify permissions

Use permission classifications in consent policies to identify the set of permissions that users are allowed to consent to. [Learn more](#)

Low Medium (Preview) High (Preview)

Define low-risk permissions here. Only delegated permissions that don't require admin consent are supported.

[+ Add permissions](#)

API used	Permissions	Description	
Microsoft Graph	offline_access	Maintain access to data you have given it access...	
Microsoft Graph	profile	View users' basic profile	
Microsoft Graph	email	View users' email address	
Microsoft Graph	openid	Sign users in	
Microsoft Graph	User.Read	Sign in and read user profile	

Related content

- [Manage app consent policies](#)
- [Permissions and consent in the Microsoft identity platform](#)

Feedback

Was this page helpful?

Yes

No

[Provide product feedback](#) ↗

Manage app consent policies

Article • 03/31/2025

App consent policies are a way to manage the permissions that apps have to access data in your organization. They're used to control what apps users can consent to and to ensure that apps meet certain criteria before they can access data. These policies help organizations maintain control over their data and ensure they only grant access to trusted apps.

In this article, you learn how to manage built-in and custom app consent policies to control when consent can be granted.

With [Microsoft Graph](#) and [Microsoft Graph PowerShell](#), you can view and manage app consent policies.

An app consent policy consists of one or more "include" condition sets and zero or more "exclude" condition sets. For an event to be considered in an app consent policy, it must match *at least* one "include" condition set, and must not match *any* "exclude" condition set.

Each condition set consists of several conditions. For an event to match a condition set, *all* conditions in the condition set must be met.

App consent policies where the ID begins with "microsoft-" are built-in policies. Some of these built-in policies are used in existing built-in directory roles. For example, the `microsoft-application-admin` app consent policy describes the conditions under which the Application Administrator and Cloud Application Administrator roles are allowed to grant tenant-wide admin consent. Built-in policies can be used in custom directory roles. They can also be used to configure user consent settings, but can't be edited or deleted.

Prerequisites

- A user or service with one of the following roles:
 - Privileged Role Administrator directory role
 - A custom directory role with the necessary [permissions to manage app consent policies](#)
 - The Microsoft Graph app role (application permission)
`Policy.ReadWrite.PermissionGrant` when connecting as an app or a service

To manage app consent policies for applications with Microsoft Graph PowerShell, connect to [Microsoft Graph PowerShell](#).

```
PowerShell
```

```
Connect-MgGraph -Scopes "Policy.ReadWrite.PermissionGrant"
```

List existing app consent policies

It's a good idea to start by getting familiar with the existing app consent policies in your organization:

1. List all app consent policies:

```
PowerShell
```

```
Get-MgPolicyPermissionGrantPolicy | ft Id, DisplayName, Description
```

2. View the "include" condition sets of a policy:

```
PowerShell
```

```
Get-MgPolicyPermissionGrantPolicyInclude -PermissionGrantPolicyId "microsoft-application-admin" | fl
```

3. View the "exclude" condition sets:

```
PowerShell
```

```
Get-MgPolicyPermissionGrantPolicyExclude -PermissionGrantPolicyId "microsoft-application-admin" | fl
```

Create a custom app consent policy using PowerShell

Follow these steps to create a custom app consent policy:

1. Create a new empty app consent policy.

```
PowerShell
```

```
New-MgPolicyPermissionGrantPolicy  
-Id "my-custom-policy"  
-DisplayName "My first custom consent policy"  
-Description "This is a sample custom app consent policy."
```

2. Add "include" condition sets.

PowerShell

```
# Include delegated permissions classified "low", for apps from
verified publishers
New-MgPolicyPermissionGrantPolicyInclude ` 
    -PermissionGrantPolicyId "my-custom-policy" ` 
    -PermissionType "delegated" ` 
    -PermissionClassification "low" ` 
    -ClientApplicationsFromVerifiedPublisherOnly
```

Repeat this step to add more "include" condition sets.

3. Optionally, add "exclude" condition sets.

PowerShell

```
# Retrieve the service principal for the Azure Management API
$azureApi = Get-MgServicePrincipal -Filter
"servicePrincipalNames/any(n:n eq 'https://management.azure.com/')"

# Exclude delegated permissions for the Azure Management API
New-MgPolicyPermissionGrantPolicyExclude ` 
    -PermissionGrantPolicyId "my-custom-policy" ` 
    -PermissionType "delegated" ` 
    -ResourceApplication $azureApi.AppId
```

Repeat this step to add more "exclude" condition sets.

After creating the app consent policy, you need to assign it to a custom role in Microsoft Entra ID. You then need to assign users to that custom role, which is attached to the app consent policy you created. For more information on how to assign the app consent policy to a custom role, see [App consent permissions for custom roles](#).

Delete a custom app consent policy using PowerShell

The following cmdlet shows how you can delete a custom app consent policy.

PowerShell

```
Remove-MgPolicyPermissionGrantPolicy -PermissionGrantPolicyId "my-custom-
policy"
```

Warning

Deleted app consent policies can't be restored. If you accidentally delete a custom app consent policy, you need to re-create the policy.

Supported conditions

The following table provides the list of supported conditions for app consent policies.

 Expand table

Condition	Description
PermissionClassification	The permission classification for the permission being granted, or "all" to match with any permission classification (including permissions that aren't classified). Default is "all."
PermissionType	The permission type of the permission being granted. Use "application" for application permissions (for example, app roles) or "delegated" for delegated permissions. Note: The value "delegatedUserConsentable" indicates delegated permissions that aren't configured by the API publisher to require admin consent. This value can be used in built-in permission grant policies, but can't be used in custom permission grant policies. Required.
ResourceApplication	The AppId of the resource application (for example, the API) for which a permission is being granted, or "any" to match with any resource application or API. Default is "any."
Permissions	The list of permission IDs for the specific permissions to match with, or a list with the single value "all" to match with any permission. Default is the single value "all." - Delegated permission IDs can be found in the OAuth2Permissions property of the API's ServicePrincipal object. - Application permission IDs can be found in the AppRoles property of the API's ServicePrincipal object.

Condition	Description
ClientApplicationIds	A list of AppId values for the client applications to match with, or a list with the single value "all" to match any client application. Default is the single value "all."
ClientApplicationTenantIds	A list of Microsoft Entra tenant IDs in which the client application is registered, or a list with the single value "all" to match with client apps registered in any tenant. Default is the single value "all."
ClientApplicationPublisherIds	A list of Microsoft Partner Network (MPN) IDs for verified publishers of the client application, or a list with the single value "all" to match with client apps from any publisher. Default is the single value "all."
ClientApplicationsFromVerifiedPublisherOnly	Set this switch to only match on client applications with a verified publishers . Disable this switch (- <code>ClientApplicationsFromVerifiedPublisherOnly:\$false</code>) to match on any client app, even if it doesn't have a verified publisher. Default is <code>\$false</code> .
scopeType	The resource scope type the preapproval applies to. Possible values: <code>group</code> for groups and teams , <code>chat</code> for chats , or <code>tenant</code> for tenant-wide access. Required.
sensitivityLabels	The sensitivity labels that are applicable to the scope type and aren't preapproved. It allows you to protect sensitive organizational data. Learn about sensitivity labels . Note: Chat resource does not support sensitivityLabels yet.

Next steps

- [Manage group owner consent policies](#)

To get help or find answers to your questions:

- [Microsoft Entra ID on Microsoft Q&A](#)

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Manage app consent policies for group owners

Article • 10/31/2023

App consent policies are a way to manage the permissions that apps have to access data in your organization. They're used to control what apps users can consent to and to ensure that apps meet certain criteria before they can access data. These policies help organizations maintain control over their data and ensure that it's being accessed only by trusted apps.

In this article, you learn how to manage built-in and custom app consent policies to control when group owner consent can be granted.

With [Microsoft Graph](#) and [Microsoft Graph PowerShell](#), you can view and manage group owner consent policies.

A group owner consent policy consists of zero or more "include" condition sets and zero or more "exclude" condition sets. For an event to be considered in a group owner consent policy, the "include" condition set must not match *any* "exclude" condition set.

Each condition set consists of several conditions. For an event to match a condition set, *all* conditions in the condition set must be met.

Group owner consent policies where the ID begins with "microsoft-" are built-in policies. For example, the `microsoft-pre-approval-apps-for-group` group owner consent policy describes the conditions under which the group owners are allowed to grant consent to applications from the preapproved list by the admin to access data for the groups they own. Built-in policies can be used in custom directory roles and to configure user consent settings, but can't be edited or deleted.

Prerequisites

- A user or service with one of the following roles:
 - Global Administrator directory role
 - Privileged Role Administrator directory role
 - A custom directory role with the necessary [permissions to manage group owner consent policies](#)
 - The Microsoft Graph app role (application permission)
Policy.ReadWrite.PermissionGrant (when connecting as an app or a service)

- To allow group owner consent subject to app consent policies, the group owner consent setting must be disabled. Once disabled, your current policy is read from the app consent policy. To learn how to disable group owner consent, see [Disable group owner consent setting](#)

To manage group owner consent policies for applications with Microsoft Graph PowerShell, connect to [Microsoft Graph PowerShell](#) and sign in with one of the roles listed in the prerequisites section. You also need to consent to the `Policy.ReadWrite.PermissionGrant` permission.

PowerShell

```
# change the profile to beta by using the `Select-MgProfile` command  
Select-MgProfile -Name "beta"
```

PowerShell

```
Connect-MgGraph -Scopes "Policy.ReadWrite.PermissionGrant"
```

Retrieve the current value for the group owner consent policy using PowerShell

Learn how to verify if your group owner consent setting has been authorized in other ways.

1. Retrieve the current value for the group owner consent setting

PowerShell

```
Get-MgPolicyAuthorizationPolicy | select -ExpandProperty  
DefaultUserRolePermissions | ft PermissionGrantPoliciesAssigned
```

If `ManagePermissionGrantPoliciesForOwnedResource` is returned in `PermissionGrantPoliciesAssigned`, your group owner consent setting might have been authorized in other ways.

2. Check if the policy is scoped to `group`.

PowerShell

```
Get-MgPolicyPermissionGrantPolicy -PermissionGrantPolicyId  
{"microsoft-all-application-permissions-for-group"} | Select -
```

ExpandProperty AdditionalProperties

If `ResourceScopeType` == `group`, your group owner consent setting has been authorized in other ways. In addition, if the app consent policy for groups has been assigned `microsoft-pre-approval-apps-for-group`, it means the preapproval feature is enabled for your tenant.

List existing group owner consent policies using PowerShell

It's a good idea to start by getting familiar with the existing group owner consent policies in your organization:

1. List all group owner consent policies:

PowerShell

```
Get-MgPolicyPermissionGrantPolicy | ft Id, DisplayName, Description
```

2. View the "include" condition sets of a policy:

PowerShell

```
Get-MgPolicyPermissionGrantPolicyInclude -PermissionGrantPolicyId
>{"microsoft-all-application-permissions-for-group"} | fl
```

3. View the "exclude" condition sets:

PowerShell

```
Get-MgPolicyPermissionGrantPolicyExclude -PermissionGrantPolicyId
>{"microsoft-all-application-permissions-for-group"} | fl
```

Create a custom group owner consent policy using PowerShell

Follow these steps to create a custom group owner consent policy:

1. Create a new empty group owner consent policy.

PowerShell

```
New-MgPolicyPermissionGrantPolicy  
  -Id "my-custom-app-consent-policy-for-group"  
  -DisplayName "My first custom app consent policy for group"  
  -Description "This is a sample custom app consent policy for group."  
  -AdditionalProperties @{includeAllPreApprovedApplications = $false;  
 resourceScopeType = "group"}
```

2. Add "include" condition sets.

PowerShell

```
# Include delegated permissions classified "low", for apps from verified publishers  
New-MgPolicyPermissionGrantPolicyInclude  
  -PermissionGrantPolicyId "my-custom-app-consent-policy-for-group"  
  -PermissionType "delegated"  
  -PermissionClassification "low"  
  -ClientApplicationsFromVerifiedPublisherOnly
```

Repeat this step to add more "include" condition sets.

3. Optionally, add "exclude" condition sets.

PowerShell

```
# Retrieve the service principal for the Azure Management API  
$azureApi = Get-MgServicePrincipal -Filter  
"servicePrincipalNames/any(n:n eq 'https://management.azure.com/')"  
  
# Exclude delegated permissions for the Azure Management API  
New-MgPolicyPermissionGrantPolicyExclude  
  -PermissionGrantPolicyId "my-custom-app-consent-policy-for-group"  
  -PermissionType "delegated"  
  -ResourceApplication $azureApi.AppId
```

Repeat this step to add more "exclude" condition sets.

Once the app consent policy for group has been created, you can [allow group owners consent](#) subject to this policy.

Delete a custom group owner consent policy using PowerShell

1. The following shows how you can delete a custom group owner consent policy.

PowerShell

```
Remove-MgPolicyPermissionGrantPolicy -PermissionGrantPolicyId "my-custom-app-consent-policy-for-group"
```

⚠️ Warning

Deleted group owner consent policies cannot be restored. If you accidentally delete a custom group owner consent policy, you will need to re-create the policy.

Supported conditions

The following table provides the list of supported conditions for group owner consent policies.

Condition	Description
PermissionClassification	The permission classification for the permission being granted, or "all" to match with any permission classification (including permissions that aren't classified). Default is "all".
PermissionType	The permission type of the permission being granted. Use "application" for application permissions (for example, app roles) or "delegated" for delegated permissions. Note: The value "delegatedUserConsentable" indicates delegated permissions that haven't been configured by the API publisher to require admin consent. This value can be used in built-in permission grant policies, but can't be used in custom permission grant policies. Required.
ResourceApplication	The AppId of the resource application (for example, the API) for which a permission is being granted, or "any" to match with any resource application or API. Default is "any".
Permissions	The list of permission IDs for the specific permissions to match with, or a list with the single value "all" to match with any permission. Default is the single value "all". - Delegated permission IDs can be found in the OAuth2Permissions property of the API's ServicePrincipal object.

Condition	Description
	<ul style="list-style-type: none"> - Application permission IDs can be found in the AppRoles property of the API's ServicePrincipal object.
ClientApplicationIds	A list of AppId values for the client applications to match with, or a list with the single value "all" to match any client application. Default is the single value "all".
ClientApplicationTenantIds	A list of Microsoft Entra tenant IDs in which the client application is registered, or a list with the single value "all" to match with client apps registered in any tenant. Default is the single value "all".
ClientApplicationPublisherIds	A list of Microsoft Partner Network (MPN) IDs for verified publishers of the client application, or a list with the single value "all" to match with client apps from any publisher. Default is the single value "all".
ClientApplicationsFromVerifiedPublisherOnly	Set this switch to only match on client applications with a verified publishers . Disable this switch (- <code>ClientApplicationsFromVerifiedPublisherOnly:\$false</code>) to match on any client app, even if it doesn't have a verified publisher. Default is <code>\$false</code> .

Warning

Deleted group owner consent policies can't be restored. If you accidentally delete a custom group owner consent policy, you will need to re-create the policy.

To get help or find answers to your questions:

- [Microsoft Entra ID on Microsoft Q&A](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#) | [Get help at Microsoft Q&A](#)

Configure the admin consent workflow

Article • 04/25/2025

In this article, you learn how to configure the admin consent workflow to enable users to request access to applications that require admin consent. You enable the ability to make requests by using an admin consent workflow. For more information on consenting to applications, see [User and admin consent](#).

The admin consent workflow gives admins a secure way to grant access to applications that require admin approval. When a user tries to access an application but is unable to provide consent, they can send a request for admin approval. The request is sent via email to admins who are designated as reviewers. A reviewer takes action on the request, and the user is notified of the action.

To approve requests, a reviewer must have the [permissions required](#) to grant admin consent for the application requested. Simply designating them as a reviewer doesn't elevate their privileges.

Prerequisites

To configure the admin consent workflow, you need:

- An Azure account. [Create an account for free](#).
- You must be a Global Administrator to turn on the admin consent workflow.

Important

Microsoft recommends that you use roles with the fewest permissions. This practice helps improve security for your organization. Global Administrator is a highly privileged role that should be limited to emergency scenarios or when you can't use an existing role.

Enable the admin consent workflow

To enable the admin consent workflow and choose reviewers:

1. Sign in to the [Microsoft Entra admin center](#) as a [Global Administrator](#).
2. Browse to **Entra ID > Enterprise apps > Consent and permissions > Admin consent settings**.

3. Under **Admin consent requests**, select Yes for **Users can request admin consent to apps they are unable to consent to**.

The screenshot shows the 'Admin consent requests' configuration page. At the top, there are 'Save' and 'Discard' buttons. Below that, a section titled 'Admin consent requests' has a sub-section 'Users can request admin consent to apps they are unable to consent to'. A 'Yes' button is selected. Another section 'Who can review admin consent requests' lists 'Reviewer type' options: 'Users', 'Groups (Preview)', and 'Roles (Preview)'. Under 'Users', it says '5 users selected.' and has '+ Add groups' and '+ Add roles' buttons. A section 'Selected users will receive email notifications for requests' has a 'Yes' button selected. Another section 'Selected users will receive request expiration reminders' also has a 'Yes' button selected. Finally, a section 'Consent request expires after (days)' shows a slider set to 30 days.

4. Configure the following settings:

- **Who can review admin consent requests** - Select users, groups, or roles that are designated as reviewers for admin consent requests. Reviewers can view, block, or deny admin consent requests, but only Global Administrators can approve admin consent requests for apps requesting for Microsoft Graph app roles (application permissions). People designated as reviewers can view incoming requests in the **My Pending** tab after they're set as reviewers. Any new reviewers aren't able to act on existing or expired admin consent requests.
- **Selected users will receive email notifications for requests** - Enable or disable email notifications to the reviewers when a request is made.
- **Selected users will receive request expiration reminders** - Enable or disable reminder email notifications to the reviewers when a request is about to expire. The first about-to-expire reminder email is likely sent out in the middle of the configured "Consent request expires after (days)." For example, if you configure the consent request to expire in three days, the first reminder email is sent out on the second day, and the last expiration email is sent out almost immediately the consent request expires.
- **Consent request expires after (days)** - Specify how long requests stay valid.

5. Select **Save**. It can take up to an hour for the workflow to become enabled.

! Note

You can add or remove reviewers for this workflow by modifying the **Who can review admin consent requests** list. A current limitation of this feature is that a reviewer retains

the ability to review requests that were made while they were designated as a reviewer and will receive expiration reminder emails for those requests after they're removed from the reviewers list. Additionally, new reviewers won't be assigned to requests that were created before they were set as a reviewer.

Configure the admin consent workflow using Microsoft Graph

To configure the admin consent workflow programmatically, use the [Update adminConsentRequestPolicy](#) API in Microsoft Graph.

Next steps

[Grant tenant-wide admin consent to an application](#)

[Review admin consent requests](#)

Review and take action on admin consent requests

Article • 04/08/2025

In this article, you learn how to review and take action on admin consent requests. To review and act on consent requests, you must be designated as a reviewer. For more information, check out the [Configure the admin consent workflow](#) article. As a reviewer, you can view all admin consent requests but you can only act on those requests that were created after you were designated as a reviewer.

When reviewing admin consent requests, you have several options to choose from:

- **Review:** This option allows administrators to evaluate the request and grant consent if deemed appropriate.
- **Deny:** Selecting this option will reject the request for consent, preventing the application from accessing the requested permissions. This action does not provide feedback to the user who made the request.
- **Block:** This option not only denies the current request but also prevents future requests for the same application from being submitted. This is useful for applications that are deemed untrustworthy or unnecessary for the organization.

For instance, if an application is found to be non-compliant with company policies, an administrator might choose to 'Block' it. Conversely, if an application is legitimate but requires further review, the administrator may opt to 'Deny' the request temporarily while seeking more information.

Prerequisites

To review and take action on admin consent requests, you need:

- An Azure account. [Create an account for free](#).
- An administrator role or a designated reviewer with the appropriate role to [review admin consent requests](#).

Review and take action on admin consent requests

To review the admin consent requests and take action:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#) who is a designated reviewer.
2. Browse to **Identity > Applications > Enterprise applications**.

3. Under **Activity**, select **Admin consent requests**.
4. Select **My Pending** tab to view and act on the pending requests.
5. Select the application that is being requested from the list.
6. Review details about the request:

- To see what permissions are being requested by the application, select **Review permissions and consent**.
- To view the application details, select the **App details** tab.
- To see who is requesting access and why, select the **Requested by** tab.

7. Evaluate the request and take the appropriate action:

- **Approve the request.** To approve a request, grant admin consent to the application. Once a request is approved, all requestors are notified that their request for access is granted. Approving a request allows all users in your tenant to access the application unless otherwise restricted with user assignment.
- **Deny the request.** To deny a request, you must provide a justification that is provided to all requestors. Once a request is denied, all requestors are notified that their request for access is denied. Denying a request won't prevent users from requesting admin consent to the application again in the future.
- **Block the request.** To block a request, you must provide a justification that is provided to all requestors. Once a request is blocked, all requestors are notified that their request to access the application is denied. Blocking a request creates a service principal object for the application in your tenant in a disabled state. Users won't be able to request admin consent to the application in the future.

Review admin consent requests using Microsoft Graph

To review the admin consent requests programmatically, use the [appConsentRequest resource type](#) and [userConsentRequest resource type](#) and their associated methods in Microsoft Graph. You can't approve or deny consent requests using Microsoft Graph.

Related content

- [Review permissions granted to apps](#)
- [Grant tenant-wide admin consent](#)

Grant tenant-wide admin consent to an application

Article • 11/29/2024

In this article, you learn how to grant tenant-wide admin consent to an application in Microsoft Entra ID. To understand how to configure individual user consent settings, see [Configure how end-users consent to applications](#).

When you grant tenant-wide admin consent to an application, you give the application access to the permissions requested on behalf of the whole organization. Granting admin consent on behalf of an organization is a sensitive operation, potentially allowing the application's publisher access to significant portions of your organization's data, or the permission to do highly privileged operations. Examples of such operations might be role management, full access to all mailboxes or all sites, and full user impersonation. Therefore you need to carefully review the permissions that the application is requesting before you grant consent.

By default, granting tenant-wide admin consent to an application allows all users to access the application unless otherwise restricted. To restrict which users can sign-in to an application, configure the app to [require user assignment](#) and then [assign users or groups to the application](#).

Important

Granting tenant-wide admin consent may revoke permissions that have already been granted tenant-wide for that application. Permissions that users have already granted on their own behalf aren't affected.

Prerequisites

Granting tenant-wide admin consent requires you to sign in as a user that is authorized to consent on behalf of the organization.

To grant tenant-wide admin consent, you need:

- A Microsoft Entra user account with one of the following roles:
 - Privileged Role Administrator, for granting consent for apps requesting any permission, for any API.

- Cloud Application Administrator or Application Administrator, for granting consent for apps requesting any permission for any API, *except* Microsoft Graph app roles (application permissions).
- A custom directory role that includes the [permission to grant permissions to applications](#), for the permissions required by the application.

Grant tenant-wide admin consent in Enterprise apps pane

You can grant tenant-wide admin consent through the **Enterprise applications** pane if the application is already provisioned in your tenant. For example, an app could be provisioned in your tenant if at least one user consents to the application. For more information, see [How and why applications are added to Microsoft Entra ID](#).

 **Tip**

Steps in this article might vary slightly based on the portal you start from.

To grant tenant-wide admin consent to an app listed in **Enterprise applications** pane:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. Enter the name of the existing application in the search box, and then select the application from the search results.

4. Select Permissions under Security.

The screenshot shows the Microsoft Entra SAML Toolkit 1 | Permissions page. On the left, there's a sidebar with 'Manage' and 'Security' sections. Under 'Manage', 'Properties', 'Owners', 'Roles and administrators', 'Users and groups', 'Single sign-on', 'Provisioning', 'Self-service', and 'Custom security attributes' are listed. Under 'Security', 'Conditional Access', 'Permissions' (which is selected), 'Token encryption', 'Activity', 'Sign-in logs', and 'Usage & insights' are listed. The main content area is titled 'Permissions'. It says, 'Below is the list of permissions that have been granted for your organization. As an administrator, you can grant permissions to this app on behalf of all users (delegated permissions). You can also grant permissions directly to this app (app permissions). [Learn more](#)'. Below this, it says, 'You can review, revoke, and restore permissions. [Learn more](#)'. Then it says, 'To configure requested permissions for apps you own, use the [app registration](#)'. A red box highlights the 'Grant admin consent for Contoso' button. Below the button, there are two tabs: 'Admin consent' (which is selected) and 'User consent'. A search bar labeled 'Search permissions' is present. A table lists API permissions:

API Name	Claim value	Permission	Type	Granted
Microsoft Graph	offline_access	Maintain access to dat...	Delegated	Admin co...
Microsoft Graph	openid	Sign users in	Delegated	Admin co...
Microsoft Graph	Application.Read.All	Read all applications	Application	Admin co...

5. Carefully review the permissions that the application requires. If you agree with the permissions the application requires, select **Grant admin consent**.

Grant admin consent in App registrations pane

You can grant tenant-wide admin consent from **App registrations** in the Microsoft Entra admin center for applications your organization develops and registers directly in your Microsoft Entra tenant.

To grant tenant-wide admin consent from **App registrations**:

1. On the Microsoft Entra admin center, browse to **Identity > Applications > App registrations > All applications**.
2. Enter the name of the existing application in the search box, and then select the application from the search results.
3. Select **API permissions** under **Manage**.
4. Carefully review the permissions that the application requires. If you agree, select **Grant admin consent**.

Construct the URL for granting tenant-wide admin consent

When you grant tenant-wide admin consent using either method described in the previous section, a window opens from the Microsoft Entra admin center to prompt for tenant-wide admin consent. If you know the client ID (also known as the application ID) of the application, you can build the same URL to grant tenant-wide admin consent.

The tenant-wide admin consent URL follows the following format:

HTTP

```
https://login.microsoftonline.com/{organization}/adminconsent?client_id={client-id}
```

Where:

- {client-id} is the application's client ID (also known as app ID).
- {organization} is the tenant ID or any verified domain name of the tenant you want to consent the application in. You can use the value organizations that causes the consent to happen in the home tenant of the user you sign in with.

As always, carefully review the permissions an application requests before granting consent.

For more information on constructing the tenant-wide admin consent URL, see [Admin consent on the Microsoft identity platform](#).

Next steps

- [Configure how end-users consent to applications](#).
- [Configure the admin consent workflow](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Grant consent on behalf of a single user by using PowerShell

Article • 12/12/2024

In this article, you learn how to grant consent on behalf of a single user by using PowerShell.

When a user grants consent for themselves, the following events occur more often:

1. A service principal for the client application is created, if it doesn't already exist. A service principal is the instance of an application or a service in your Microsoft Entra tenant. Access granted to the app or service is associated with this service principal object.
2. For each API to which the application requires access, a delegated permission grant to that API is created for the permissions that the application needs. The access is granted on behalf of the user. A delegated permission grant authorizes an application to access an API on behalf of a user, when that user signs in.
3. The user is assigned the client application. Assigning the application to the user ensures that the application is listed in the [My Apps](#) portal for that user. The user can review and revoke the access that granted on their behalf from their My Apps portal.

Prerequisites

To grant consent to an application on behalf of one user, you need:

- A user account with a Privileged Role Administrator, Application Administrator, or Cloud Application Administrator

Grant consent on behalf of a single user

Before you start, record the following details from the Microsoft Entra admin center:

- The app ID for the app that you're granting consent. For purposes of this article, we call it the client application.
- The API permissions that the client application requires. Find out the app ID of the API and the permission IDs or claim values.
- The username or object ID for the user on whose behalf access is granted.

For this example, we use [Microsoft Graph PowerShell](#) to grant consent on behalf of a single user. The client application is [Microsoft Graph Explorer](#), and we grant access to the Microsoft Graph API.

To grant consent to an application on behalf of one user using Microsoft Graph PowerShell, you need to sign in as at least a [Cloud Application Administrator](#).

PowerShell

```
# The app for which consent is being granted. In this example, we're
# granting access
# to Microsoft Graph Explorer, an application published by Microsoft.
$clientAppId = "de8bc8b5-d9f9-48b1-a8ad-b748da725064" # Microsoft Graph
Explorer

# The API to which access will be granted. Microsoft Graph Explorer makes
# API
# requests to the Microsoft Graph API, so we'll use that here.
$resourceAppId = "00000003-0000-0000-c000-000000000000" # Microsoft Graph
API

# The permissions to grant. Here we're including "openid", "profile",
#"User.Read"
# and "offline_access" (for basic sign-in), as well as "User.ReadBasic.All"
# (for
# reading other users' basic profile).
$permissions = @("openid", "profile", "offline_access", "User.Read",
"User.ReadBasic.All")

# The user on behalf of whom access will be granted. The app will be able to
access
# the API on behalf of this user.
$userUpnOrId = "user@example.com"

# Step 0. Connect to Microsoft Graph PowerShell. We need User.ReadBasic.All
to get
#   users' IDs, Application.ReadWrite.All to list and create service
principals,
#   DelegatedPermissionGrant.ReadWrite.All to create delegated permission
grants,
#   and AppRoleAssignment.ReadWrite.All to assign an app role.
#   WARNING: These are high-privilege permissions!
Connect-MgGraph -Scopes ("User.ReadBasic.All Application.ReadWrite.All "
+ "DelegatedPermissionGrant.ReadWrite.All "
+ "AppRoleAssignment.ReadWrite.All")

# Step 1. Check if a service principal exists for the client application.
#   If one doesn't exist, create it.
$clientSp = Get-MgServicePrincipal -Filter "appId eq '$($clientAppId)'"
if (-not $clientSp) {
    $clientSp = New-MgServicePrincipal -AppId $clientAppId
}
```

```

# Step 2. Create a delegated permission that grants the client app access to
the
#     API, on behalf of the user. (This example assumes that an existing
delegated
#     permission grant does not already exist, in which case it would be
necessary
#     to update the existing grant, rather than create a new one.)
$user = Get-MgUser -UserId $userUpnOrId
$resourceSp = Get-MgServicePrincipal -Filter "appId eq '$($resourceAppId)'"
$scopeToGrant = $permissions -join " "
$grant = New-MgOAuth2PermissionGrant -ResourceId $resourceSp.Id `

                                         -Scope $scopeToGrant `

                                         -ClientId $clientSp.Id `

                                         -ConsentType "Principal" `

                                         -PrincipalId $user.Id

# Step 3. Assign the app to the user. This ensures that the user can sign in
if assignment
#     is required, and ensures that the app shows up under the user's My
Apps portal.
if ($clientSp.AppRoles | ? { $_.AllowedMemberTypes -contains "User" }) {
    Write-Warning ("A default app role assignment cannot be created because
the " `

                  + "client application exposes user-assignable app roles.

You must " `

                  + "assign the user a specific app role for the app to be
listed " `

                  + "in the user's My Apps portal.")
} else {
    # The app role ID 00000000-0000-0000-0000-000000000000 is the default
app role
    # indicating that the app is assigned to the user, but not for any
specific
    # app role.
    $assignment = New-MgServicePrincipalAppRoleAssignedTo `

                                         -ServicePrincipalId $clientSp.Id `

                                         -ResourceId $clientSp.Id `

                                         -PrincipalId $user.Id `

                                         -AppRoleId "00000000-0000-0000-0000-000000000000"
}

```

Next steps

- Configure the admin consent workflow
- Configure how users consent to applications
- Permissions and consent in the Microsoft identity platform

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Review permissions granted to enterprise applications

Article • 03/06/2025

In this article, you learn how to review permissions granted to applications in your Microsoft Entra tenant. You might need to review permissions when you detect a malicious application, or one that has more permissions than is necessary. You learn how to revoke permissions granted to the application using Microsoft Graph API and existing versions of PowerShell.

The steps in this article apply to all applications that were added to your Microsoft Entra tenant via user or admin consent. For more information on consenting to applications, see [User and admin consent](#).

Prerequisites

To review permissions granted to applications, you need:

- A Microsoft Entra account with an active subscription. [Create an account for free ↗](#).
- One of the following roles:
 - Cloud Application Administrator
 - Application Administrator.
 - A Service principal owner who isn't an administrator is able to invalidate refresh tokens.

Review and revoke permissions in the Microsoft Entra admin center

You can access the Microsoft Entra admin center to view the permissions granted to an app. You can revoke permissions granted by admins for your entire organization, and you can get contextual PowerShell scripts to perform other actions.

For information on how to restore revoked or deleted permissions, see [Restore permissions granted to applications](#).

To review an application's permissions granted for the entire organization or to a specific user or group:

1. Sign in to the [Microsoft Entra admin center](#) ↗ as at least a [Cloud Application Administrator](#).

2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. Select the application that you want to restrict access to.
4. Select **Permissions**.
5. To view permissions that apply to your entire organization, select the **Admin consent** tab. To view permissions granted to a specific user or group, select the **User consent** tab.
6. To view the details of a given permission, select the permission from the list. The **Permission Details** pane opens. After reviewing the permissions granted to an application, you can revoke permissions granted by admins for your entire organization.

! Note

You can't revoke permissions in the **User consent** tab using the portal. You can revoke these permissions using Microsoft Graph API calls or PowerShell cmdlets. Go to the PowerShell and Microsoft Graph tabs of this article for more information.

To revoke permissions in the **Admin consent** tab:

1. View the list of permissions in the **Admin consent** tab.
2. Choose the permission you would like to revoke, then select the ... control for that permission.

Admin consent		User consent	
		<input type="text"/> Search permissions	
API Name	Claim value	Permission	Type
Microsoft Graph			
Microsoft Graph	offline_access	Maintain access to dat...	Delegated
Microsoft Graph	email	View users' email addr...	Delegated
			Admin consent
			Revoke Permission
			An administrator
			...

3. Select **Revoke permission**.

! Note

Revoking the current granted permission doesn't stop users from re-consenting to the application's requested permissions. You need to [stop the application from requesting the permissions through dynamic consent](#). If you want to block users from consenting altogether, read [Configure how users consent to applications](#).

Other authorization to consider

Delegated and application permissions aren't the only ways to grant applications and users access to protected resources. Admins should be aware of other authorization systems that might grant access to sensitive information. Examples of various authorization systems at Microsoft include [Microsoft Entra built-in roles](#), [Exchange RBAC](#), and [Teams resource-specific consent](#).

Related content

- [Configure user consent setting](#)
 - [Configure admin consent workflow](#)
 - [Restore revoked permissions](#)
-

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Restore revoked permissions granted to applications

Article • 10/23/2023

In this article, you learn how to restore previously revoked permissions that were granted to an application. You can restore permissions for an application that was granted permissions to access your organization's data. You can also restore permissions for an application that was granted permissions to act as a user.

Currently, restoring permissions is only possible through Microsoft Graph PowerShell and Microsoft Graph API calls. You can't restore permissions through the Microsoft Entra admin center. In this article, you learn how to restore permissions using Microsoft Graph PowerShell.

Prerequisites

To restore previously revoked permissions for an application, you need:

- An Azure account with an active subscription. [Create an account for free](#).
- One of the following roles: Global Administrator, Cloud Application Administrator, Application Administrator.
- A Service principal owner who isn't an administrator is able to invalidate refresh tokens.

Restore revoked permissions for an application

You can try different methods for restoring permissions:

- Use the **Grant admin consent** button on the **Permissions** page for the app to apply consent again. This consent applies the set of permissions that the app's developer originally requested in the app manifest.

Note

Regranting admin consent will remove any granted permissions that are not part of the default set configured by the developer.

- If you know the specific permission that was revoked, you can grant it again manually using [PowerShell](#) or the [Microsoft Graph API](#).

- If you don't know the revoked permissions, you can use the scripts provided in this article to detect and restore revoked permissions.

First, set the `servicePrincipalId` value in the script to the ID value for the enterprise app whose permissions you want to restore. This ID is also called the `object ID` in the Microsoft Entra admin center [Enterprise applications](#) page.

Then, run each script with `$ForceGrantUpdate = $false` in order to see a list of delegated or app-only permissions that maybe have been removed. Even if the permissions have already been restored, revoke events from your audit logs may still appear in the script results.

Leave `$ForceGrantUpdate` set to `$true` if you want the script to attempt to restore any revoked permissions it detects. The scripts ask for confirmation, but don't ask for individual approval for each permission that it restores.

Be cautious when granting permissions to apps. To learn more on how to evaluate permissions, see [Evaluate permissions](#).

Restore delegated permissions

PowerShell

```
# WARNING: Setting $ForceGrantUpdate to true will modify permission grants
without
# prompting for confirmation. This can result in unintended changes to your
# application's security settings. Use with caution!
$ForceGrantUpdate = $false

# Set the start and end dates for the audit log search
# If setting date use yyyy-MM-dd format
# endDate is set to tomorrow to include today's audit logs
$startDate = (Get-Date).AddDays(-7).ToString('yyyy-MM-dd')
$endDate = (Get-Date).AddDays(1).ToString('yyyy-MM-dd')

# Set the service principal ID
$servicePrincipalId = "efe87e5d-05cb-4b19-9b36-1eb923448697"

Write-Host "Searching for audit logs between $startDate and $endDate" -
ForegroundColor Green
Write-Host "Searching for audit logs for service principal
$servicePrincipalId" -ForegroundColor Green

if ($ForceGrantUpdate -eq $true) {
    Write-Host "WARNING: ForceGrantUpdate is set to true. This will modify
permission grants without prompting for confirmation. This can result in
unintended changes to your application's security settings. Use with
caution!" -ForegroundColor Red
```

```

$continue = Read-Host "Do you want to continue? (Y/N)"
if ($continue -eq "Y" -or $continue -eq "y") {
    Write-Host "Continuing..."
} else {
    Write-Host "Exiting..."
    exit
}
}

# Connect to MS Graph
Connect-MgGraph -Scopes
"AuditLog.Read.All","DelegatedPermissionGrant.ReadWrite.All" -ErrorAction
Stop | Out-Null

# Create a hashtable to store the OAuth2PermissionGrants
$oAuth2PermissionGrants = @{}

function Merge-Scopes($oldScopes, $newScopes) {
    $oldScopes = $oldScopes.Trim() -split '\s+'
    $newScopes = $newScopes.Trim() -split '\s+'
    $mergedScopesArray = $oldScopes + $newScopes | Select-Object -Unique
    $mergedScopes = $mergedScopesArray -join ' '
    return $mergedScopes.Trim()
}

# Function to merge scopes if multiple OAuth2PermissionGrants are found in
# the audit logs
function Add-Scopes($resourceId, $newScopes) {
    if($oAuth2PermissionGrants.ContainsKey($resourceId)) {
        $oldScopes = $oAuth2PermissionGrants[$resourceId]
        $oAuth2PermissionGrants[$resourceId] = Merge-Scopes $oldScopes
    }
    else {
        $oAuth2PermissionGrants[$resourceId] = $newScopes
    }
}

function Get-ScopeDifference ($generatedScope, $currentScope) {
    $generatedScopeArray = $generatedScope.Trim() -split '\s+'
    $currentScopeArray = $currentScope.Trim() -split '\s+'
    $difference = $generatedScopeArray | Where-Object { $_ -notin
    $currentScopeArray }
    $difference = $difference -join ' '
    return $difference.Trim()
}

# Set the filter for the audit log search
$filterOAuth2PermissionGrant = "activityDateTime ge $startDate and
activityDateTime le $endDate" +
    " and Result eq 'success'" +
    " and ActivityDisplayName eq 'Remove delegated permission grant'" +
    " and targetResources/any(x: x/id eq '$servicePrincipalId')"
try {
    # Retrieve the audit logs for removed OAuth2PermissionGrants

```

```

    $oAuth2PermissionGrantsAuditLogs = Get-MgAuditLogDirectoryAudit -Filter
$filterOAuth2PermissionGrant -All -ErrorAction Stop
}
catch {
    Disconnect-MgGraph | Out-Null
    throw $_
}

# Remove User Delegated Permission Grants
$oAuth2PermissionGrantsAuditLogs = $oAuth2PermissionGrantsAuditLogs | Where-
Object {
    -not ($_.TargetResources.ModifiedProperties.OldValue -eq '"Principal"')
}

# Merge duplicate OAuth2PermissionGrants from AuditLogs using Add-Scopes
foreach ($auditLog in $oAuth2PermissionGrantsAuditLogs) {
    $resourceId = $auditLog.TargetResources[0].Id
    # We only want to process OAuth2PermissionGrant Audit Logs where
    $servicePrincipalId is the clientId not the resourceId
    if ($resourceId -eq $servicePrincipalId) {
        continue
    }
    $oldScope = $auditLog.TargetResources[0].ModifiedProperties | Where-
Object { $_.DisplayName -eq "DelegatedPermissionGrant.Scope" } | Select-
Object -ExpandProperty OldValue
    if ($oldScope -eq $null) {
        $oldScope = ""
    }
    $oldScope = $oldScope.Replace("'", '')
    $newScope = $auditLog.TargetResources[0].ModifiedProperties | Where-
Object { $_.DisplayName -eq "DelegatedPermissionGrant.Scope" } | Select-
Object -ExpandProperty NewValue
    if ($newScope -eq $null) {
        $newScope = ""
    }
    $newScope = $newScope.Replace("'", '')
    $scope = Merge-Scopes $oldScope $newScope
    Add-Scopes $resourceId $scope
}

$permissionCount = 0
foreach ($resourceId in $oAuth2PermissionGrants.keys) {
    $scope = $oAuth2PermissionGrants[$resourceId]
    $params = @{
        clientId = $servicePrincipalId
        consentType = "AllPrincipals"
        resourceId = $resourceId
        scope = $scope
    }

    try {
        $currentOAuth2PermissionGrant = Get-MgOAuth2PermissionGrant -Filter
"clientId eq '$servicePrincipalId' and consentType eq 'AllPrincipals' and
resourceId eq '$resourceId'" -ErrorAction Stop
        $action = "Creating"
    }
}
```

```

    if ($currentOAuth2PermissionGrant -ne $null) {
        $action = "Updating"
    }
    Write-Host -----
    if ($ForceGrantUpdate -eq $true) {
        Write-Host "$action OAuth2PermissionGrant with the following
parameters:"
    } else {
        Write-Host "Potentially removed OAuth2PermissionGrant scopes
with the following parameters:"
    }
    Write-Host "    clientId: $($params.clientId)"
    Write-Host "    consentType: $($params.consentType)"
    Write-Host "    resourceId: $($params.resourceId)"
    if ($currentOAuth2PermissionGrant -ne $null) {
        $scopeDifference = Get-ScopeDifference $scope
    }
    $currentOAuth2PermissionGrant.Scope
    if ($scopeDifference -eq "") {
        Write-Host "OAuth2PermissionGrant already exists with the
same scope" -ForegroundColor Yellow
        if ($ForceGrantUpdate -eq $true) {
            Write-Host "Skipping Update" -ForegroundColor Yellow
        }
        continue
    }
    else {
        Write-Host "    scope diff: '$scopeDifference'"
    }
}
else {
    Write-Host "    scope: '$($params.scope)'"
}
if ($ForceGrantUpdate -eq $true -and $currentOAuth2PermissionGrant -
eq $null) {
    New-MgOAuth2PermissionGrant -BodyParameter $params -ErrorAction
Stop | Out-Null
    Write-Host "OAuth2PermissionGrant was created successfully" -
ForegroundColor Green
}
if ($ForceGrantUpdate -eq $true -and $currentOAuth2PermissionGrant -
ne $null) {
    Write-Host "    Current Scope:
'$($currentOAuth2PermissionGrant.scope)' " -ForegroundColor Yellow
    Write-Host "    Merging with scopes from audit logs" -
ForegroundColor Yellow
    $params.scope = Merge-Scopes $currentOAuth2PermissionGrant.scope
$params.scope
    Write-Host "    New Scope: '$($params.scope)' " -ForegroundColor
Yellow
    Update-MgOAuth2PermissionGrant -OAuth2PermissionGrantId
$currentOAuth2PermissionGrant.id -BodyParameter $params -ErrorAction Stop |
Out-Null
    Write-Host "OAuth2PermissionGrant was updated successfully" -
ForegroundColor Green
}

```

```
$permissionCount++  
}  
catch {  
    Disconnect-MgGraph | Out-Null  
    throw $_  
}  
}  
  
Disconnect-MgGraph | Out-Null  
  
if ($ForceGrantUpdate -eq $true) {  
    Write-Host "-----"  
    Write-Host "$permissionCount OAuth2PermissionGrants were created/updated  
successfully" -ForegroundColor Green  
} else {  
    Write-Host "-----"  
    Write-Host "$permissionCount OAuth2PermissionGrants were found" -  
ForegroundColor Green  
}
```

View activity logs of application permissions

Article • 04/28/2025

Microsoft Entra is a platform that allows you to create and manage applications for your organization. You can grant different permissions to your applications, such as accessing data, or performing actions. It's important to review these permissions periodically to ensure they remain appropriate and secure.

One way to review permissions granted to your apps is by using activity logs, which record the activities and events that occur in your Microsoft Entra applications. Activity logs help you to monitor the usage and performance of your applications, and to identify any potential issues or risks. By reviewing the activity logs, you can see what permissions your applications have and whether they're complying with your policies and expectations.

In this article, you:

- View activity logs to see API permission granting and removing activity for a specific application.
- View activity logs to see API permission granting and removing activity for all applications.
- Understand which audit logs are used to track granting and removing API permissions from app to app.

Prerequisites

To view Activity Logs for applications, you need:

- A user account. If you don't already have one, you can [create an account for free ↗](#).
- One of the following roles: Reports Reader, Security Reader, Security Administrator, Global Reader

How to view permission audit logs for all applications in your directory

Only certain events recorded in the Activity Logs are needed to see application permission activity. To view all events using the Microsoft Entra admin center, Take the following steps:

1. Sign in to the [Microsoft Entra admin center ↗](#) with at least a [Reports reader](#) role
2. Browse to **Entra ID > Enterprise apps**.

3. In the left-hand navigation underneath **Activity**, browse to **Audit logs**.
4. Filter the audit logs by using the information included in the **Audit logs** section to select only the needed logs to view permission activity for your applications.
5. Use **Manage view** on the top command bar to edit the columns shown. Select the **Date** column to view more detailed information per audit log.

How to view permission audit logs for a specific resource application

It can be helpful to deep dive into the activity of a resource application to see which applications already have access through API permissions. For example, you might want to monitor the activity logs for the Microsoft Graph application, so you can see when permissions are granted for the resources it protects.

To view the activity logs for a resource application:

1. Sign in to the [Microsoft Entra admin center](#) with at least a **Reports reader** role
2. Browse to **Entra ID > Enterprise apps**.
3. Search for the resource application that owns the permission. For example, if you want to view which applications were awarded the Microsoft Graph `Mail.Read` permission in the last 30 days, search for *Microsoft Graph*.
4. In the left-hand navigation underneath **Activity**, browse to **Audit logs**.
5. Filter the audit logs by using the information included in the **Audit logs** section to select only the needed logs to view permission activity for your applications.
6. Use **Manage view** on the top command bar to edit the columns shown. Select the **Date** column to view more detailed information per audit log.

Audit logs

The following table outlines the scenarios and audit values available for the granting and revoking of permissions granted to apps.

 Expand table

Scenario	Audit Service	Audit Category	Audit Activity	Audit Actor	Audit log limitations
Granting app-only access to an app	Core Directory	ApplicationManagement	Add app role assignment to the service principal	User context	
Revoking app-only access to an app	Core Directory	ApplicationManagement	Remove app role assignment from the service principal	User context	
Granting delegated access to an app	Core Directory	ApplicationManagement	Add delegated permission grant	User context	
User grants consent to an application	Core Directory	ApplicationManagement	Consent to application	User context	

Related content

- [How to access Activity Logs through Microsoft Graph and PowerShell](#)
- [Microsoft Entra audit log categories and activities](#)

Manage users and groups assignment to an application

Article • 04/14/2025

This article shows you how to assign users and groups to an enterprise application in Microsoft Entra ID. When you assign a user to an application, the application appears in the user's [My Apps](#) portal for easy access. If the application exposes app roles, you can also assign a specific app role to the user.

When you assign a group to an application, only users in the group have access. The assignment doesn't cascade to nested groups.

Group-based assignment requires Microsoft Entra ID P1 or P2 edition. Nested group memberships aren't currently supported. For more licensing requirements for the features discussed in this article, see the [Microsoft Entra pricing page](#).

For greater control, certain types of enterprise applications can be configured to require user assignment. For more information on requiring user assignment for an app, see [Manage access to an application](#). Applications that require users to be assigned to the application must have their permissions consented by an administrator, even if the user consent policies for your directory would otherwise allow a user to consent on behalf of themselves.

(!) Note

If you encounter limitations when managing groups through the portal, such as with application access policy groups, consider using alternative methods like [PowerShell](#) or [Microsoft Graph API](#).

Prerequisites

To assign users to an enterprise application, you need:

- A Microsoft Entra account with an active subscription. If you don't already have one, you can [Create an account for free](#).
- One of the following roles:
 - Cloud Application Administrator
 - Application Administrator
 - User Administrator
 - Owner of the service principal.

- Microsoft Entra ID P1 or P2 for group-based assignment. For more licensing requirements for the features discussed in this article, see the [Microsoft Entra pricing page](#).

Assign users and groups to an application using the Microsoft Entra admin center

To assign a user or group account to an enterprise application:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. Enter the name of the existing application in the search box, and then select the application from the search results.
4. Select **Users and groups**, and then select **Add user/group**.

The screenshot shows the Microsoft Entra admin center interface. On the left, there's a navigation sidebar with categories like Home, Favorites, Identity, Overview, Users, Groups, Devices, Applications (with sub-options like Enterprise applications, App registrations), Protection, Identity governance, External Identities, and Show more. Under Applications, the 'Enterprise applications' option is selected. The main content area shows the details for 'Microsoft Entra SAML Toolkit 1'. The top navigation bar includes a search bar, a back button, and various icons. Below the title, there are tabs for Overview, Deployment Plan, Diagnose and solve problems, Manage (Properties, Owners, Roles and administrators), and the highlighted 'Users and groups'. A red box highlights the 'Users and groups' tab. To its right, there's a button labeled '+ Add user/group' and other actions like Edit assignment, Remove, Update credentials, Columns, and Got feedback?.

Display Name	Object Type	Role assigned
No application assignments found		

5. On the **Add Assignment** pane, select **None Selected** under **Users and groups**.
6. Search for and select the user or group that you want to assign to the application. For example, `contosouser1@contoso.com` or `contosoteam1@contoso.com`.
7. Select **Select**.
8. Under **Select a role**, select the role that you want to assign to the user or group. If you haven't defined any roles yet, the default role is **Default Access**.

9. On the **Add Assignment** pane, select **Assign** to assign the user or group to the application.

Unassign users, and groups, from an application

1. Follow the steps on the [Assign users, and groups, to an application](#) section to navigate to the **Users and groups** pane.
2. Search for and select the user or group that you want to unassign from the application.
3. Select **Remove** to unassign the user or group from the application.

Related content

- [Assign custom security attributes](#)
- [Disable user sign-in](#).

Manage custom security attributes for an application

Article • 03/06/2025

Custom security attributes in Microsoft Entra ID are business-specific attributes (key-value pairs) that you can define and assign to Microsoft Entra objects. For example, you can assign custom security attribute to filter your applications or to help determine who gets access. This article describes how to assign, update, list, or remove custom security attributes for Microsoft Entra enterprise applications.

Prerequisites

To assign or remove custom security attributes for an application in your Microsoft Entra tenant, you need:

- A Microsoft Entra account with an active subscription. [Create an account for free](#).
- **Attribute Assignment Administrator** role.
- Make sure you have existing custom security attributes. To learn how to create a security attribute, see [Add or deactivate custom security attributes in Microsoft Entra ID](#).

 **Important**

By default, [Global Administrator](#) and other administrator roles do not have permissions to read, define, or assign custom security attributes.

Assign, update, list, or remove custom attributes for an application

Learn how to work with custom attributes for applications in Microsoft Entra ID.

Assign custom security attributes to an application

Undertake the following steps to assign custom security attributes through the Microsoft Entra admin center.

1. Sign in to the [Microsoft Entra admin center](#) as an [Attribute Assignment Administrator](#).

2. Browse to **Identity > Applications > Enterprise applications**.
3. Find and select the application you want to add a custom security attribute to.
4. In the Manage section, select **Custom security attributes**.
5. Select **Add assignment**.
6. In **Attribute set**, select an attribute set from the list.
7. In **Attribute name**, select a custom security attribute from the list.
8. Depending on the properties of the selected custom security attribute, you can enter a single value, select a value from a predefined list, or add multiple values.
 - For freeform, single-valued custom security attributes, enter a value in the **Assigned values** box.
 - For predefined custom security attribute values, select a value from the **Assigned values** list.
 - For multi-valued custom security attributes, select **Add values** to open the **Attribute values** pane and add your values. When finished adding values, select **Done**.

The screenshot shows the 'Custom security attributes' page for the 'TestApp' application. The left sidebar includes links for Single sign-on, Provisioning, Application proxy, Self-service, and Custom security attributes (which is selected). The main area has tabs for Security (Conditional Access, Permissions, Token encryption) and Activity (Sign-in logs). At the top, there are Save, Discard, Add assignment, Remove assignment, and Got feedback? buttons. Below these are search and filter fields. A table lists attributes: Attribute set (Engineering), Attribute name (Project), Attribute descrip... (Active projects for ...), Data type (String), Multi-valued (Yes), Assigned val... (1 value), and a delete icon. A magnifying glass icon is in the bottom right corner.

Attribute set	Attribute name	Attribute descrip...	Data type	Multi-valued	Assigned val...
Engineering	Project	Active projects for ...	String	Yes	1 value

9. When finished, select **Save** to assign the custom security attributes to the application.

Update custom security attribute assignment values for an application

1. Sign in to the [Microsoft Entra admin center](#) as an **Attribute Assignment Administrator**.
2. Browse to **Identity > Applications > Enterprise applications**.

3. Find and select the application that has a custom security attribute assignment value you want to update.

4. In the Manage section, select **Custom security attributes**.

5. Find the custom security attribute assignment value you want to update.

Once you assigned a custom security attribute to an application, you can only change the value of the custom security attribute. You can't change other properties of the custom security attribute, such as attribute set or custom security attribute name.

6. Depending on the properties of the selected custom security attribute, you can update a single value, select a value from a predefined list, or update multiple values.

7. When finished, select **Save**.

Filter applications based on custom security attributes

You can filter the list of custom security attributes assigned to applications on the **All applications** page.

1. Sign in to the [Microsoft Entra admin center](#) as at least an **Attribute Assignment Reader**.

2. Browse to **Identity > Applications > Enterprise applications**.

3. Select **Add filters** to open the Pick a field pane.

If you don't see **Add filters**, select the banner to enable the Enterprise applications search preview.

4. For **Filters**, select **Custom security attribute**.

5. Select your attribute set and attribute name.

6. For **Operator**, you can select equals (==), not equals (!=), or starts with.

7. For **Value**, enter or select a value.

8. To apply the filter, select **Apply**.

Remove custom security attribute assignments from applications

1. Sign in to the Microsoft Entra admin center  as a [Attribute Assignment Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications**.
3. Find and select the application that has the custom security attribute assignments you want to remove.
4. In the **Manage** section, select **Custom security attributes (preview)**.
5. Add check marks next to all the custom security attribute assignments you want to remove.
6. Select **Remove assignment**.

Next steps

- [Add or deactivate custom security attributes in Microsoft Entra ID](#)
 - [Assign, update, list, or remove custom security attributes for a user](#)
 - [Troubleshoot custom security attributes in Microsoft Entra ID](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback !\[\]\(6609a55b4533263c6c837612b86292d6_img.jpg\)](#)

Disable user sign-in for an application

Article • 03/06/2025

There might be situations while configuring or managing an application where you don't want tokens to be issued for an application. Or, you might want to block an application that you don't want your employees to try to access. To block user access to an application, you can disable user sign-in for the application, which prevents all tokens from being issued for that application.

In this article, you learn how to prevent users from signing in to an application in Microsoft Entra ID through both the Microsoft Entra admin center and PowerShell. If you're looking for how to block specific users from accessing an application, use [user or group assignment](#).

Prerequisites

To disable user sign-in, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free](#).
- One of the following roles:
 - Cloud Application Administrator
 - Application Administrator
 - owner of the service principal

Disable user sign-in using the Microsoft Entra admin center

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. Search for the application you want to disable a user from signing in, and select the application.
4. Select **Properties**.
5. Select **No** for **Enabled for users to sign-in?**.
6. Select **Save**.

Related content

- Remove a user or group assignment from an enterprise app
-

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Configure sign-in auto-acceleration

Article • 11/29/2024

This article provides an introduction to configuring Microsoft Entra authentication behavior for federated users using Home Realm Discovery (HRD) policy. It covers using auto-acceleration sign-in to skip the username entry screen and automatically forward users to federated sign-in endpoints. To learn more about HRD policy, check out the [Home Realm Discovery](#) article.

Prerequisites

To configure HRD policy for an application in Microsoft Entra ID, you need:

- An Azure account with an active subscription. If you don't already have one, you can [create an account for free](#).
- The Application Administrator role

Auto-acceleration sign-in

Some organizations configure domains in their Microsoft Entra tenant to federate with another identity provider (IDP), such as Active Directory Federation Services (ADFS) for user authentication. When a user signs into an application, they're first presented with a Microsoft Entra sign-in page. After they type their User Principal Name (UPN), if they are in a federated domain they're then taken to the sign-in page of the IDP serving that domain. Under certain circumstances, administrators might want to direct users to the sign-in page when they're signing in to specific applications. As a result users can skip the initial Microsoft Entra ID page. This process is referred to as "sign-in auto-acceleration."

For federated users with cloud-enabled credentials, such as Short Message Service (SMS) sign-in or FIDO keys, you should prevent sign-in auto-acceleration. See [Disable auto-acceleration sign-in](#) to learn how to prevent domain hints with HRD.

ⓘ Important

Starting April 2023, organizations who use auto-acceleration or smartlinks might begin to see a new screen added to the sign-in UI. This screen, termed the Domain Confirmation Dialog, is part of Microsoft's general commitment to security hardening and requires the user to confirm the domain of the tenant in which they are signing in to. If you see the Domain Confirmation Dialog and do not recognize

the tenant domain listed, you should cancel the authentication flow and contact your IT Admin.

For more information, please visit [Domain Confirmation Dialog](#).

Set up an HRD policy using Microsoft Graph PowerShell

We use Microsoft Graph PowerShell cmdlets to walk through a few scenarios, including:

- Setting up HRD policy to do auto-acceleration for an application in a tenant with a single federated domain.
- Setting up HRD policy to do auto-acceleration for an application to one of several domains that are verified for your tenant.
- Setting up HRD policy to enable a legacy application to do direct username/password authentication to Microsoft Entra ID for a federated user.
- Listing the applications for which a policy is configured.

In the following examples, you create, update, link, and delete HRD policies on application service principals in Microsoft Entra ID.

1. Before you begin, run the Connect command to sign in to Microsoft Entra ID with at least the [Application Administrator](#) role:

```
PowerShell  
connect-MgGraph -scopes "Policy.Read.All"
```

2. Run the following command to see all the policies in your organization:

```
PowerShell  
Get-MgPolicyHomeRealmDiscoveryPolicy -Property Id, displayName
```

If nothing is returned, it means you have no policies created in your tenant.

Create an HRD policy using Microsoft Graph PowerShell

In this example, you create a policy such that when you assign it to an application, it either:

- Auto-accelerates users to a federated identity provider sign-in screen when they're signing in to an application when there's a single domain in your tenant.
- Auto-accelerates users to a federated identity provider sign-in screen if there's more than one federated domain in your tenant.
- Enables non-interactive username/password sign-in directly to Microsoft Entra ID for federated users for the applications the policy is assigned to.

The following policy auto-accelerates users to a federated identity provider sign-in screen when they're signing in to an application when there's a single domain in your tenant.

1. Run the Connect command to sign in to Microsoft Entra ID with at least the [Application Administrator](#) role:

PowerShell

```
connect-MgGraph -scopes "Policy.ReadWrite.ApplicationConfiguration"
```

2. Run the following command to create a new HRD policy:

PowerShell

```
# Define the parameters for the policy
$params = @{
    definition = @(
        '{"HomeRealmDiscoveryPolicy":{
            "AccelerateToFederatedDomain":true,
        }
    }'
)
displayName = "BasicAutoAccelerationPolicy"
isOrganizationDefault = $true
}
# Create a new Home Realm Discovery Policy
New-MgPolicyHomeRealmDiscoveryPolicy -BodyParameter $params
```

The following policy auto-accelerates users to a federated identity provider sign-in screen when there's more than one federated domain in your tenant. If you have more than one federated domain that authenticates users for applications, you need to specify the domain to auto-accelerate.

PowerShell

```
connect-MgGraph -scopes "Policy.ReadWrite.ApplicationConfiguration"
```

```
# Define the parameters for the New-MgPolicyHomeRealmDiscoveryPolicy cmdlet
```

```

$params = @{
    definition = @(
        '{"HomeRealmDiscoveryPolicy":{
            "AccelerateToFederatedDomain":true,
            "PreferredDomain":"federated.example.edu"
        }}"
    )
    displayName = "MultiDomainAutoAccelerationPolicy"
    isOrganizationDefault = $true
}

# Create the new policy
New-MgPolicyHomeRealmDiscoveryPolicy -BodyParameter $params

```

The following policy enables username/password authentication for federated users directly with Microsoft Entra ID for specific applications:

PowerShell

```

connect-MgGraph -scopes "Policy.ReadWrite.ApplicationConfiguration"

# Define the parameters for the New-MgPolicyHomeRealmDiscoveryPolicy cmdlet
$params = @{
    definition = @(
        '{"HomeRealmDiscoveryPolicy":{
            "AllowCloudPasswordValidation":true
        }}"
    )
    displayName = "EnableDirectAuthPolicy"
}

New-MgPolicyHomeRealmDiscoveryPolicy -BodyParameter $params

```

To see your new policy and get its **ObjectID**, run the following command:

PowerShell

```
Get-MgPolicyHomeRealmDiscoveryPolicy -Property Id, displayName
```

To apply the HRD policy after creating it, you can assign it to multiple service principals.

Locate the service principal to assign the policy using Microsoft Graph PowerShell

You need the **ObjectID** of the service principals to which you want to assign the policy. There are several ways to find the **ObjectID** of service principals.

You can use the [Microsoft Entra admin center](#). Using this option:

1. Browse to **Identity > Applications > Enterprise applications > All applications**.
2. Enter the name of the existing application in the search box, and then select the application from the search results. Copy the Object ID of the application.

Because you're using Microsoft Graph PowerShell, run the following cmdlet to list the service principals and their IDs.

PowerShell

```
connect-MgGraph -scopes "Application.Read.All"
Get-MgServicePrincipal
```

Assign the policy to your service principal using Microsoft Graph PowerShell

After you have the **ObjectID** of the service principal of the application for which you want to configure auto-acceleration, run the following command. This command associates the HRD policy that you created with the service principal that you located in the previous sections.

PowerShell

```
connect-MgGraph -scopes "Policy.ReadWrite.ApplicationConfiguration",
"Application.ReadWrite.All"

# Define the parameters for the New-
MgServicePrincipalHomeRealmDiscoveryPolicy cmdlet
$assignParams = @{
    "@odata.id" =
    "https://graph.microsoft.com/v1.0/policies/homeRealmDiscoveryPolicies/<polic
yId>"
}

New-MgServicePrincipalHomeRealmDiscoveryPolicyByRef -ServicePrincipalId
$servicePrincipalId -BodyParameter $assignParams
```

You can repeat this command for each service principal to which you want to add the policy.

In the case where an application already has a Home Realm Discovery policy assigned, you can't add a second one. In that case, change the definition of the HRD policy that is assigned to the application to add extra parameters.

Check which service principals your HRD policy is assigned to using Microsoft Graph PowerShell

Run the following command to list the service principals to which the policy is assigned:

PowerShell

```
Get-MgPolicyHomeRealmDiscoveryPolicyApplyTo -HomeRealmDiscoveryPolicyId "<ObjectId of the Policy>"  
# Replace with the actual ObjectId of the Policy
```

Ensure you test the sign-in experience for the application to check that the new policy is working.

Remove an HRD policy from an application using Microsoft Graph PowerShell

1. Get the ObjectId of the policy.

Use the previous example for getting the **ObjectId** of the policy, and that of the application service principal from which you want to remove it.

2. Remove the policy assignment from the application service principal.

PowerShell

```
Remove-MgServicePrincipalHomeRealmDiscoveryPolicyHomeRealmDiscoveryPolicyByRef  
-ServicePrincipalId $servicePrincipalId -HomeRealmDiscoveryPolicyId  
$homeRealmDiscoveryPolicyId
```

3. Check removal by listing the service principals to which the policy is assigned.

PowerShell

```
Get-MgPolicyHomeRealmDiscoveryPolicyApplyTo -HomeRealmDiscoveryPolicyId "<ObjectId of the Policy>"  
# Replace with the actual ObjectId of the Policy
```

Delete the HRD policy using Microsoft Graph PowerShell

To delete the HRD policy you created, run the following command:

PowerShell

```
Remove-MgPolicyHomeRealmDiscoveryPolicy -HomeRealmDiscoveryPolicyId "<ObjectId of the Policy>" # Replace with the actual ObjectId of the Policy
```

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Disable auto-acceleration sign-in

Article • 11/29/2024

In this article, you learn how to disable auto-acceleration sign-in for certain domains and applications using the Home Realm Discovery (HRD) policy. With this policy configured, administrators can ensure that users always use their managed credentials, improving security and providing a consistent sign-in experience.

Home Realm Discovery (HRD) policy offers administrators multiple ways to control how and where their users authenticate. The `domainHintPolicy` section of the HRD policy is used to help migrate federated users to cloud managed credentials like [FIDO](#), by ensuring that they always visit the Microsoft Entra sign-in page and aren't auto-accelerated to a federated IDP because of domain hints. To learn more about HRD policy, see [Home Realm Discovery](#).

This policy is needed in situations where admins can't control or update domain hints during sign-in. For example, `outlook.com/contoso.com` sends the user to a sign-in page with the `&domain_hint=contoso.com` parameter appended, to auto-accelerate the user directly to the federated IDP for the `contoso.com` domain. Users with managed credentials sent to a federated IDP can't sign in using their managed credentials, reducing security, and frustrating users with randomized sign-in experiences. Admins rolling out managed credentials should also set up this policy to ensure that users can always use their managed credentials.

Prerequisites

To disable auto-acceleration sign-in for an application in Microsoft Entra ID, you need:

- An Azure account with an active subscription. If you don't already have one, you can [create an account for free](#).
- One of the following roles: Cloud Application Administrator, Application Administrator, or owner of the service principal.

Configure HRD to prevent domain hints using Microsoft Graph PowerShell

Admins of federated domains should set up this section of the HRD policy in a four-phase plan. The goal of this plan is to eventually get all users in a tenant to use their managed credentials regardless of domain or application, save those apps that have

hard dependencies on `domain_hint` usage. This plan helps admins find those apps, exempt them from the new policy, and continue rolling out the change to the rest of the tenant.

Pick a domain to initially roll this change out to. This domain is your test domain, so pick one that might be more receptive to changes in UX (For example, seeing a different sign-in page). The following example is configured to ignore all domain hints from all applications that use this domain name. Set this policy in your tenant-default HRD policy:

Run the Connect command to sign in to Microsoft Entra ID with at least the [Application Administrator](#) role:

PowerShell

```
connect-MgGraph -scopes "Policy.ReadWrite.ApplicationConfiguration"
```

1. Run the following command to prevent domain hints for the test domain.

PowerShell

```
# Define the Home Realm Discovery Policy parameters
$params = @{
    definition = @(
        @{
            "HomeRealmDiscoveryPolicy": {
                "DomainHintPolicy": {
                    "IgnoreDomainHintForDomains": [
                        "federated.example.edu"
                    ],
                    "RespectDomainHintForDomains": [],
                    "IgnoreDomainHintForApps": [],
                    "RespectDomainHintForApps": []
                }
            }
        }
    )
    displayName = "Home Realm Discovery Domain Hint Exclusion Policy"
    isOrganizationDefault = $true
}

# Define the Home Realm Discovery Policy ID (ensure this is set to a valid ID)
$homeRealmDiscoveryPolicyId = "<Your-Policy-ID-Here>" # Replace with your actual policy ID

# Update the policy to ignore domain hints for the specified domains
Update-MgPolicyHomeRealmDiscoveryPolicy -HomeRealmDiscoveryPolicyId $homeRealmDiscoveryPolicyId -BodyParameter $params
```

Ensure to Replace with the `app-client-Guid` with the actual app GUIDs and the placeholder domain value with the actual domain.

2. Gather feedback from the test domain users. Collect details for applications that broke as a result of this change - they have a dependency on domain hint usage, and should be updated. For now, add them to the `RespectDomainHintForApps` section:

PowerShell

```
# Define the Home Realm Discovery Policy parameters
$params = @{
    definition = @(
        '{'
            "HomeRealmDiscoveryPolicy": {
                "DomainHintPolicy": {
                    "IgnoreDomainHintForDomains": [
                        "federated.example.edu",
                            "RespectDomainHintForDomains": [],
                            "IgnoreDomainHintForApps": [],
                            "RespectDomainHintForApps": ["app1-clientID-Guid",
                                "app2-clientID-Guid"]
                        }
                    }
                }
            }
        )
    displayName = "Home Realm Discovery Domain Hint Exclusion Policy"
    isOrganizationDefault = $true
}
# Define the Home Realm Discovery Policy ID (ensure this is set to a
# valid ID)
$homeRealmDiscoveryPolicyId = "<Your-Policy-ID-Here>" # Replace with
your actual policy ID

# Update the policy to ignore domain hints for the specified domains
Update-MgPolicyHomeRealmDiscoveryPolicy -HomeRealmDiscoveryPolicyId
$homeRealmDiscoveryPolicyId -BodyParameter $params
```

Ensure to Replace with the `app-client-Guid` with the actual app GUIDs and the placeholder domain value with the actual domain.

3. Continue expanding rollout of the policy to new domains, and collecting more feedback.

PowerShell

```
# Define the Home Realm Discovery Policy parameters
$params = @{
    definition = @(
        '{'
```

```

        "HomeRealmDiscoveryPolicy": {
            "DomainHintPolicy": {
                "IgnoreDomainHintForDomains": ["federated.example.edu",
"otherDomain.com", "anotherDomain.com"],
                "RespectDomainHintForDomains": [],
                "IgnoreDomainHintForApps": [],
                "RespectDomainHintForApps": ["app1-clientID-Guid",
"app2-clientID-Guid"]
            }
        }
    }
}

displayName = "Home Realm Discovery Domain Hint Exclusion Policy"
isOrganizationDefault = $true
}

# Define the Home Realm Discovery Policy ID (ensure this is set to a
# valid ID)
$homeRealmDiscoveryPolicyId = "<Your-Policy-ID-Here>" # Replace with
your actual policy ID

# Update the policy to ignore domain hints for the specified domains
Update-MgPolicyHomeRealmDiscoveryPolicy -HomeRealmDiscoveryPolicyId
$homeRealmDiscoveryPolicyId -BodyParameter $params

```

Ensure to Replace with the `app-client-Guid` with the actual app GUIDs and the placeholder domain value with the actual domain.

4. Complete your rollout - target all domains, exempting those that should continue to be accelerated:

PowerShell

```

$params = @{
definition = @(
{
    "HomeRealmDiscoveryPolicy": {
        "DomainHintPolicy": {
            "IgnoreDomainHintForDomains": ["*"],
            "RespectDomainHintForDomains":
["guestHandlingDomain.com"],
            "IgnoreDomainHintForApps": [],
            "RespectDomainHintForApps": ["app1-clientID-Guid",
"app2-clientID-Guid"]
        }
    }
}
)

displayName = "Home Realm Discovery Domain Hint Exclusion Policy"
isOrganizationDefault = $true
}

```

```

# Define the Home Realm Discovery Policy ID (ensure this is set to a
valid ID)
$homeRealmDiscoveryPolicyId = "<Your-Policy-ID-Here>" # Replace with
your actual policy ID

# Update the policy to ignore domain hints for the specified domains
Update-MgPolicyHomeRealmDiscoveryPolicy -HomeRealmDiscoveryPolicyId
$homeRealmDiscoveryPolicyId -BodyParameter $params

```

Ensure to Replace with the `app-client-Guid` with the actual app GUIDs and the placeholder domain value with the actual domain.

After step 4 is complete all users, except users in `guestHandlingDomain.com`, can sign-in at the Microsoft Entra sign-in page even when domain hints would otherwise cause an auto-acceleration to a federated IDP. The exception to this setting is if the app requesting sign-in is one of the exempted ones - for those apps, all domain hints are still accepted.

DomainHintPolicy details

The DomainHintPolicy section of the HRD policy is a JSON object that allows an admin to opt out certain domains and applications from domain hint usage. Functionally, this section tells the Microsoft Entra sign-in page to behave as if a `domain_hint` parameter on the sign-in request wasn't present.

The Respect and Ignore policy sections

[] Expand table

Section	Meaning	Values
<code>IgnoreDomainHintForDomains</code>	If this domain hint is sent in the request, ignore it.	Array of domain addresses (for example <code>contoso.com</code>). Also supports <code>all_domains</code>
<code>RespectDomainHintForDomains</code>	If this domain hint is sent in the request, respect it even if <code>IgnoreDomainHintForApps</code> indicates that the app in the request shouldn't auto-accelerate. This property is for slowing the rollout of deprecating domain hints within your network – you can indicate	Array of domain addresses (for example <code>contoso.com</code>). Also supports <code>all_domains</code>

Section	Meaning	Values
	that some domains should still be accelerated.	
<code>IgnoreDomainHintForApps</code>	If a request from this application comes with a domain hint, ignore it.	Array of application IDs (GUIDs). Also supports <code>all_apps</code>

Section	Meaning	Values
<code>RespectDomainHintForApps</code>	If a request from this application comes with a domain hint, respect it even if <code>IgnoreDomainHintForDomains</code> includes that domain. Used to ensure some apps keep working if you discover they break without domain hints.	Array of application IDs (GUIDs). Also supports <code>all_apps</code>

Policy evaluation

The DomainHintPolicy logic runs on each incoming request that contains a domain hint and accelerates based on two pieces of data in the request – the domain in the domain hint, and the client ID (the app). In short - 'Respect' for a domain or app takes precedence over an instruction to "Ignore" a domain hint for a given domain or application.

- In the absence of any domain hint policy, or if none of the four sections reference the app or domain hint mentioned, [the rest of the HRD policy is evaluated](#).
- If either one (or both) of `RespectDomainHintForApps` or `RespectDomainHintForDomains` section includes the app or domain hint in the request, then the user is auto-accelerated to the federated IDP as requested.
- If either one (or both) of `IgnoreDomainHintsForApps` or `IgnoreDomainHintsForDomains` references the app or the domain hint in the request, and they're not referenced by the "Respect" sections, then the request won't be auto-accelerated, and the user remains at the Microsoft Entra sign-in page to provide a username.

Once a user enters a username at the sign-in page, they can use their managed credentials. If they choose not to use a managed credential, or they have none registered, they're taken to their federated IDP for credential entry as usual.

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Create collections on the My Apps portal

Article • 04/25/2025

Your users can use the My Apps portal to view and start the cloud-based applications they have access to. By default, all the applications a user can access are listed together on a single page. To better organize this page for your users, if you have a Microsoft Entra ID P1 or P2 license you can set up collections. With a collection, you can group together applications that are related. For example, by job role, task, or project and display them on a separate tab. A collection essentially applies a filter to the applications a user can already access, so the user sees only those applications in the collection that have been assigned to them.

(!) Note

This article covers how an admin can enable and create collections. For information for the end user about how to use the My Apps portal and collections, see [Access and use collections](#).

Prerequisites

To create collections on the My Apps portal, you need:

- An Azure account with an active subscription. [Create an account for free](#).
- One of the following roles: Cloud Application Administrator, Application Administrator, or owner of the service principal.

Create a collection

To create a collection, you must have a Microsoft Entra ID P1 or P2 license.

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Entra ID > Enterprise apps**.
3. Under **Manage**, select **App launchers**.
4. Select **New collection**. In the **New collection** page, enter a **Name** for the collection (we recommend not using "collection" in the name). Then enter a **Description**.
5. Select the **Applications** tab. Select **+ Add application**, and then in the **Add applications** page, select all the applications you want to add to the collection, or use the **Search** box

to find applications.

The screenshot shows the 'Add applications' dialog box. On the left, there's a sidebar with tabs: Basics (selected), Applications, Owners, and Users and groups. Below the tabs is a button '+ Add application'. A note says: 'Add the applications you want to include in 1 collection. Note that users with read access to this collection will see them in this order.' Under the 'Name' section, it says 'No applications found.' At the bottom of the sidebar are buttons: 'Review + create', 'Previous', 'Next', 'Add' (highlighted in blue), and 'Discard'. The main area is titled 'Select applications' with a search bar: 'First 50 applications shown. To search, enter a display name.' It lists several applications with their icons: AirWatch, Apptio, Box, Egnyte, and Expense App Provisioning Agent. To the right of the list are up and down arrows for reordering. Below the list is a section titled 'Selected applications' which is currently empty.

6. When you're finished adding applications, select **Add**. The list of selected applications appears. You can use the arrows to change the order of applications in the list.
7. Select the **Owners** tab. Select **+ Add users and groups**, and then in the **Add users and groups** page, select the users or groups you want to assign ownership to. When you're finished selecting users and groups, choose **Select**.
8. Select the **Users and groups** tab. Select **+ Add users and groups**, and then in the **Add users and groups** page, select the users or groups you want to assign the collection to. Or use the **Search** box to find users or groups. When you're finished selecting users and groups, choose **Select**.
9. Select **Review + Create**. The properties for the new collection appear.

 **Note**

Admin collections are managed through the [Microsoft Entra admin center](#), not from [My Apps portal](#). For example, if you assign users or groups as an owner, then they can only manage the collection through the Microsoft Entra admin center.

Note

There is a known issue with Office apps in collections. If you already have at least one Office app in a collection and want to add more, follow these steps:

1. Select the collection you'd like to manage, then select the **Applications** tab.
2. Remove all Office apps from the collection but do not save the changes.
3. Select **+ Add application**.
4. In the **Add applications** page, select all the Office apps you want to add to the collection (including the ones that you removed in step 2).
5. When you're finished adding applications, select **Add**. The list of selected applications appears. You can use the arrows to change the order of applications in the list.
6. Select **Save** to apply the changes.

View audit logs

The Audit logs record My Apps collections operations, including collection creation end-user actions. The following events are generated from My Apps:

- Create admin collection
- Edit admin collection
- Delete admin collection
- Self-service application adding (end user)
- Self-service application deletion (end user)

You can access audit logs in the [Microsoft Entra admin center](#) by selecting **Entra ID > Enterprise apps > Audit logs** in the Activity section. For **Service**, select **My Apps**.

Get support for My Account pages

From the My Apps page, a user can select **My account > View account** to open their account settings. On the Microsoft Entra ID **My Account** page, users can manage their security info, devices, passwords, and more. They can also access their Office account settings.

In case you need to submit a support request for an issue with the Microsoft Entra account page or the Office account page, follow these steps so your request is routed properly:

- For issues with the **Microsoft Entra ID "My Account"** page, open a support request from within the Microsoft Entra admin center. Go to [Microsoft Entra admin center > Learn &](#)

[support](#) > New support request.

- For issues with the Office "My account" page, open a support request from within the Microsoft 365 admin center. Go to [Microsoft 365 admin center](#) > Support.

Next steps

[End-user experiences for applications in Microsoft Entra ID](#)

Enable self-service application assignment

Article • 04/29/2025

In this article, you learn how to enable self-service application access using the Microsoft Entra admin center.

Before your users can self-discover applications from the [My Apps portal](#), you need to enable **Self-service application access** for the applications. This functionality is available for applications that were added from the Microsoft Entra Gallery. It's also available for [Microsoft Entra application proxy](#), or applications added using [user or admin consent](#).

Using this feature, you can:

- Let users self-discover applications from the My Apps portal without bothering the IT group.
- Add those users to a preconfigured group so you can see who requests access, remove access, and manage the roles assigned to them.
- Optionally allow a business approver to approve application access requests so the IT group doesn't have to.
- Optionally configure up to 10 individuals who might approve access to this application.
- Optionally allow a business approver to set the passwords those users can use to sign in to the application, right from the business approver's My Apps portal
- Optionally automate the assignment of self-service assigned users to an application role directly.

Prerequisites

To enable self-service application access, you need:

- A Microsoft Entra user account. If you don't already have one, [create an account for free ↗](#).
- One of the following roles: Cloud Application Administrator, or Application Administrator.
- A Microsoft Entra ID P1 or P2 license is required for users to request to join a self-service app and for owners to approve or deny requests. Without a Microsoft Entra ID P1 or P2 license, users can't add self-service apps.

Enable self-service application access to allow users to find their own applications

Self-service application access is a great way to allow users to self-discover applications, and optionally allow the business group to approve access to those applications. For password single-sign on applications, you can also allow the business group to manage the credentials assigned to those users from their own My Apps portal.

To enable self-service application access to an application, undertake the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Entra ID > Enterprise apps > All applications**.
3. Enter the name of the existing application in the search box, and then select the application from the search results.
4. In the left navigation menu, select **Self-service**.
5. To enable Self-service application access for this application, set **Allow users to request access to this application?** to **Yes**.
6. Next to **To which group should assigned users be added?**, select **Select group**. Choose a group, and then select **Select**. When a user's request is approved, they're added to this group. When viewing this group's membership, you're able to see who has access to the application through self-service access.

 **Note**

This setting doesn't support groups synchronized from on-premises.

7. **Optional:** To require business approval before users are allowed access, set **Require approval before granting access to this application?** to **Yes**.
8. **Optional:** Next to **Who is allowed to approve access to this application?**, select **Select approvers** to specify the business approvers who are allowed to approve access to this application. Select up to ten individual business approvers, and then select **Select**.

 **Note**

Groups aren't supported. You can select up to ten individual business approvers. If you specify multiple approvers, any single approver can approve an access request.

9. Optional: Next to **To which role should users be assigned in this application?**, select **Select Role** to assign self-service approved users to a role. Choose the role to which these users should be assigned, and then select **Select**. This option is for applications that expose roles.

10. Select the **Save** button at the top of the pane to finish.

Once you complete self-service application configuration, users can navigate to their My Apps portal, and select **Request new apps** to find the apps that are enabled with self-service access. Business approvers also see a notification in their My Apps portal. You can enable an email notifying them when a user requests access to an application that requires their approval.

Related content

[Setting up Microsoft Entra ID for self-service group management](#)

Hide an enterprise application

Article • 04/25/2025

Learn how to hide enterprise applications in Microsoft Entra ID. When an application is hidden, users still have permissions to the application.

Prerequisites

To hide an application from the My Apps portal and Microsoft 365 launcher, you need:

- A Microsoft Entra account with an active subscription. [Create an account for free ↗](#).
- One of the following roles:
 - Cloud Application Administrator
 - Application Administrator.
 - Global Administrator is required to hide all Microsoft 365 applications.

Hide an application from the end user

Use the following steps to hide an application from My Apps portal and Microsoft 365 application launcher.

1. Sign in to the [Microsoft Entra admin center ↗](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Entra ID > Enterprise apps > All applications**.
3. Search for the application you want to hide, and select the application.
4. In the left navigation pane, select **Properties**.
5. Select **No** for the **Visible to users?** question.
6. Select **Save**.

Note

These instructions apply only to non-first-party Microsoft Enterprise Applications. To learn more about first-party Microsoft applications see [First-party Microsoft applications in sign-in reports](#). Administrators also need to keep in mind that hiding the application from the users doesn't prevent them from signing into these applications via methods other than the My Apps portal, such as shared links or service dependencies.

Hide Microsoft 365 applications from the My Apps portal

Use the following steps to hide all Microsoft 365 applications from the My Apps portal. The applications are still visible in the Office 365 portal.

ⓘ Important

Microsoft recommends that you use roles with the fewest permissions. This practice helps improve security for your organization. Global Administrator is a highly privileged role that should be limited to emergency scenarios or when you can't use an existing role.

1. Sign in to the [Microsoft Entra admin center](#) as a **Global Administrator**.
2. Browse to **Entra ID > Enterprise apps**.
3. Select **App launchers** under **Manage** menu items.
4. Select **Settings**.
5. Enable the option of **Users can only see Microsoft 365 apps in the Microsoft 365 portal**.
6. Select **Save**.

Related content

- [Remove a user or group assignment from an enterprise app](#)

Use AD FS application migration to move AD FS apps to Microsoft Entra ID

Article • 06/10/2024

In this article, you learn how to migrate your Active Directory Federation Services (AD FS) applications to Microsoft Entra ID using the AD FS application migration.

The AD FS application migration provides IT Admins guided experience to migrate AD FS relying party applications from AD FS to Microsoft Entra ID. The wizard gives you a unified experience to discover, evaluate, and configure new Microsoft Entra application. It provides one-click configuration for basic SAML URLs, claims mapping, and user assignments to integrate the application with Microsoft Entra ID.

The AD FS application migration tool is designed to provide end-to-end support to migrate your on-premises AD FS applications to Microsoft Entra ID.

With AD FS application migration you can:

- **Evaluate AD FS relying party application sign-in activities**, which helps you to identify the usage and impact of the given applications.
- **Analyze AD FS to Microsoft Entra migration feasibility** that helps you to identify migration blockers or actions required to migrate their applications to Microsoft Entra platform.
- **Configure new Microsoft Entra application using one-click application migration process**, which automatically configures a new Microsoft Entra application for the given AD FS application.

Prerequisites

To use the AD FS application migration:

- Your organization must be currently using AD FS to access applications.
- You have a Microsoft Entra ID P1 or P2 license.
- You should have one of the following roles assigned,
 - Cloud Application Administrator
 - Application Administrator
 - Global Reader (read-only access)
 - Report Reader (read-only access)
- Microsoft Entra Connect should be installed on the on-premises environments, alongside Microsoft Entra Connect Health AD FS health agents.

- Microsoft Entra Connect ↗
- Microsoft Entra Connect Health AD FS agents ↗

There are couple reasons why you won't see all the applications that you're expecting after you have installed Microsoft Entra Connect Health agents for AD FS:

- The AD FS application migration dashboard only shows AD FS applications that have user logins in the last 30 days.
- Microsoft related AD FS relying party applications aren't available on the dashboard.

View AD FS application migration dashboard in Microsoft Entra ID

The AD FS application migration dashboard is available in the Microsoft Entra admin center under **Usage & insights** reporting. There are a two entry points to the wizard:

From **Enterprise applications** section:

1. Sign in to the Microsoft Entra admin center ↗ as at least a **Cloud Application Administrator**.
2. Browse to **Identity > Applications > Enterprise applications**.
3. Under **Usage & Insights**, select **AD FS application migration** to access the AD FS applications migration dashboard.

From **Monitoring & health** section:

1. Sign in to the Microsoft Entra admin center ↗ as at least a **Cloud Application Administrator**.
2. Browse to **Identity > Monitoring & health > Enterprise applications**.
3. Under **Manage**, select **Usage & Insights**, and then select **AD FS application migration** to access the AD FS applications migration dashboard.

The AD FS application migration dashboard shows the list of all your AD FS relying party applications that have actively had sign-in traffic in the last 30 days period.

The dashboard has the date range filter. The filter allows you to select all the active AD FS relying party application as per selected time range. The filter supports last 1 day, 7 days, and 30 days period.

There are three tabs that give the complete list of applications, configurable applications, and previously configured applications. From this dashboard, you see an overview of overall progress of your migration work.

The three tabs on the dashboard are:

- **All apps** - shows the list of all applications that are discovered from your on-premises environment.
- **Ready to migrate** - shows list of all the applications that have **Ready** or **Needs review** migration status.
- **Ready to configure** - shows the list of all the Microsoft Entra applications that were previously migrated using AD FS application migration wizard.

Application migration status

The Microsoft Entra Connect and Microsoft Entra Connect Health agents for AD FS reads your AD FS relying party application configurations and sign-in audit logs. This data about each AD FS application is analyzed to determine whether the application can be migrated as-is, or if additional review is needed. Based on the result of this analysis, migration status for the given application is indicated as one of the following statuses:

- **Ready to migrate** means, the AD FS application configuration is fully supported in Microsoft Entra ID and can be migrated as-is.
- **Needs review** means, some of the application's settings can be migrated to Microsoft Entra ID, but you need to review the settings that can't be migrated as-is. However, those aren't blocker for the migration.
- **Additional steps required** means, Microsoft Entra ID doesn't support some of the application's settings, so the application can't be migrated in its current state.

Let's review each tab on the AD FS application migration dashboard in more detail.

All apps tab

The **All apps** tab shows all active AD FS relying party applications from selected date range. User can analyze the impact of each application by using the aggregated sign-in data. They can also navigate to the details pane by using the **Migration status** link.

To view details about each validation rule, see [AD FS application migration validation rules](#).

Summary**⚠ Resolve migration issues to migrate this application to Microsoft Entra ID**

We've detected on-premises settings for this application that make it incompatible for Microsoft Entra ID. The issues preventing migration are listed below.

Potential migration issues

- ⚠ At least one non-migratable rule was detected for undefined.

Configuration tests passed

- ✓ No additional WS-Federation endpoints were found.
- ✓ AllowedAuthenticationClassReferences is not set up.
- ✓ AlwaysRequireAuthentication is not set up.
- ✓ AutoUpdateEnabled is not set up.
- ✓ No Additional Claim Providers were configured.
- ✓ Relying Party is not set to encrypt claims.
- ✓ Relying Party is not set to encrypt name ID.
- ✓ MonitoringEnabled is not set up.
- ✓ NotBeforeSkew is not set up.
- ✓ RequestMFAFromClaimsProviders is not set up.
- ✓ SignedSamlRequestsRequired is not set up.
- ✓ TokenLifetime is set to a supported value.
- ✓ AdditionalAuthentication is valid.
- ✓ DelegationAuthorization is valid.
- ✓ AccessControlPolicy is valid.

Select a message to open additional migration rule details. For a full list of the properties tested, see the following configuration tests table.

Check the results of claim rule tests

If you have configured a claim rule for the application in AD FS, the experience provides a granular analysis of all the claim rules. You see which claim rules you can move to Microsoft Entra ID and which ones need further review.

1. Select an app from the list of apps in the **All apps** tab, then select the status in the **Migration status** column to view migration details. You see a summary of the configuration tests that passed, along with any potential migration issues.
2. On the **Migration rule details** page, expand the results to display details about potential migration issues and to get additional guidance. For a detailed list of all claim rules tested, see the [claim rule tests](#) section in this article.

The following example shows migration rule details for the `IssuanceTransform` rule. It lists the specific parts of the claim that need to be reviewed and addressed before you can migrate the application to Microsoft Entra ID.

Migration rule details

X

! At least one non-migratable rule was detected for undefined.

Claim type: supported claim

The application has custom issuance transform rules defined in AD FS. Microsoft Entra ID supports the customizing the claims issued in the token. To learn more, see [Customize claims issued in the SAML token for enterprise applications](#). Refer to the [claim rule tests](#) table below for information about the rawIssuanceTransformation rules configured in AD FS.

The following raw IssuanceTransform rules are configured in AD FS

▼ ✓ Rule Name: MapClaims

▼ ✓ Rule Name: LdapClaims

^ ! Rule Name: GetADGroups Issuance

```
@RuleName = "GetADGroups"  
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]  
=> add(store = "Active Directory", types = ("http://temp/variable"), query = ";tokenGroups;{0}", param = c.Value);
```

1. UNSUPPORTED_ISSUANCE_CLASS: The issuance statement uses ADD to add additional claims to the incoming claim set. In Microsoft Entra ID this may be configured as multiple claims transformation. For more info, see [Customize claims emitted in tokens for a specific app in Microsoft Entra ID](#).

^ ! Rule Name: Roles Condition

```
@RuleName = "Roles"  
c:[Type == "http://temp/variable", Value =~ "(?i)^CL-AWS-(\\d{12})"]  
=> issue(Type = "https://aws.amazon.com/SAML/Attributes/Role", Value = RegExReplace(c.Value, "CLD-AWS-(\\d{12})-",  
"arn:aws:iam::$1:saml-provider/ADFS,arn:aws:iam::$1:role/"));
```

1. UNSUPPORTED_CONDITION_PARAMETER: The condition statement uses regular expressions to evaluate if the claim matches a certain pattern. To achieve a similar functionality in Microsoft Entra ID, you can use pre-defined transformations such as IfEmpty(), StartWith(), Contains(), and others. For more info, see [Customize claims issued in the SAML token for enterprise applications](#).

Claim rule tests

The following table lists all claim rule tests that are performed on AD FS applications.

[+] Expand table

Property	Description
UNSUPPORTED_CONDITION_PARAMETER	The condition statement uses Regular Expressions to evaluate if the claim matches a certain pattern. To achieve a similar functionality in Microsoft Entra ID, you can use predefined transformation such as IfEmpty(), StartWith(), Contains(), among others. For more information, see Customize claims issued in the SAML token for enterprise applications .
UNSUPPORTED_CONDITION_CLASS	The condition statement has multiple conditions that need to be evaluated before running the issuance statement. Microsoft Entra ID can support this functionality with the claim's transformation functions where you can evaluate multiple claim values. For more

Property	Description
	information, see Customize claims issued in the SAML token for enterprise applications .
UNSUPPORTED_RULE_TYPE	The claim rule couldn't be recognized. For more information on how to configure claims in Microsoft Entra ID, see Customize claims issued in the SAML token for enterprise applications .
CONDITION_MATCHES_UNSUPPORTED_ISSUER	The condition statement uses an Issuer that isn't supported in Microsoft Entra ID. Currently, Microsoft Entra doesn't source claims from stores different than Active Directory or Microsoft Entra ID. If this is blocking you from migrating applications to Microsoft Entra ID, let us know .
UNSUPPORTED_CONDITION_FUNCTION	The condition statement uses an aggregate function to issue or add a single claim regardless of the number of matches. In Microsoft Entra ID, you can evaluate the attribute of a user to decide what value to use for the claim with functions like IfEmpty(), StartWith(), Contains(), among others. For more information, see Customize claims issued in the SAML token for enterprise applications .
RESTRICTED_CLAIM_ISSUED	The condition statement uses a claim that is restricted in Microsoft Entra ID. You might be able to issue a restricted claim, but you can't modify its source or apply any transformation. For more information, see Customize claims emitted in tokens for a specific app in Microsoft Entra ID .
EXTERNAL_ATTRIBUTE_STORE	The issuance statement uses an attribute store different than Active Directory. Currently, Microsoft Entra doesn't source claims from stores different than Active Directory or Microsoft Entra ID. If this result is blocking you from migrating applications to Microsoft Entra ID, let us know .
UNSUPPORTED_ISSUANCE_CLASS	The issuance statement uses ADD to add claims to the incoming claim set. In Microsoft Entra ID, this can be configured as multiple claim transformations. For more information, see Customize claims issued in the SAML token for enterprise applications .

Property	Description
UNSUPPORTED_ISSUANCE_TRANSFORMATION	<p>The issuance statement uses Regular Expressions to transform the value of the claim to be emitted. To achieve similar functionality in Microsoft Entra ID, you can use predefined transformation such as <code>Extract()</code>, <code>Trim()</code>, and <code>ToLower()</code>. For more information, see Customize claims issued in the SAML token for enterprise applications.</p>

Ready to migrate tab

The **Ready to migrate** tab shows all the applications that have migration status as **Ready** or **Needs review**.

You can use the sign-in data to identify the impact of each application and select the right applications for the migration. Select **Begin migration** link to initiate the assisted one-click application migration process.

Ready to configure tab

This tab shows list of all the Microsoft Entra applications that were previously migrated using AD FS application migration wizard.

The **Application name** is the name of new Microsoft Entra application. **Application identifier** is same as of AD FS relying party application identifier that can be used to correlate the application with your AD FS environment. The **Configure application in Microsoft Entra** link enables you to navigate to the newly configured Microsoft Entra application within the **Enterprise application** section.

Migrate an app from AD FS to Microsoft Entra ID using AD FS application migration wizard

1. To initiate the application migration, select the **Begin migration** link for the application you want to migrate from the **Ready to migrate** tab.
2. The link redirects you to assisted one-click application migration section of the AD FS application migration wizard. All the configurations on the wizard are imported from your on-premises AD FS environment.

Before we go through the details of the various tabs in the wizard, it's important to understand the supported and unsupported configurations.

Supported configurations

The assisted AD FS application migration supports the following configurations:

- Supports SAML application configuration only.
- The option to customize the new Microsoft Entra application name.
- Allows users to select any application template from the application template galley.
- Configuration of basic SAML application configurations that is, identifier and reply URL.
- Configuration of Microsoft Entra application to allow all users from the tenant.
- Auto assignment of groups to the Microsoft Entra application.
- Microsoft Entra compatible claims configuration extracted from the AD FS relying party claims configurations.

Unsupported configurations:

The AD FS application migration doesn't support the following configurations:

- OIDC (OpenID Connect), OAuth, and WS-Fed configurations aren't supported.
- Auto configuration of Conditional Access policies isn't supported, however, user can configure the same after configuration of new application into their tenant.
- The signing certificate isn't migrated from the AD FS relying party application. The following tabs exist in the AD FS application migration wizard:

Let's look at the details of each tab in the assisted one-click application migration section of the AD FS application migration wizard

Basics tab

- **Application name** that is prepopulated with AD FS relying party application name. You can use it as the name for your new Microsoft Entra application. You can also modify the name to any other value you prefer.
- **Application template**. Select any application template that is most suitable for your application. You can skip this option if you don't want to use any template.

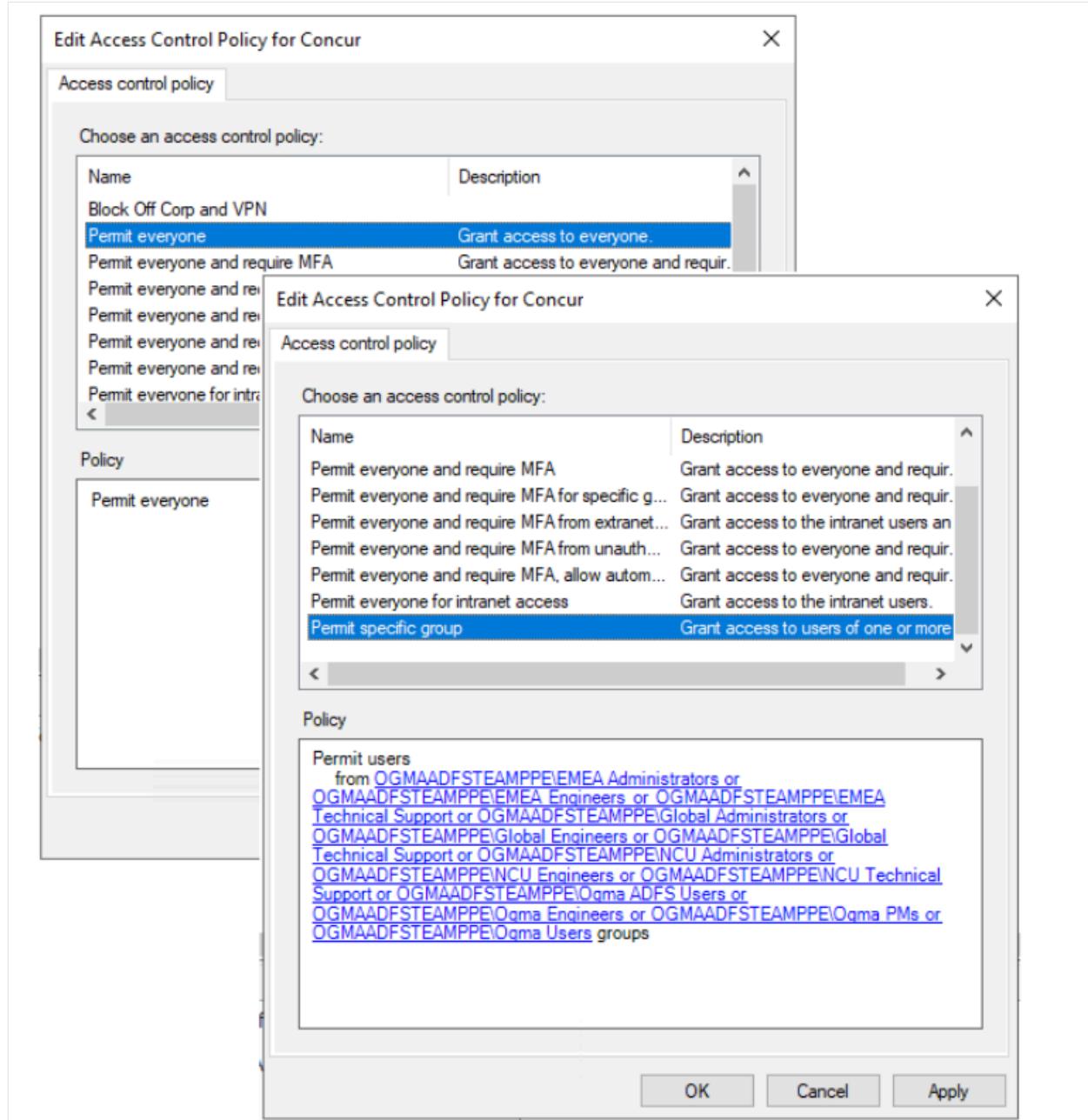
User & groups tab

The on-click configuration automatically assigns the users and groups to your Microsoft Entra application that are same as of your on-premises configuration.

All the groups are extracted from the access control policies of the AD FS relying party application. Groups should be synced into your Microsoft Entra tenant using Microsoft Entra Connect agents. In case groups are mapped with AD FS relying party application, but aren't synced with Microsoft Entra tenant. Those groups are skipped from configuration.

Assisted users and groups configuration supports the following configurations from the on-premises AD FS environment:

- Permit everyone from the tenant.
- Permit specific groups.



These are the users and groups you can view on the configuration wizard. This is a read-only view, you can't make any changes to this section.

SAML configurations tab

This tab shows the basic SAML properties that are used for the Single sign-on settings of the Microsoft Entra application. Currently, only required properties are mapped which are Identifier and Reply URL only.

These settings are directly implemented from your AD FS relying party application and can't be modified from this tab. However, after configuring application, you can modify these from the Microsoft Entra admin center's Single sign-on pane of your enterprise application.

Concur Properties

Organization	Endpoints	Proxy Endpoints	Notes	Advanced
Monitoring	Identifiers	Encryption	Signature	Accepted Claims

Specify the display name and identifiers for this relying party trust.

Display name:

Relying party identifier:

Example: <https://fs.contoso.com/adfs/services/trust>

Relying party identifiers:

https://	Concur Properties	X		
Monitoring	Identifiers	Encryption	Signature	Accepted Claims
Organization	Endpoints	Proxy Endpoints	Notes	Advanced

Specify the endpoints to use for SAML and WS-FederationPassive protocols.

URL	Index	Binding
SAML Assertion Consumer Endpoints https://www.com/SAMLRe...	0	POST

< >

Contoso | Application migration ...

Microsoft Entra ID

Configurations displayed on following tabs are extracted from your on-premises AD FS Relying party trust. These configurations will be used to configure the equivalent application into your Microsoft Entra tenant. [Learn more.](#)

Basics Users & groups **SAML configurations** Claims Next Steps Review + create

 These SAML settings are pulled directly from AD FS. You will be able to edit these after you migrate the application to the Microsoft Entra platform.

Basic SAML configuration settings	URL
Identifier	https://customer.contoso.com/mgmt
Reply URL	https://adfshelp.microsoft.com/ClaimsXray/TokenResponse

[Review + create](#)

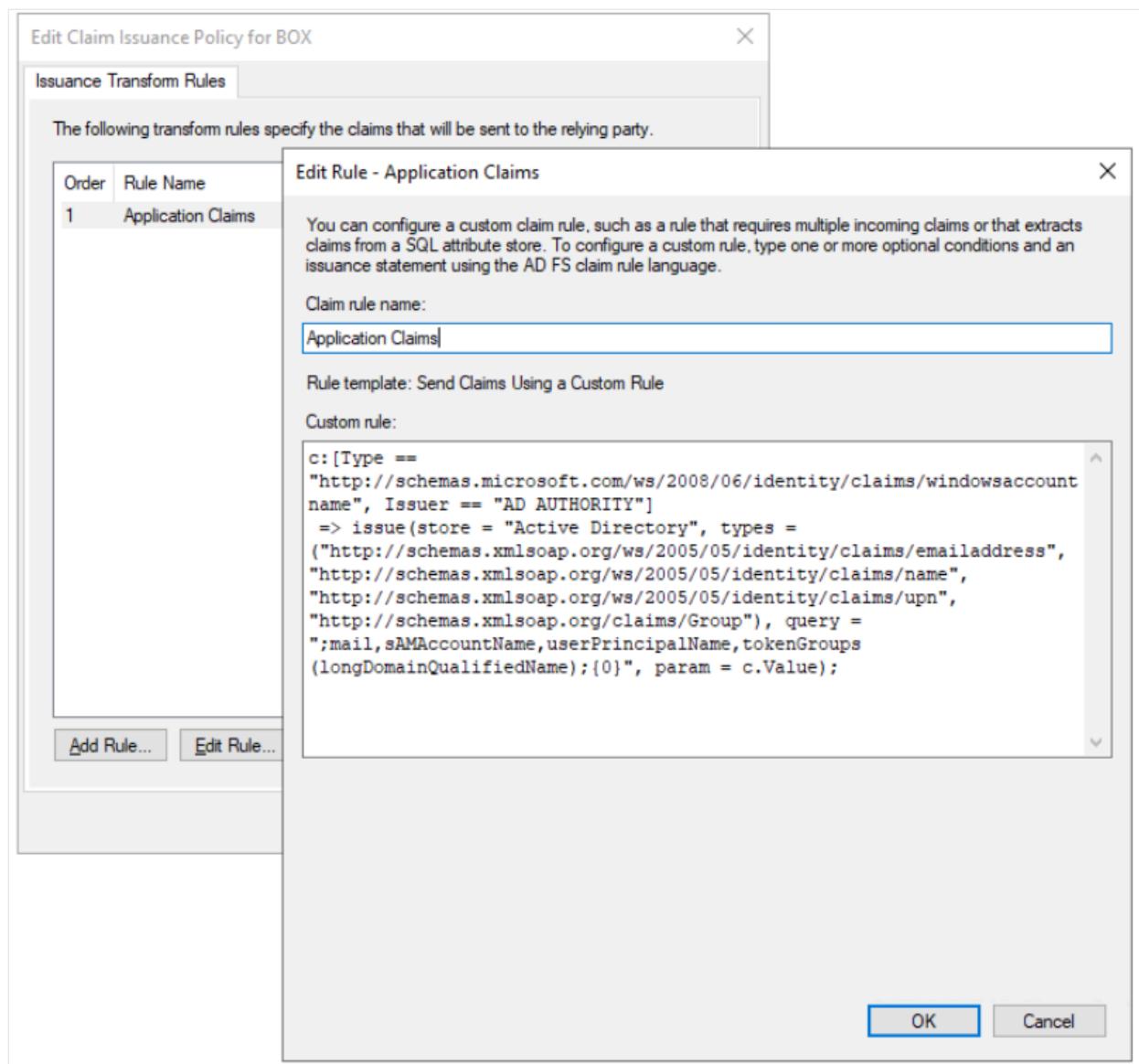
[Previous](#)

[Next: Claims >](#)

Claims tab

All AD FS claims don't get translated as is to the Microsoft Entra claims. The migration wizard supports specific claims only. If you find any missing claims, you can configure them on the migrated enterprise application in Microsoft Entra admin center.

In case, AD FS relying party application has `nameidentifier` configured which is supported in Microsoft Entra ID, then it's configured as `nameidentifier`. Otherwise, `user.userprincipalname` is used as default `nameidentifier` claim.



This is read-only view, you can't make any changes here.

... > Usage & insights | AD FS application migration (Preview) > AD FS application migration (Preview) >

Contoso | Application migration

Microsoft Entra ID

Configurations displayed on following tabs are extracted from your on-premises AD FS Relying party trust. These configurations will be used to configure the equivalent application into your Microsoft Entra tenant. [Learn more](#).

Basics Users & groups SAML configurations **Claims** Next Steps Review + create

These claims are pulled directly from AD FS. You will be able to edit these after you migrate this application to Microsoft Entra ID.

Claim name	Value
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier	user.userprincipalname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/onpremisessamaccountname	user.onpremisessamaccountname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	user.surname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/mail	user.mail
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	user.givenname

Review + create < Previous Next: Next Steps >

Next steps tab

This tab provides information about next steps or reviews that are expected from the user's side. The following example shows the list of configurations for this AD FS relying party application, which aren't supported in Microsoft Entra ID.

From this tab, you can access the relevant documentation to investigate and understand the issues.

The screenshot shows the 'Next Steps' tab selected in the navigation bar. A callout box highlights a note about rules configured in AD FS that don't translate to Microsoft Entra ID. Below, a table lists various configurations with corresponding error icons and descriptions.

Configuration test	Migration details
Automatic application updates configured	⚠️ This setting in AD FS lets you specify whether AD FS is configured to automatically update the application based on changes within the federation metadata. Microsoft Entra ID doesn't support this today but should not block the migration of the application to Microsoft Entra ID.
Monitoring enabled	⚠️ This setting in AD FS lets you specify whether AD FS is configured to automatically update the application based on changes within the federation metadata. Microsoft Entra ID doesn't support this today, but this should not block the migration of the application to Microsoft Entra ID.
Application allows a time skew	⚠️ AD FS allows a time skew based on the NotBefore and NotOnOrAfter times in the SAML token. Microsoft Entra ID handles this by default.
Multifactor authentication defined	⚠️ This setting in AD FS determines the behavior for multifactor authentication when the user comes from a different claims provider. In Microsoft Entra ID, you can enable external collaboration using Microsoft Entra B2B. Then, you can apply Conditional Access policies to protect guest access. Learn more
Custom token lifetime configured	⚠️ The application is configured for a custom token lifetime. The AD FS default is one hour. Microsoft Entra ID supports this functionality using Conditional Access. Learn more
Multifactor authentication defined	⚠️ The relying party has rules to prompt for multifactor authentication. To move to Microsoft Entra, translate those rules into Conditional Access policies. If you're using an on-premises multifactor authentication, we recommend that you move to Microsoft Entra multifactor authentication. Learn more

At the bottom, there are buttons for 'Review + create', '< Previous', and 'Next: Review + create >'.

Review + create tab

This tab shows the summary of all the configurations that you have seen from the previous tabs. You can review it once again. If you're happy with all the configurations and you want to go ahead with application migration, select the **Create** button to start the migration process. This migrates the new application into your Microsoft Entra tenant.

The application migration is currently a nine step process that you can monitor using the notifications. The workflow completes the following actions:

- Creates an application registration
- Creates a service principal
- Configures SAML settings
- Assigns users and groups to the application
- Configures claims

Once the migration process is complete, you see a notification message that reads **Application migration successful**.

The screenshot shows a 'Notifications' page with a single item. At the top, there's a header 'Notifications' and a close button 'X'. Below the header, there's a link 'More events in the activity log →' and a 'Dismiss all' button with a dropdown arrow. The main notification is titled 'Application configuration successful' with a checkmark icon. It contains the message: 'New Microsoft Entra application configured as 'Contoso - Entra'. You can review or update application configurations by clicking this notification or in the 'Enterprise Application' section.' At the bottom right of the notification box, it says 'a few seconds ago'.

On application migration completion, you get redirected to the **Ready to configure** tab where all previously migrated applications are shown, including the latest ones that you've configured.

Review and configure the enterprise application

1. From the **Ready to configure** tab, you can use the **Configure application in Microsoft Entra** link to navigate to the newly configured application under the "Enterprise applications" section. By default, it goes into the **SAML-based Sign-on** page of your application.

The screenshot shows the Microsoft Entra admin center interface for managing a SAML-based sign-on application. The left sidebar lists various management options like Overview, Deployment Plan, Diagnose and solve problems, Properties, Owners, Roles and administrators, Users and groups, Single sign-on (which is selected), Provisioning, Self-service, and Custom security attributes. The main content area is titled "Set up Single Sign-On with SAML" and provides an overview of the SSO implementation. It includes sections for "Basic SAML Configuration" (Identifier, Reply URL, Sign on URL, Relay State, Logout URL), "Attributes & Claims" (listing user attributes like onpremisesamaccountname, surname, mail, givenname, and Unique User Identifier), and "SAML Certificates" (listing a token signing certificate with status Active and thumbprint BF55E9608B9E69A1A98B7E11F146068E0877901C). Each section has an "Edit" button.

2. From the **SAML-based Sign-on** pane, all AD FS relying party application settings are already applied to the newly migrated Microsoft Entra application. The **Identifier** and **Reply URL** properties from the **Basic SAML Configuration** and list of claims from the **Attributes & Claims** tabs of the AD FS application migration wizard are the same as those on the enterprise application.
3. From the **Properties** pane of the application, the application template logo implies that the application is linked to the selected application template. On the **Owners** page, the current administrator user gets added as a one of the owners of the application.
4. From **Users and groups** pane, all required groups are already assigned to the application.

After reviewing the migrated enterprise application, you can update the application as per your business needs. You can add or update claims, assign more users and groups or configure Conditional Access policies to enable support for multifactor authentication or other conditional authorization features.

Rollback

The one-click configuration of AD FS application migration wizard migrates the new application into Microsoft Entra tenant. However, the migrated application remains

inactive until you redirect your sign-in traffic to it. Until then, if you want to roll back, you can delete the newly migrated Microsoft Entra application from your tenant.

The wizard doesn't provide any automated clean-up. In case you don't want to proceed with setting up the migrated application, you have to manually delete the application from your tenant. For instructions on how to delete an application registration and its corresponding enterprise application, see the following URLs:

- [Delete an application registration](#)
- [Delete an enterprise application](#)

Troubleshooting tips

Can't see all my AD FS applications in the report

If you have installed Microsoft Entra Connect Health agents for AD FS but you still see the prompt to install it or you don't see all your AD FS applications in the report, it might be that you don't have active AD FS applications, or your AD FS applications are Microsoft application.

Note

The AD FS application migration lists all the AD FS applications in your organization with active users sign-in in the last 30 days only. The report doesn't display Microsoft related relying parties in AD FS such as Office 365. For example, relying parties with name `urn:federation:MicrosoftOnline`, `microsoftonline`, `microsoft:winhello:cert:prov:server` don't show up in the list.

Why am I seeing the validation error "application with same identifier already exists"?

Each application within your tenant should have a unique application identifier. If you see this error message, it means you already have another application with the same identifier in your Microsoft Entra tenant. In this case, you either need to update the existing application identifier or update your AD FS relying party application identifier and wait for 24 hours to get updates reflected.

Next steps

- Managing applications with Microsoft Entra
 - Manage access to apps
-

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Migrate applications from Okta to Microsoft Entra ID

Article • 04/25/2025

In this tutorial, you learn how to migrate your applications from Okta to Microsoft Entra ID.

Prerequisites

To manage the application in Microsoft Entra ID, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free](#).
- One of the following roles: Cloud Application Administrator, Application Administrator, or owner of the service principal.

Create an inventory of current Okta applications

Before migration, document the current environment and application settings. You can use the Okta API to collect this information. Use an API explorer tool such as [Postman](#).

To create an application inventory:

1. With the Postman app, from the Okta admin console, generate an API token.
2. On the API dashboard, under **Security**, select **Tokens > Create Token**.

The screenshot shows the Okta Admin Console interface. The left sidebar is a navigation menu with sections like Dashboard, Directory, Applications, Security, API, Workflow, Reports, and Settings. The 'API' section is currently selected. The main content area is titled 'API' and shows the 'Tokens' tab selected. There is a 'Create Token' button highlighted with a blue box. Below it is a table with columns for Token Types, Token Name, Created, Expires, Last Used, and Revoke. The table shows three tokens: All (3), Okta AD Agent (2), and Okta API (1). At the bottom of the page, there are links for Help, Copyright notice (© 2021 Okta, Inc. Privacy Version 2021.03.0 OPI Preview Cell (US) Status site), Download Okta Plugin, and Feedback.

3. Enter a token name and then select **Create Token**.

Create Token

What do you want your token to be named?

API Authorization

The token name is used for tracking API calls.

Create Token

Cancel

4. Record the token value and save it. After you select **OK, got it**, it isn't accessible.

Create Token

Token created successfully!

Please make a note of this token as it will be the only time that you will be able to view it. After this, it will be stored as a hash for your protection.

Token Value

00

-hRrOL



OK, got it

5. In the Postman app, in the workspace, select **Import**.

6. On the **Import** page, select **Link**. To import the API, insert the following link:

<https://developer.okta.com/docs/api/postman/example.oktapreview.com.environment>

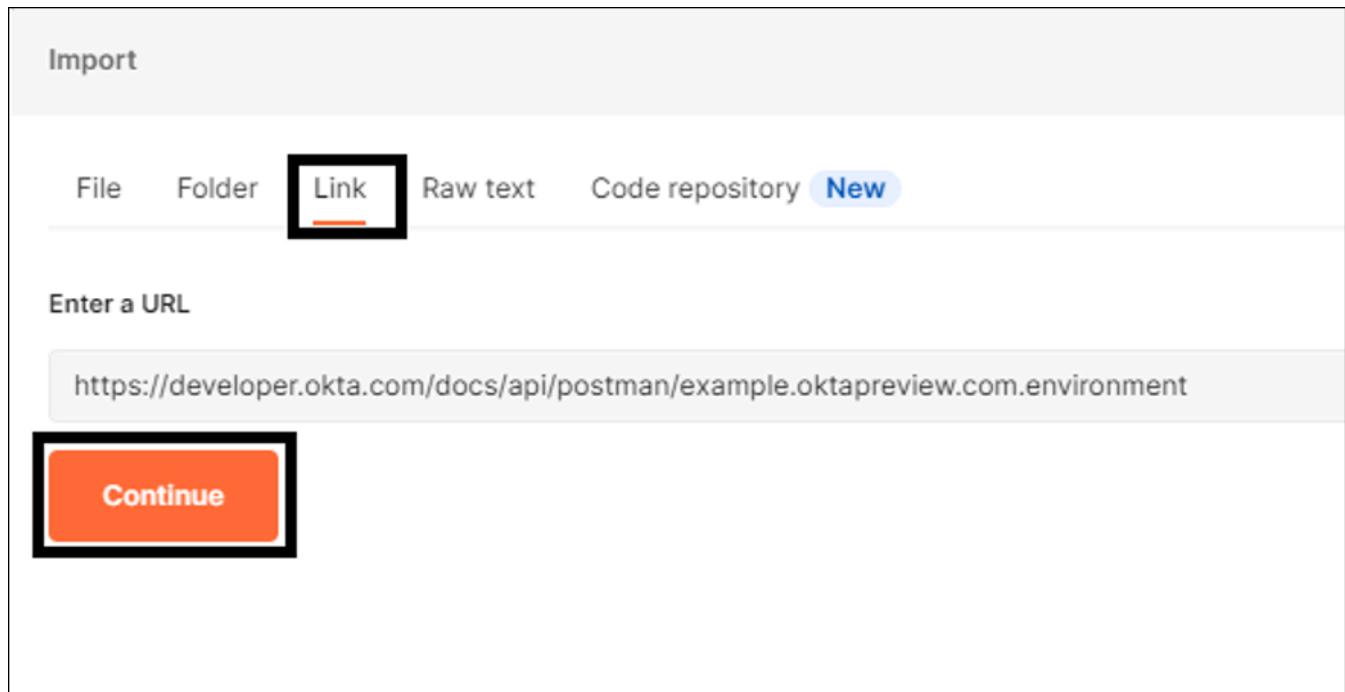
Import

File Folder **Link** Raw text Code repository New

Enter a URL

https://developer.okta.com/docs/api/postman/example.oktapreview.com.environment

Continue



➊ Note

Don't modify the link with your tenant values.

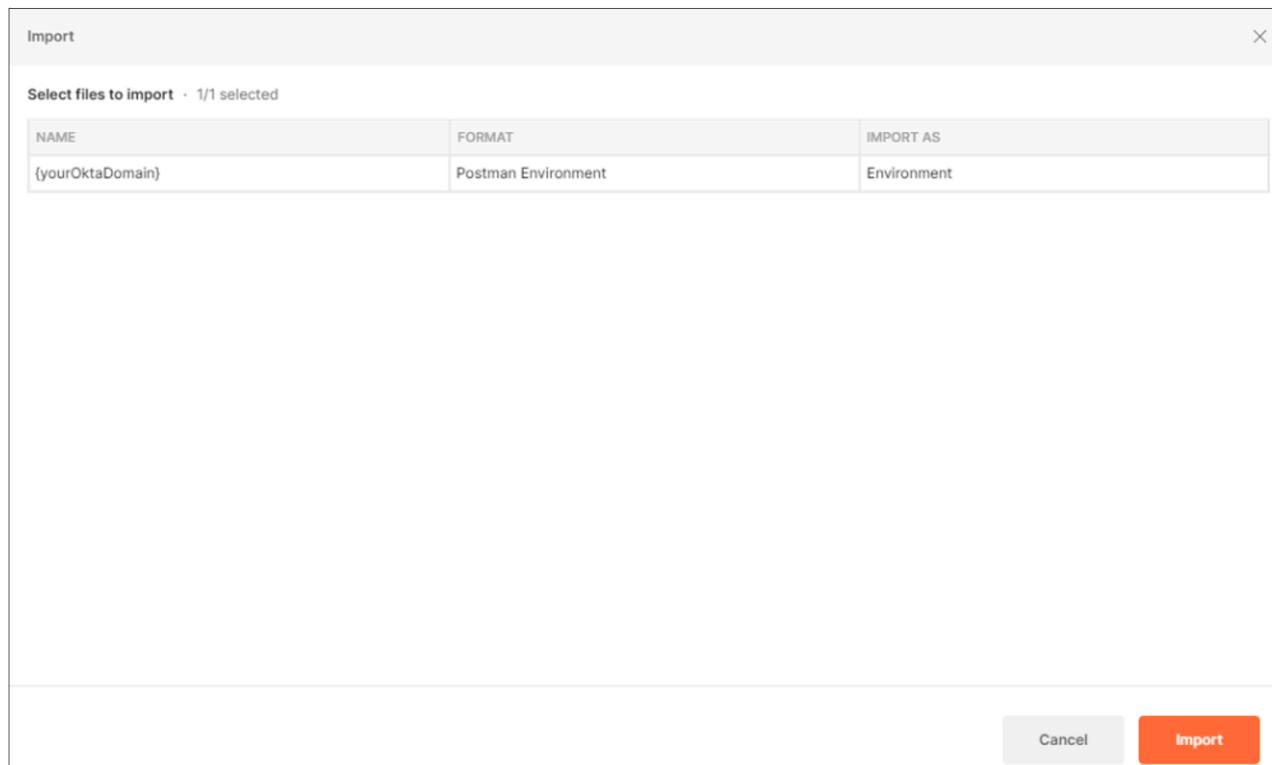
7. Select **Import**.

Import

Select files to import · 1/1 selected

NAME	FORMAT	IMPORT AS
{yourOktaDomain}	Postman Environment	Environment

Cancel Import



8. After the API is imported, change the **Environment** selection to **{yourOktaDomain}**.

9. To edit your Okta environment, select the eye icon. Then select **Edit**.

The screenshot shows the Postman Global Variables interface. It displays a table of variables with their initial and current values. The variables are:

VARIABLE	INITIAL VALUE	CURRENT VALUE
url	https://(yourOktaDomain)	https://(yourOktaDomain)
apikey		
email-suffix	example.com	example.com
username		
password		
userfield	Globals	

Below the table, there is a note about global variables: "Global variables are a set of variables that are always available in a workspace." A link to "Learn more about globals" is provided.

- In the **Initial Value** and **Current Value** fields, update the values for the URL and API key. Change the name to reflect your environment.

- Save the values.

The screenshot shows the Postman Global Variables interface with updated values. The variables are:

VARIABLE	INITIAL VALUE	CURRENT VALUE
url	https://[REDACTED].oktapreview.com/	https://[REDACTED].oktapreview.com/
apikey	00-[REDACTED]-hRrOL	00-[REDACTED]-hRrOL
email-suffix	example.com	example.com
username		

- Load the API into Postman

- Select Apps > Get List Apps > Send.

! Note

You can print the applications in your Okta tenant. The list is in JSON format.

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Shows the Test Workspace with various collections, environments, and mock servers. A specific endpoint, `GET List Apps`, is highlighted.
- Request URL:** `([url])/api/v1/apps`
- Method:** GET
- Params:** Key: Key, Value: Value
- Send Button:** The "Send" button is highlighted in blue.
- Response:** Status: 200 OK, Time: 968 ms, Size: 60.11 KB. The response body is displayed in JSON format:

```

1  [
2   {
3     "id": "0oa7wE7ejnqGsmuNp8h7",
4     "name": "seassure",
5     "label": "Okta Admin Console",
6     "status": "ACTIVE",
7     "lastUpdated": "2020-12-04T21:48:04.000Z",
8     "created": "2016-08-16T18:13:03.000Z",
9     "accessibility": {
10       "selfService": false,
11       "errorRedirecturi": null,
12       "loginRedirecturi": null
13     },
14     "visibility": {
15       "autoSubmitToolbar": false,
16       "hide": {
17         "ios": true
18       }
19     }
20   }
21 ]

```

We recommend you copy and convert this JSON list to a CSV format:

- Use a public converter such as [Konklone](#) ↗
- Or for PowerShell, use [ConvertFrom-Json](#) and [ConvertTo-Csv](#)

! Note

To have a record of the applications in your Okta tenant, download the CSV.

Migrate a SAML application to Microsoft Entra ID

To migrate a SAML 2.0 application to Microsoft Entra ID, configure the application in your Microsoft Entra tenant for application access. In this example, we convert a Salesforce instance.

1. Sign in to the [Microsoft Entra admin center](#) ↗ as at least a [Cloud Application Administrator](#).
2. Browse to **Entra ID > Enterprise apps > All applications**, then select **New application**.
3. In **Microsoft Entra Gallery**, search for **Salesforce**, select the application, and then select **Create**.

The screenshot shows the Microsoft Entra Gallery interface. On the left, there's a search bar with 'salesforce' typed in, and a sidebar with filters for 'Federated SSO' and 'Provisioning'. Below the search bar, it says 'Showing 6 of 6 results'. There are six application cards displayed: 1. 'Salesforce' by Salesforce, Inc. (selected, showing its details on the right). 2. 'DICE Salesforce'. 3. 'LegalForce' by 株式会社LegalOn Technologies. 4. 'Visual Workforce' by Visual Workforce, Inc. On the right side, a detailed view for the 'Salesforce' application is shown. It includes fields for 'Name' (set to 'Salesforce'), 'Publisher' (set to 'Salesforce, Inc.'), and 'Provisioning' (set to 'Automatic provisioning supported'). It also shows 'Single Sign-On Mode' (set to 'Password-based Sign-on'), 'URL' (set to 'https://www.salesforce.com/'), and provisioning options like 'SAML-based Sign-on' and 'Linked Sign-on'. A link to a 'Salesforce integration tutorial' is also present. At the bottom right of this panel is a blue 'Create' button.

4. After the application is created, on the **Single sign-on (SSO)** tab, select **SAML**.

The screenshot shows the Microsoft Entra admin center interface. On the left, there's a sidebar with navigation links for Overview, Deployment Plan, Diagnose and solve problems, Manage (Properties, Owners, Roles and administrators, Users and groups, Single sign-on, Provisioning, Self-service, Custom security attributes), Security (Conditional Access, Permissions, Token encryption), and Service Setup (Home, Object Manager). The 'Single sign-on' link under 'Manage' is highlighted.

The main content area has a heading 'Select a single sign-on method' with a 'Help me decide' link. It lists four options:

- Disabled**: Single sign-on is not enabled. The user won't be able to launch the app from My Apps. (Icon: crossed-out gear)
- SAML**: Rich and secure authentication to applications using the SAML (Security Assertion Markup Language) protocol. (Icon: gear with a chain)
- Password-based**: Password storage and replay using a web browser extension or mobile app. (Icon: lock)
- Linked**: Link to an application in My Apps and/or Office 365 application launcher. (Icon: linked circles)

5. Download the **Certificate (Raw)** and **Federation Metadata XML** to import it into Salesforce.

6. On the Salesforce administration console, select **Identity > Single Sign-On Settings > New from Metadata File**.

The screenshot shows the Salesforce Service Setup interface. The left sidebar includes links for Omni-Channel, PLATFORM TOOLS, AUTOMATION, USER INTERFACE, MOBILE, SETTINGS, and Identity (which is selected and highlighted). Under Identity, there are sub-links for Auth. Providers, Identity Provider, Identity Verification History, Login Flows, Login History, and Single Sign-On Settings (which is also highlighted).

The main content area is titled 'Single Sign-On Settings'. It contains sections for 'Single Sign-On Settings' (Configure single sign-on to authenticate users in salesforce.com from external environments), 'Delegated Authentication' (Disable login with Salesforce credentials, checked), 'Federated Single Sign-On Using SAML' (SAML Enabled, checked), and 'SAML Single Sign-On Settings' (New, New from Metadata File, New from Metadata URL). A table for 'SAML Single Sign-On Settings' is shown with columns for Action, Name, SAML Version, Issuer, and Entity ID.

7. Upload the XML file you downloaded from the Microsoft Entra admin center. Then select **Create**.

8. Upload the certificate you downloaded from Azure. Select **Save**.

9. Record the values in the following fields. The values are in Azure.

- Entity ID
- Login URL
- Logout URL

10. Select Download Metadata.

11. To upload the file to the Microsoft Entra admin center, in the Microsoft Entra ID **Enterprise applications** page, in the SAML SSO settings, select **Upload metadata file**.

12. Ensure the imported values match the recorded values. Select **Save**.

The screenshot shows the 'Basic SAML Configuration' page for a 'Salesforce | SAML-based Sign-on' application. The left sidebar lists various management options like Overview, Deployment Plan, and Single sign-on. The main configuration area is divided into three sections:

- Identifier (Entity ID) ***: Patterns: https://*.my.salesforce.com
- Reply URL (Assertion Consumer Service URL) ***: Patterns: https://MYDOMAIN.my.salesforce.com/<ENTITYID>
- Sign on URL ***: Patterns: https://mydomain-dev-ed.my.salesforce.com

Below these, there are optional fields for **Relay State (Optional)** and **Logout Url (Optional)**, both with their respective patterns filled in.

13. In the Salesforce administration console, select **Company Settings > My Domain**. Go to **Authentication Configuration** and then select **Edit**.

The screenshot shows the Salesforce Service Setup interface. The left sidebar has a tree view with 'Company Settings' expanded, and 'My Domain' is selected and highlighted with a red box. The main content area is titled 'My Domain' and contains sections for 'Routing' and 'Policies'. Under 'Policies', there are 'Login Policy' and 'Redirect Policy' settings. Below that is the 'Authentication Configuration' section, which includes fields for 'Login Page Type' (set to 'Standard'), 'Authentication Service' (with 'Azure' checked), 'Logo File', 'Background Color' (#F4F6F9), and 'Right Frame URL'. There are also checkboxes for 'Use the native browser for user authentication on iOS' and 'Use the native browser for user authentication on Android'. The 'Edit' button in the 'Authentication Configuration' section is also highlighted with a red box.

14. For a sign-in option, select the new SAML provider you configured. Select **Save**.

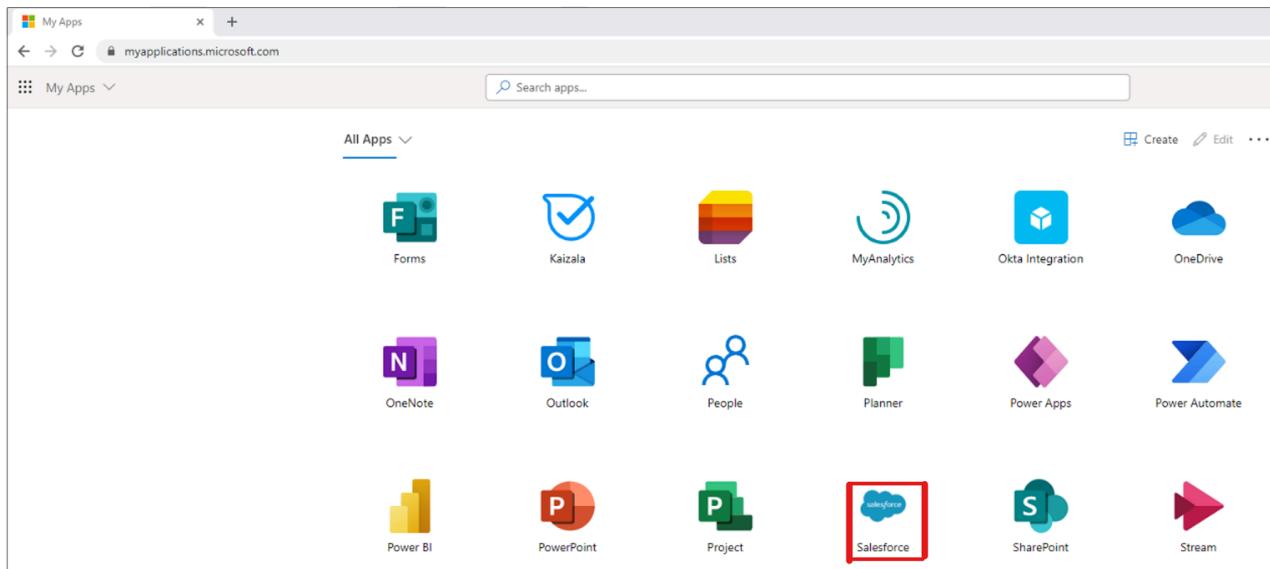
This screenshot shows the same 'My Domain' setup page as the previous one, but the 'Edit' button in the 'Authentication Configuration' section is highlighted with a red box. The 'Authentication Service' dropdown is set to 'Azure', and the 'MVTradersMigration' provider is checked in the list of available providers.

15. In Microsoft Entra ID, on the **Enterprise applications** page, select **Users and groups**. Then add test users.

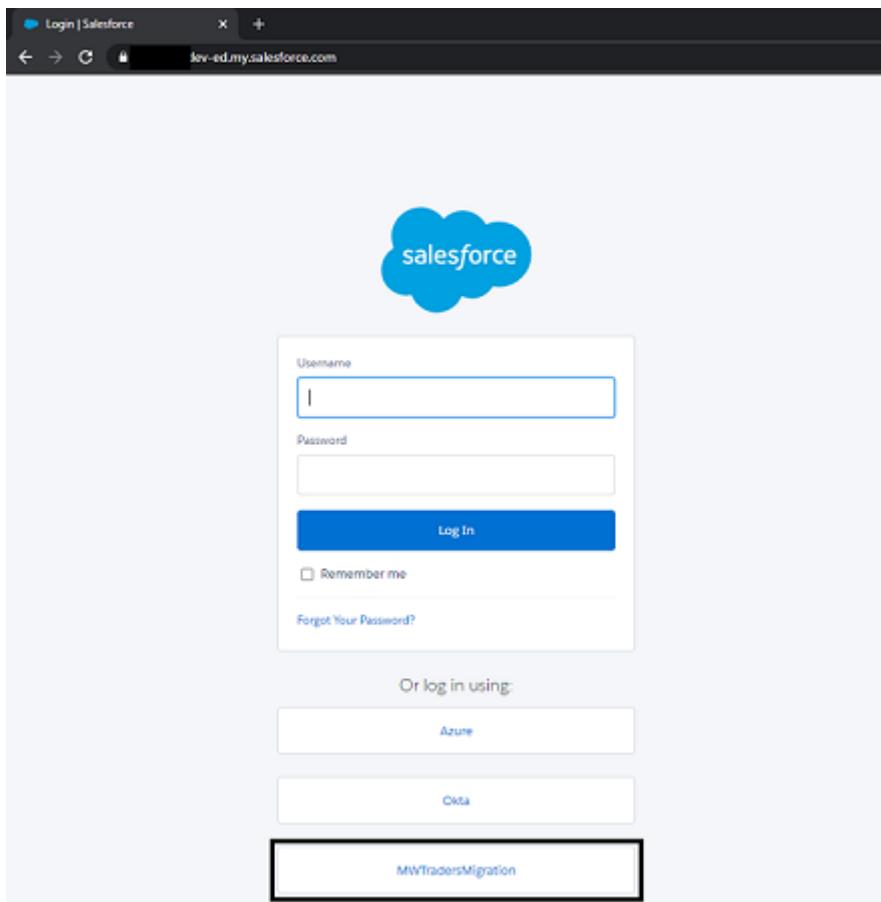
This screenshot shows the Microsoft Entra ID interface under the 'Enterprise applications' section. The 'Salesforce' application is selected. The 'Users and groups' tab is active in the sidebar. A table lists two users: 'User One' and 'User Two', both of whom are assigned to the 'Chatter External User' role. The table columns include 'Display Name', 'Object Type', and 'Role assigned'.

Display Name	Object Type	Role assigned
User One	User	Chatter External User
User Two	User	Chatter External User

16. To test the configuration, sign in as a test user. Go to the Microsoft apps gallery  and then select **Salesforce**.



17. To sign in, select the configured identity provider (IdP).



Note

If configuration is correct, the test user lands on the Salesforce home page. For troubleshooting help, see the [debugging guide](#).

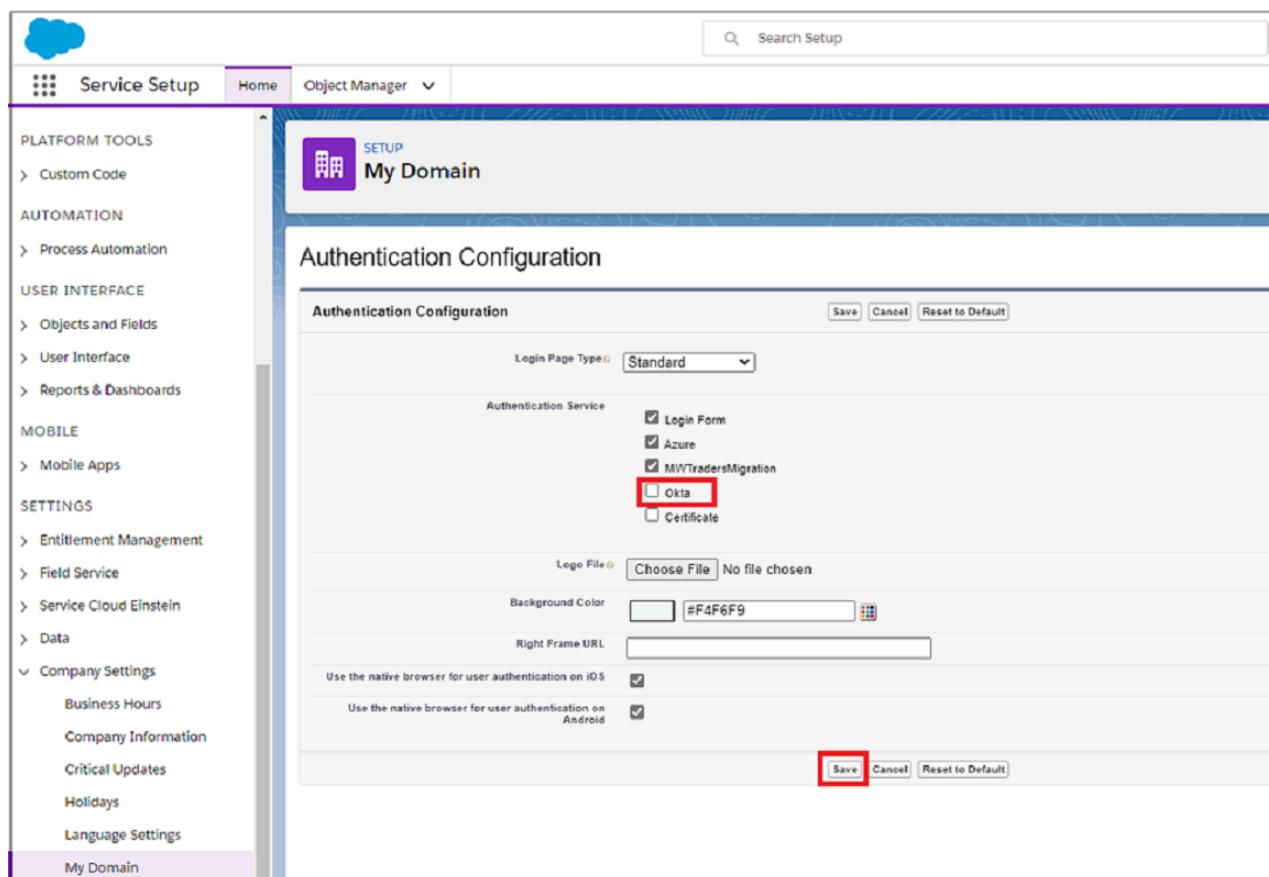
18. On the **Enterprise applications** page, assign the remaining users to the Salesforce application, with the correct roles.

! Note

After you add the remaining users to the Microsoft Entra application, users can test the connection to ensure they have access. Test the connection before the next step.

19. On the Salesforce administration console, select **Company Settings > My Domain**.

20. Under **Authentication Configuration**, select **Edit**. For authentication service, clear the selection for **Okta**.



Migrate an OpenID Connect or OAuth 2.0 application to Microsoft Entra ID

To migrate an OpenID Connect (OIDC) or OAuth 2.0 application to Microsoft Entra ID, in your Microsoft Entra tenant, configure the application for access. In this example, we convert a custom OIDC app.

To complete the migration, repeat configuration for all applications in the Okta tenant.

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Entra ID > Enterprise apps > All applications**.
3. Select **New application**.
4. Select **Create your own application**.
5. On the menu that appears, name the OIDC app and then select **Register an application you're working on to integrate with Microsoft Entra ID**.
6. Select **Create**.
7. On the next page, set up the tenancy of your application registration. For more information, see [Tenancy in Microsoft Entra ID](#). Go to **Accounts in any organizational directory (Any Microsoft Entra directory - Multitenant) > Register**.

... > [Salesforce | SAML-based Sign-on](#) > [SAML-based Sign-on](#) > [Enterprise applications | All applications](#) > [App registrations](#) > **Register an application** ...

* Name

The user-facing display name for this application (this can be changed later).

Hack4



Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (Contoso only - Single tenant)
- Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)
- Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

[Help me choose...](#)

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Select a platform



e.g. <https://example.com/auth>

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

By proceeding, you agree to the [Microsoft Platform Policies](#)

Register

8. On the **App registrations** page, under **Microsoft Entra ID**, open the created registration.

(!) Note

Depending on the application scenario, there are various configuration actions. Most scenarios require an app client secret.

8. On the **Overview** page, record the **Application (client) ID**. You use this ID in your application.
9. On the left, select **Certificates & secrets**. Then select **+ New client secret**. Name the client secret and set its expiration.
10. Record the value and ID of the secret.

(!) Note

If you misplace the client secret, you can't retrieve it. Instead, regenerate a secret.

11. On the left, select **API permissions**. Then grant the application access to the OIDC stack.
12. Select **+ Add permission** > **Microsoft Graph** > **Delegated permissions**.
13. In the **OpenId permissions** section, select **email**, **openid**, and **profile**. Then select **Add permissions**.
14. To improve user experience and suppress user consent prompts, select **Grant admin consent for Tenant Domain Name**. Wait for the **Granted** status to appear.

The screenshot shows the Azure portal interface for managing API permissions. The left sidebar shows navigation links like Home, Migration Tenant, Hack4, Overview, Quickstart, Integration assistant, Branding, Authentication, Certificates & secrets, Token configuration, API permissions (which is selected), Expose an API, App roles | Preview, Owners, Roles and administrators | Preview, Manifest, Support + Troubleshooting, Troubleshooting, and New support request. The main content area has a header with a search bar, refresh button, and feedback link. A message says "Successfully granted admin consent for the requested permissions." Below this, under "Configured permissions", it says "Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs." It shows a table with four rows for Microsoft Graph permissions: email, openid, profile, and User.Read. All have "Granted for Migration T..." status. There are three dots at the end of the table row.

API / Permissions name	Type	Description	Admin consent req...	Status
email	Delegated	View users' email address	No	Granted for Migration T...
openid	Delegated	Sign users in	No	Granted for Migration T...
profile	Delegated	View users' basic profile	No	Granted for Migration T...
User.Read	Delegated	Sign in and read user profile	No	Granted for Migration T...

15. If your application has a redirect URI, enter the URI. If the reply URL targets the **Authentication** tab, followed by **Add a platform** and **Web**, enter the URL.
16. Select **Access tokens and ID tokens**.
17. Select **Configure**.
18. If needed, on the **Authentication** menu, under **Advanced settings** and **Allow public client flows**, select **Yes**.

The screenshot shows the Azure portal's 'Hack4 | Authentication' configuration page. On the left, there's a sidebar with options like Overview, Quickstart, Integration assistant, Manage (with sub-options: Branding & properties, Authentication, Certificates & secrets, Token configuration, API permissions, Expose an API, App roles, Owners, Roles and administrators, Manifest), Support + Troubleshooting (Troubleshooting, New support request), and a 'Save' button at the bottom. The main content area is titled 'Platform configurations' and includes a note about redirect URLs and specific authentication settings. It features a 'Supported account types' section with a radio button for 'Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)' selected. Below that is an 'Advanced settings' section with a 'Allow public client flows' toggle set to 'Yes'. A 'Help me decide...' link is also present. At the bottom right is a 'Configure' button. The overall interface is clean and modern, typical of the Azure cloud management platform.

19. Before you test, in your OIDC-configured application, import the application ID and client secret.

! Note

Use the previous steps to configure your application with settings such as Client ID, Secret, and Scopes.

Migrate a custom authorization server to Microsoft Entra ID

Okta authorization servers map one-to-one to application registrations that [expose an API](#).

Map the default Okta authorization server to Microsoft Graph scopes or permissions.

The screenshot shows the Azure portal interface for managing an API. The left sidebar has a tree view with 'Expose an API' selected. A red box highlights 'Expose an API'. The main content area shows the 'Scopes defined by this API' section, which is currently empty. A red box highlights the '+ Add a scope' button. Below it is a table with columns: Scopes, Who can consent, Admin consent disp..., User consent displa..., and State. The table shows 'No scopes have been defined'. The 'Authorized client applications' section is also shown, indicating no client applications have been authorized.

Next steps

- Migrate Okta federation to Microsoft Entra ID
- Migrate Okta sync provisioning to Microsoft Entra Connect-based synchronization
- Migrate Okta sign-on policies to Microsoft Entra Conditional Access

Tutorial: Migrate Okta sign-on policies to Microsoft Entra Conditional Access

Article • 10/23/2023

In this tutorial, learn to migrate an organization from global or application-level sign-on policies in Okta Conditional Access in Microsoft Entra ID. Conditional Access policies secure user access in Microsoft Entra ID and connected applications.

Learn more: [What is Conditional Access?](#)

This tutorial assumes you have:

- Office 365 tenant federated to Okta for sign-in and multifactor authentication
- Microsoft Entra Connect server, or Microsoft Entra Connect cloud provisioning agents configured for user provisioning to Microsoft Entra ID

Prerequisites

See the following two sections for licensing and credentials prerequisites.

Licensing

There are licensing requirements if you switch from Okta sign-on to Conditional Access. The process requires a Microsoft Entra ID P1 license to enable registration for Microsoft Entra multifactor authentication.

Learn more: [Assign or remove licenses in the Microsoft Entra admin center](#)

Enterprise Administrator credentials

To configure the service connection point (SCP) record, ensure you have Enterprise Administrator credentials in the on-premises forest.

Evaluate Okta sign-on policies for transition

Locate and evaluate Okta sign-on policies to determine what will be transitioned to Microsoft Entra ID.

1. In Okta go to **Security > Authentication > Sign On**.

The screenshot shows the Okta Authentication interface. The top navigation bar has tabs for 'Password' and 'Sign On', with 'Sign On' being the active tab. A blue button labeled 'Add New Okta Sign-on Policy' is visible. On the left, a sidebar lists two policies: '1 Global MFA Sign On Policy' and '2 Default Policy'. The main content area displays the details for the 'Global MFA Sign On Policy'. It includes a description 'Enforce MFA for all Applications connected to Okta', a status indicator 'Inactive', and an 'Edit' or 'Delete' button. Below this, a table shows a single rule: 'Priority 1 Rule Name Enforce MFA Access Allowed Status Active'. Further down, configuration settings include 'Excludes Users' (No), 'Location' (Not in zone - All Zones), 'Login Type' (Any), 'Prompt For Factor' (Yes), 'Prompt Mode' (Per Session), 'Factor Lifetime' (15 minutes), and 'Session Lifetime' (12 hours).

2. Go to Applications.
3. From the submenu, select Applications
4. From the Active apps list, select the Microsoft Office 365 connected instance.

Microsoft Office 365

Active ▾

General Sign On Mobile Provisioning Import Assignments Push Groups

Settings Edit

Sign on methods

The sign-on method determines how a user signs into and manages their credentials for an application. Some sign-on methods require additional configuration in the 3rd party application.

Application username is determined by the user profile mapping. [Configure profile mapping](#)

Secure Web Authentication

WS-Federation

WS-Federation is not configured until you complete the setup instructions.

[View Setup Instructions](#)

WS-Federation Automatic (recommended)

About

WS-Federation allows for federated single sign-on to Office 365.

You can sync passwords to this app

If you enable password push, you can automatically synchronize Okta passwords to Microsoft Office 365.

Application Username

Choose a format to use as the default username value when assigning the application to users.

If you select **None** you will be prompted to enter the username manually when assigning an application with password or profile push provisioning features.

5. Select Sign On.

6. Scroll to the bottom of the page.

The Microsoft Office 365 application sign-on policy has four rules:

- **Enforce MFA for mobile sessions** - requires MFA from modern authentication or browser sessions on iOS or Android
- **Allow trusted Windows devices** - prevents unnecessary verification or factor prompts for trusted Okta devices
- **Require MFA from untrusted Windows devices** - requires MFA from modern authentication or browser sessions on untrusted Windows devices
- **Block legacy authentication** - prevents legacy authentication clients from connecting to the service

The following screenshot is conditions and actions for the four rules, on the Sign On Policy screen.

Sign On Policy

Priority	Rule name	Status	Actions
1	Enforce MFA for Mobile Sessions	Active	 
	CONDITIONS	ACTIONS	
	 In group: MFA Enforcement Group  Anywhere  Web browser, Modern Authentication on iOS, Android  Any	 Require multifactor every session	
2	Allow Trusted Windows Devices	Active	 
	CONDITIONS	ACTIONS	
	 In group: MFA Enforcement Group  Anywhere  Web browser, Modern Authentication on Windows  Trusted	 Allow access	
3	Require MFA from Untrusted Windows Devices	Active	 
	CONDITIONS	ACTIONS	
	 In group: MFA Enforcement Group  Anywhere  Web browser, Modern Authentication on Windows  Not trusted	 Require multifactor every session	
4	Block Legacy Authentication	Active	 
	CONDITIONS	ACTIONS	
	 In group: MFA Enforcement Group  Anywhere  Exchange ActiveSync on any mobile platform, any desktop platform  Any	 Deny access	

Configure Conditional Access policies

Configure Conditional Access policies to match Okta conditions. However, in some scenarios, you might need more setup:

- Okta network locations to named locations in Microsoft Entra ID

- Using the location condition in a Conditional Access policy
- Okta device trust to device-based Conditional Access (two options to evaluate user devices):
 - See the following section, **Microsoft Entra hybrid join configuration** to synchronize Windows devices, such as Windows 10, Windows Server 2016 and 2019, to Microsoft Entra ID
 - See the following section, **Configure device compliance**
 - See, [Use Microsoft Entra hybrid join](#), a feature in Microsoft Entra Connect server that synchronizes Windows devices, such as Windows 10, Windows Server 2016, and Windows Server 2019, to Microsoft Entra ID
 - See, [Enroll the device in Microsoft Intune](#) and assign a compliance policy

Microsoft Entra hybrid join configuration

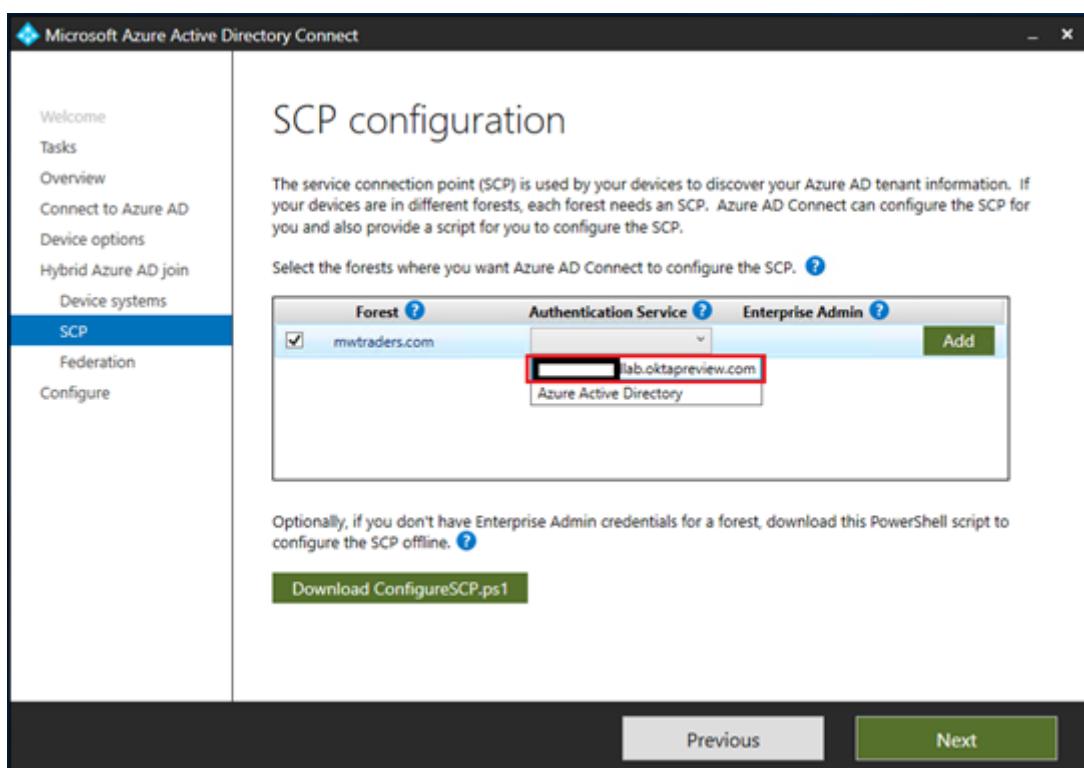
To enable Microsoft Entra hybrid join on your Microsoft Entra Connect server, run the configuration wizard. After configuration, enroll devices.

ⓘ Note

Microsoft Entra hybrid join isn't supported with the Microsoft Entra Connect cloud provisioning agents.

1. Configure Microsoft Entra hybrid join.

2. On the SCP configuration page, select the Authentication Service dropdown.



3. Select an Okta federation provider URL.
4. Select **Add**.
5. Enter your on-premises Enterprise Administrator credentials
6. Select **Next**.

 **Tip**

If you blocked legacy authentication on Windows clients in the global or app-level sign-on policy, make a rule that enables the Microsoft Entra hybrid join process to finish. Allow the legacy authentication stack for Windows clients. To enable custom client strings on app policies, contact the [Okta Help Center](#).

Configure device compliance

Microsoft Entra hybrid join is a replacement for Okta device trust on Windows. Conditional Access policies recognize compliance for devices enrolled in Microsoft Intune.

Device compliance policy

- [Use compliance policies to set rules for devices you manage with Intune](#)
- [Create a compliance policy in Microsoft Intune](#)

Windows 10/11, iOS, iPadOS, and Android enrollment

If you deployed Microsoft Entra hybrid join, you can deploy another group policy to complete auto-enrollment of these devices in Intune.

- [Enrollment in Microsoft Intune](#)
- [Quickstart: Set up automatic enrollment for Windows 10/11 devices](#)
- [Enroll Android devices](#)
- [Enroll iOS/iPadOS devices in Intune](#)

Configure Microsoft Entra multifactor authentication tenant settings

💡 Tip

Steps in this article may vary slightly based on the portal you start from.

Before you convert to Conditional Access, confirm the base MFA tenant settings for your organization.

1. Sign in to the [Microsoft Entra admin center](#) as a **Global Administrator**.
2. Browse to **Identity > Users > All users**.
3. Select **Per-user MFA** on the top menu of the **Users** pane.
4. The legacy Microsoft Entra multifactor authentication portal appears. Or select [Microsoft Entra multifactor authentication portal](#).

The screenshot shows the Microsoft Entra multifactor authentication portal. At the top, there's a navigation bar with the Microsoft logo and the email address 'admin@mwtraders.onmicrosoft.com'. Below the navigation, the title 'multi-factor authentication' is displayed, followed by 'users service settings'. A note says 'Before you begin, take a look at the [multi-factor auth deployment guide](#)'. There are filter options 'View: Sign-in allowed users' and 'Multi-Factor Auth status: Enabled', along with a 'bulk update' button. The main table has columns for 'DISPLAY NAME', 'USER NAME', and 'MULTI-FACTOR AUTH STATUS'. A message 'There are no users to display.' is shown. On the right, a button labeled 'Select a user' is visible.

5. Confirm there are no users enabled for legacy MFA: On the **Multifactor authentication** menu, on **Multifactor authentication status**, select **Enabled** and **Enforced**. If the tenant has users in the following views, disable them in the legacy menu.

The screenshot shows a Microsoft web interface for managing multi-factor authentication (MFA) users. At the top, there's a navigation bar with the Microsoft logo and a user email address: admin@mvtraders.onmicrosoft.com. Below the header, the title "multi-factor authentication" and "users service settings" is displayed. A note says, "Before you begin, take a look at the [multi-factor auth deployment guide](#)." There are filter options: "View: Sign-in allowed users" with a search icon, "Multi-Factor Auth status: Enforced" with a dropdown arrow, and a "bulk update" button. The main table has columns: DISPLAY NAME, USER NAME, and MULTI-FACTOR AUTH STATUS. A message in the table body says, "There are no users to display." To the right of the table, a large blue button says "Select a user". At the bottom left, there's a copyright notice: "©2021 Microsoft Legal | Privacy".

6. Ensure the **Enforced** field is empty.
7. Select the **Service settings** option.
8. Change the **App passwords** selection to **Do not allow users to create app passwords to sign in to non-browser apps**.

Microsoft admin@mwtraders.onmicrosoft.com | ?

multi-factor authentication

users **service settings**

app passwords ([learn more](#))

Allow users to create app passwords to sign in to non-browser apps
 Do not allow users to create app passwords to sign in to non-browser apps

trusted ips ([learn more](#))

Skip multi-factor authentication for requests from federated users on my intranet

Skip multi-factor authentication for requests from following range of IP address subnets

192.168.1.0/27
192.168.1.0/27
192.168.1.0/27

verification options ([learn more](#))

Methods available to users:

Call to phone
 Text message to phone
 Notification through mobile app
 Verification code from mobile app or hardware token

remember multi-factor authentication on trusted device ([learn more](#))

Allow users to remember multi-factor authentication on devices they trust (between one to 365 days)
Number of days users can trust devices for **90**

NOTE: For the optimal user experience, we recommend using Conditional Access sign-in frequency to extend session lifetimes on trusted devices, locations, or low-risk sessions as an alternative to 'Remember MFA on a trusted device' settings. If using 'Remember MFA on a trusted device,' be sure to extend the duration to 90 or more days. [Learn more about reauthentication prompts.](#)

save

9. Clear the checkboxes for **Skip multifactor authentication for requests from federated users on my intranet** and **Allow users to remember multifactor authentication on devices they trust (between one to 365 days)**.

10. Select **Save**.

Require Trusted Devices for Access

Conditional access policy

 Delete

Control user access based on conditional access policy to bring signals together, to make decisions, and enforce organizational policies. [Learn more](#)

Control user access based on all or specific cloud apps or actions. [Learn more](#)

Select what this policy applies to

Cloud apps User actions

Name *

Require Trusted Devices for Access

Include Exclude

- None
- All cloud apps
- Select apps

Select

Office 365



Assignments

Users and groups [\(1\)](#)



Specific users included

Cloud apps or actions [\(1\)](#)



1 app included

Conditions [\(1\)](#)



1 condition selected

Access controls

Grant [\(1\)](#)



1 control selected

Session [\(1\)](#)



0 controls selected

Enable policy

Report-only On Off

Save

Note

See [Optimize reauthentication prompts and understand session lifetime for Microsoft Entra multifactor authentication](#).

Build a Conditional Access policy

To configure Conditional Access policies, see [Best practices for deploying and designing Conditional Access](#).

After you configure the prerequisites and established base settings, you can build Conditional Access policy. Policy can be targeted to an application, a test group of users, or both.

Before you get started:

- [Understand Conditional Access policy components](#)
- [Building a Conditional Access policy](#)

1. Sign in to the [Microsoft Entra admin center](#).
2. Browse to **Identity**.
3. To learn how to create a policy in Microsoft Entra ID. See, [Common Conditional Access policy: Require MFA for all users](#).
4. Create a device trust-based Conditional Access rule.

Require Trusted Devices for Access

Conditional access policy



Control user access based on conditional access policy to bring signals together, to make decisions, and enforce organizational policies. [Learn more](#)

Name *

Require Trusted Devices for Access

Assignments

Users and groups ⓘ



Specific users included

Cloud apps or actions ⓘ



1 app included

Conditions ⓘ



1 condition selected

Access controls

Grant ⓘ



1 control selected

Session ⓘ



0 controls selected

Control user access based on users and groups assignment for all users, specific groups of users, directory roles, or external guest users

[Learn more](#)

Include Exclude

None

All users

Select users and groups

All guest and external users ⓘ

Directory roles ⓘ

Users and groups

Select

1 group

EM

Enforce MFA

Enable policy

Report-only On Off

Save

Keep your account secure

Your organization requires you to set up the following methods of proving who you are.

Success!

Great job! You have successfully set up your security info. Choose "Done" to continue signing in.

Default sign-in method: Microsoft Authenticator - notification



Microsoft Authenticator

Done

5. After you configure the location-based policy and device trust policy, [Block legacy authentication with Microsoft Entra ID with Conditional Access](#).

With these three Conditional Access policies, the original Okta sign-on policies experience is replicated in Microsoft Entra ID.

Enroll pilot members in MFA

Users register for MFA methods.

For individual registration, users go to [Microsoft Sign-in pane](#).

To manage registration, users go to [Microsoft My Sign-Ins | Security Info](#).

Learn more: [Enable combined security information registration in Microsoft Entra ID](#).

ⓘ Note

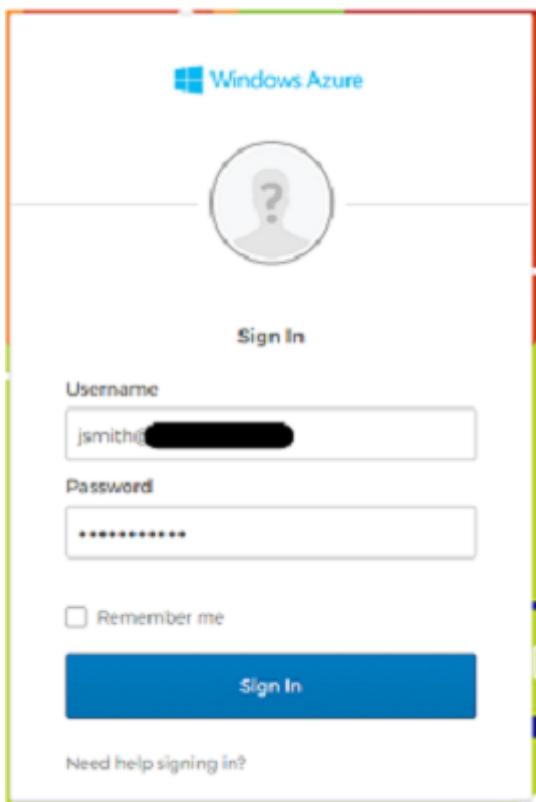
If users registered, they're redirected to the [My Security](#) page, after they satisfy MFA.

Enable Conditional Access policies

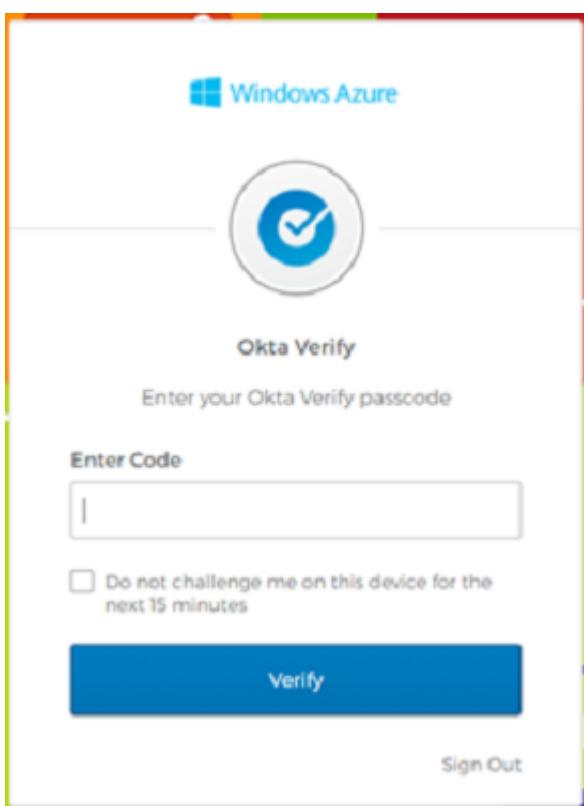
1. To test, change the created policies to **Enabled test user login**.

Policy Name	State
Require MFA from non-trusted networks	On
Require MFA from Untrusted Devices	On
Block Legacy Authentication	On

2. On the Office 365 **Sign-In** pane, the test user John Smith is prompted to sign in with Okta MFA and Microsoft Entra multifactor authentication.

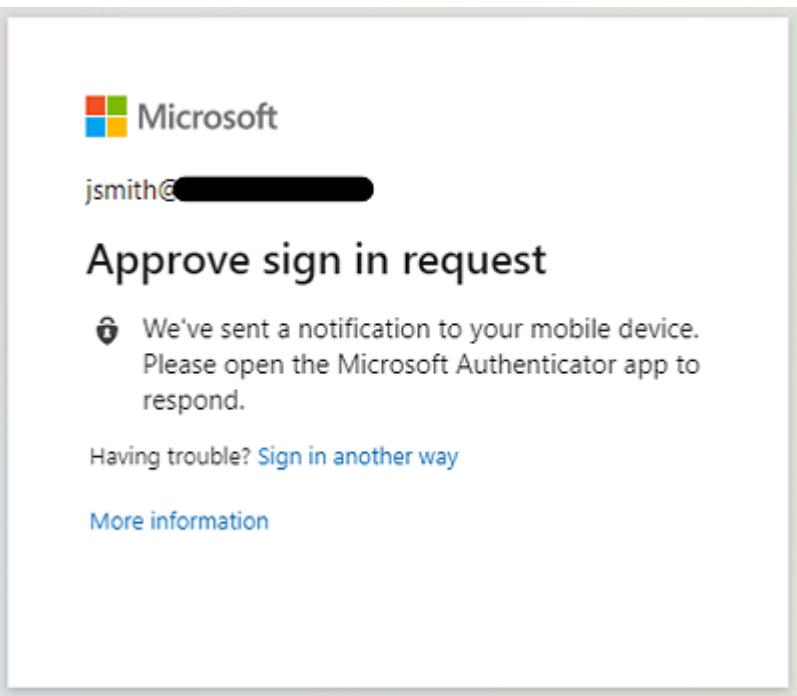


3. Complete the MFA verification through Okta.



4. The user is prompted for Conditional Access.

5. Ensure the policies are configured to be triggered for MFA.



Add organization members to Conditional Access policies

After you conduct testing on pilot members, add the remaining organization members to Conditional Access policies, after registration.

To avoid double-prompts between Microsoft Entra multifactor authentication and Okta MFA, opt out from Okta MFA: modify sign-on policies.

1. Go to the Okta admin console
2. Select **Security > Authentication**
3. Go to **Sign-on Policy**.

Note

Set global policies to **Inactive** if all applications from Okta are protected by application sign-on policies.

4. Set the **Enforce MFA** policy to **Inactive**. You can assign the policy to a new group that doesn't include the Microsoft Entra users.

The screenshot shows the Okta Authentication interface under the 'Sign On' tab. A blue box highlights the 'Add New Okta Sign-on Policy' button. On the left, a sidebar lists two policies: 'Global MFA Sign On Policy' (selected) and 'Default Policy'. The main pane displays the 'Global MFA Sign On Policy' details. It includes a status dropdown set to 'Inactive' (highlighted with a yellow box), a 'Description' field ('Enforce MFA for all Applications connected to Okta'), and an 'Assigned to groups' section ('Everyone'). An 'Add Rule' button is present. A table lists one rule: 'Priority 1, Rule Name Enforce MFA, Access Allowed, Status Active'.

5. On the application-level sign-on policy pane, select the **Disable Rule** option.
6. Select **Inactive**. You can assign the policy to a new group that doesn't include the Microsoft Entra users.
7. Ensure there's at least one application-level sign-on policy enabled for the application that allows access without MFA.

3	 Require MFA from Untrusted Windows Devices	Disabled	
CONDITIONS			
 In group: MFA Enforcement Group  Anywhere  Web browser, Modern Authentication on Windows  Not trusted			
ACTIONS			
 Require multifactor every session			
4	 Block Legacy Authenticaiton	Disabled	
CONDITIONS			
 In group: MFA Enforcement Group  Anywhere  Exchange ActiveSync on any mobile platform, any desktop platform  Any			
ACTIONS			
 Deny access			
5	 Allow Sign on to Office 365	Active	
CONDITIONS			
 User assigned this app  Anywhere  Any client  Any			
ACTIONS			
 Allow access			

8. Users are prompted for Conditional Access the next time they sign in.

Next steps

- Tutorial: Migrate your applications from Okta to Microsoft Entra ID
- Tutorial: Migrate Okta federation to Microsoft Entra ID-managed authentication
- Tutorial: Migrate Okta sync provisioning to Microsoft Entra Connect-based synchronization

Tutorial: Migrate Okta sync provisioning to Microsoft Entra Connect synchronization

Article • 12/04/2024

In this tutorial, learn to migrate user provisioning from Okta to Microsoft Entra ID and migrate User Sync or Universal Sync to Microsoft Entra Connect. This capability enables provisioning into Microsoft Entra ID and Office 365.

ⓘ Note

When migrating synchronization platforms, validate steps in this article against your environment before you remove Microsoft Entra Connect from staging mode or enable the Microsoft Entra cloud provisioning agent.

Prerequisites

When you switch from Okta provisioning to Microsoft Entra ID, there are two choices. Use a Microsoft Entra Connect server or Microsoft Entra cloud provisioning.

Learn more: [Comparison between Microsoft Entra Connect and cloud sync](#).

Microsoft Entra cloud provisioning is the most familiar migration path for Okta customers who use Universal Sync or User Sync. The cloud provisioning agents are lightweight. You can install them on, or near, domain controllers like the Okta directory sync agents. Don't install them on the same server.

When you synchronize users, use a Microsoft Entra Connect server if your organization needs any of the following technologies:

- Device synchronization: Microsoft Entra hybrid join or Hello for Business
- Pass-through authentication
- Support for more than 150,000 objects
- Support for writeback

To use Microsoft Entra Connect, you need to sign in with a Hybrid Identity Administrator role.

ⓘ Note

Take all prerequisites into consideration when you install Microsoft Entra Connect or Microsoft Entra cloud provisioning. Before you continue with installation, see [Prerequisites for Microsoft Entra Connect](#).

Confirm ImmutableID attribute synchronized by Okta

The ImmutableID attribute ties synchronized objects to their on-premises counterparts. Okta takes the Active Directory objectGUID of an on-premises object and converts it to a Base-64-encoded string. By default, it then stamps that string to the ImmutableID field in Microsoft Entra ID.

You can connect to Microsoft Graph PowerShell and examine the current ImmutableID value. If you haven't used the Microsoft Graph PowerShell module, run:

```
Install-Module AzureAD
```

in an administrative session before you run the following commands:

```
Powershell
```

```
Import-Module AzureAD  
Connect-MgGraph
```

If you have the module, a warning might appear to update to the latest version.

1. Import the installed module.
2. In the authentication window, sign in as at least a [Hybrid Identity Administrator](#).
3. Connect to the tenant.
4. Verify ImmutableID value settings. The following example is the default of converting the objectGUID into the ImmutableID.
5. Manually confirm the conversion from objectGUID to Base64 on-premises. To test an individual value, use these commands:

```
PowerShell
```

```
Get-MgUser onpremupn | fl objectguid  
$objectguid = 'your-guid-here-1010'  
[System.Convert]::ToBase64String(([GUID]$objectGUID).ToArray())
```

ObjectGUID mass-validation methods

Before you move to Microsoft Entra Connect, it's critical to validate that the ImmutableID values in Microsoft Entra ID match their on-premises values.

The following command gets on-premises Microsoft Entra users and exports a list of their objectGUID values and ImmutableID values already calculated to a CSV file.

1. Run the following command in Microsoft Graph PowerShell on an on-premises domain controller:

PowerShell

```
Get-ADUser -Filter * -Properties objectGUID | Select-Object
UserPrincipalName, Name, objectGUID, @{Name = 'ImmutableID';
Expression = {
[System.convert]::ToBase64String((GUID).tobytearray())
} } | export-csv C:\Temp\OnPremIDs.csv
```

2. Run the following command in a Microsoft Graph PowerShell session to list the synchronized values:

PowerShell

```
Get-MgUser -all $true | Where-Object {$_.dirsyncenabled -like
"true"} | Select-Object UserPrincipalName, @{Name = 'objectGUID';
Expression = {
[GUID][System.Convert]::FromBase64String($_.ImmutableID) } },
ImmutableID | export-csv C:\\temp\\AzureADSyncedIDS.csv
```

3. After both exports, confirm user ImmutableID values match.

 **Important**

If your ImmutableID values in the cloud don't match objectGUID values, you've modified the defaults for Okta sync. You've likely chosen another attribute to determine ImmutableID values. Before going the next section, identify which source attribute populates ImmutableID values. Before you disable Okta sync, update the attribute Okta is syncing.

Install Microsoft Entra Connect in staging mode

After you prepare your list of source and destination targets, install a Microsoft Entra Connect server. If you use Microsoft Entra Connect cloud provisioning, skip this section.

1. Download and install Microsoft Entra Connect on a server. See, [Custom installation of Microsoft Entra Connect](#).
2. In the left panel, select **Identifying users**.
3. On the **Uniquely identifying your users** page, under **Select how users should be identified with Microsoft Entra ID**, select **Choose a specific attribute**.
4. If you didn't modify the Okta default, select **mS-DS-ConsistencyGUID**.

 **Warning**

This step is critical. Ensure the attribute you select for a source anchor populates your Microsoft Entra users. If you select the wrong attribute, uninstall and reinstall Microsoft Entra Connect to reselect this option.

5. Select **Next**.
6. In the left panel, select **Configure**.
7. On the **Ready to configure** page, select **Enable staging mode**.
8. Select **Install**.
9. Verify the **ImmutableID** values match.
10. When the configuration is complete, select **Exit**.
11. Open **Synchronization Service** as an administrator.
12. Find the **Full Synchronization** to the domain.onmicrosoft.com connector space.
13. Confirm there are users under the **Connectors with Flow Updates** tab.
14. Verify no pending deletions in the export.
15. Select the **Connectors** tab.
16. Highlight the domain.onmicrosoft.com connector space.
17. Select **Search Connector Space**.
18. In the **Search Connector Space** dialog, under **Scope**, select **Pending Export**.

19. Select **Delete**.
20. Select **Search**. If all objects match, no matching records appear for **Deletes**.
21. Record objects pending deletion and their on-premises values.
22. Clear **Delete**.
23. Select **Add**.
24. Select **Modify**.
25. Select **Search**.
26. Update functions appear for users being synchronized to Microsoft Entra ID via Okta. Add new objects Okta isn't syncing, which are in the organizational unit (OU) structure selected during Microsoft Entra Connect installation.
27. To see what Microsoft Entra Connect communicates with Microsoft Entra ID, double-click an update.

 **Note**

If there are **add** functions for a user in Microsoft Entra ID, their on-premises account doesn't match the cloud account. Entra Connect creates a new object and records new and unexpected adds.

28. Before you exit the staging mode, correct the **ImmutableID** value in Microsoft Entra ID.

In this example, Okta stamped the **mail** attribute to the user's account, although the on-premises value wasn't accurate. When Microsoft Entra Connect takes over the account, the **mail** attribute is deleted from the object.

29. Verify updates include attributes expected in Microsoft Entra ID. If multiple attributes are being deleted, you can populate on-premises AD values before you remove the staging mode.

 **Note**

Before you continue, ensure user attributes are syncing and appear on the **Pending Export** tab. If they're deleted, ensure the **ImmutableID** values match and the user is in a selected OU for synchronization.

Install Microsoft Entra Connect cloud sync agents

After you prepare your list of source and destination targets, install and configure Microsoft Entra Connect cloud sync agents. See, [Tutorial: Integrate a single forest with a single Microsoft Entra tenant](#).

ⓘ Note

If you use a Microsoft Entra Connect server, skip this section.

Disable Okta provisioning to Microsoft Entra ID

After you verify the Microsoft Entra Connect installation, disable Okta provisioning to Microsoft Entra ID.

1. Go to the Okta portal
2. Select **Applications**.
3. Select the Okta app that provisions users to Microsoft Entra ID.
4. Select the **Provisioning** tab.
5. Select the **Integration** section.

The screenshot shows the Okta application integration settings for the Microsoft Office 365 app. The navigation bar at the top includes 'Back to Applications', the app logo, and tabs for 'Active' (selected), 'View Logs', and 'Monitor Imports'. Below the tabs are links for 'General', 'Sign On', 'Mobile', 'Provisioning' (which is underlined in blue), 'Import', 'Assignments', and 'Push Groups'. The 'Provisioning' tab is active, showing the 'Integration' section. Under 'Integration', there is a checkbox for 'Enable API integration' which is checked. A note below says 'Enter your Microsoft Office 365 credentials to enable user import and provisioning features.' with a 'Test API Credentials' button. On the left, a sidebar has sections for 'Settings', 'To App', 'To Okta', and 'Integration', with 'Integration' currently selected. On the right, there is an 'Edit' link and fields for 'Admin Username' (redacted) and 'Admin Password' (redacted). At the bottom, there is a checkbox for 'Import Groups'.

6. Select **Edit**.

7. Clear the **Enable API integration** option.

8. Select **Save**.

The screenshot shows the Microsoft Office 365 application configuration interface. At the top, there's a navigation bar with icons for Back to Applications, Active (dropdown), and several other management tools like Lock, Sync, and Import. Below the navigation is a tabs menu with General, Sign On, Mobile, Provisioning (which is underlined in blue), Import, Assignments, and Push Groups. On the left, a sidebar has two sections: Settings (selected) and Integration (highlighted with a light blue background). In the main content area, a message reads "Provisioning is not enabled" followed by a sub-message: "Enable provisioning to automate Microsoft Office 365 user account creation, deactivation, and updates." At the bottom right of this area is a blue rectangular button labeled "Configure API Integration".

ⓘ Note

If you have multiple Office 365 apps that handle provisioning to Microsoft Entra ID, ensure they switched off.

Disable staging mode in Microsoft Entra Connect

After you disable Okta provisioning, the Microsoft Entra Connect server can synchronize objects.

ⓘ Note

If you use Microsoft Entra Connect cloud sync agents, skip this section.

1. From the desktop, run the installation wizard from the desktop.
2. Select **Configure**.
3. Select **Configure staging mode**
4. Select **Next**.

5. Enter the credentials of the Hybrid Identity Administrator account for your environment.
6. Clear **Enable staging mode**.
7. Select **Next**.
8. Select **Configure**.
9. After configuration, open the **Synchronization Service** as an administrator.
10. On the domain.onmicrosoft.com connector, view the **Export**.
11. Verify additions, updates, and deletions.
12. Migration is complete. Rerun the installation wizard to update and expand Microsoft Entra Connect features.

Enable cloud sync agents

Tip

Steps in this article might vary slightly based on the portal you start from.

After you disable Okta provisioning, the Microsoft Entra Connect cloud sync agent can synchronize objects.

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Hybrid Identity Administrator**.
2. Browse to **Identity > Hybrid management > Microsoft Entra Connect > Connect Sync**.
3. Select **Configuration** profile.
4. Select **Enable**.
5. Return to the provisioning menu and select **Logs**.
6. Confirm the provisioning connector updated in-place objects. The cloud sync agents are nondestructive. Updates fail if a match isn't found.
7. If a user is mismatched, make updates to bind the **ImmutableID** values.
8. Restart the cloud provisioning sync.

Next steps

- [Tutorial: Migrate your applications from Okta to Microsoft Entra ID](#)
- [Tutorial: Migrate Okta federation to Microsoft Entra ID managed authentication](#)
- [Tutorial: Migrate Okta sign-on policies to Conditional Access](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Migrate Okta federation to Microsoft Entra authentication

Article • 12/06/2024

In this tutorial, learn to federate Office 365 tenants with Okta for single sign-on (SSO).

You can migrate federation to Microsoft Entra ID in a staged manner to ensure a good authentication experience for users. In a staged migration, you can test reverse federation access to remaining Okta SSO applications.

ⓘ Note

Scenario described in this tutorial is only one possible way of implementing the migration. You should try to adapt the information to your specific setup.

Prerequisites

- An Office 365 tenant federated to Okta for SSO
- A Microsoft Entra Connect server or Microsoft Entra Connect cloud provisioning agents configured for user provisioning to Microsoft Entra ID
- One of the following roles: Application Administrator, Cloud Application Administrator, or Hybrid Identity Administrator.

Configure Microsoft Entra Connect for authentication

Customers that federate their Office 365 domains with Okta might not have a valid authentication method in Microsoft Entra ID. Before you migrate to managed authentication, validate Microsoft Entra Connect and configure it for user sign-in.

Set up the sign-in method:

- **Password hash synchronization** - an extension of the directory synchronization feature implemented by Microsoft Entra Connect server or cloud-provisioning agents
 - Use this feature to sign in to Microsoft Entra services like Microsoft 365
 - Sign in to the service with the password to sign in to the on-premises Active Directory instance

- See, [What is password hash synchronization with Microsoft Entra ID?](#)
- **Pass-through authentication** - sign in to on-premises and cloud applications with the same passwords
 - When users sign in through Microsoft Entra ID, the pass-through authentication agent validates passwords against the on-premises AD
 - See, [User sign-in with Microsoft Entra pass-through authentication](#)
- **Seamless SSO** - signs in users on corporate desktops connected to the corporate network
 - Users have access to cloud applications without other on-premises components
 - See, [Microsoft Entra seamless SSO](#)

To create a seamless authentication user experience in Microsoft Entra ID, deploy seamless SSO to password hash synchronization or pass-through authentication.

For prerequisites of seamless SSO see, [Quickstart: Microsoft Entra seamless single sign-on](#).

For this tutorial, you configure password hash synchronization and seamless SSO.

Configure Microsoft Entra Connect for password hash synchronization and seamless SSO

1. On the Microsoft Entra Connect server, open the [Microsoft Entra Connect app](#).
2. Select **Configure**.
3. Select **Change user sign-in**.
4. Select **Next**.
5. Enter the credentials of the Hybrid Identity Administrator of the Microsoft Entra Connect server.
6. The server is configured for federation with Okta. Change the selection to **Password Hash Synchronization**.
7. Select **Enable single sign-on**.
8. Select **Next**.
9. For the local on-premises system, enter the domain administrator credentials.
10. Select **Next**.
11. On the final page, select **Configure**.
12. Ignore the warning for Microsoft Entra hybrid join.

Configure staged rollout features

 **Tip**

Steps in this article might vary slightly based on the portal you start from.

Before you test defederating a domain, in Microsoft Entra ID use a cloud authentication staged rollout to test defederating users.

Learn more: [Migrate to cloud authentication using Staged Rollout](#)

After you enable password hash sync and seamless SSO on the Microsoft Entra Connect server, configure a staged rollout:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Hybrid Identity Administrator](#).
2. Browse to **Identity > Hybrid management > Microsoft Entra Connect > Connect Sync**.
3. Confirm **Password Hash Sync** is enabled in the tenant.
4. Select **Enable staged rollout for managed user sign-in**.
5. After the server configuration, **Password Hash Sync** setting can change to **On**.
6. Enable the setting.
7. **Seamless single sign-on** is **Off**. If you enable it, an error appears because you enabled it in the tenant.
8. Select **Manage groups**.

Home > MW Traders >

Enable staged rollout features (Preview)

[Troubleshoot](#) | [Got feedback?](#)

This feature is intended to help you transition from federation to cloud authentication. When your transition is complete, please change the tenant wide authentication method to cloud authentication. [Learn more](#).

Password Hash Sync ⓘ

[On](#) [Off](#) [Manage groups](#)

Pass-through authentication ⓘ

[On](#) [Off](#) [Manage groups](#)

Seamless single sign-on ⓘ

[On](#) [Off](#) [Manage groups](#)

9. Add a group to the password hash sync rollout.
10. Wait about 30 minutes for the feature to take effect in your tenant.
11. When the feature takes effect, users aren't redirected to Okta when attempting to access Office 365 services.

The staged rollout feature has some unsupported scenarios:

- Legacy authentication protocols such as Post Office Protocol 3 (POP3) and Simple Mail Transfer Protocol (SMTP) aren't supported.
- If you configured Microsoft Entra hybrid join for Okta, the Microsoft Entra hybrid join flows go to Okta until the domain is defederated.
 - A sign-on policy remains in Okta for legacy authentication of Microsoft Entra hybrid join Windows clients.

Create an Okta app in Microsoft Entra ID

Users that converted to managed authentication might need access to applications in Okta. For user access to those applications, register a Microsoft Entra application that links to the Okta home page.

Configure the enterprise application registration for Okta.

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.

Enterprise applications | Overview

Contoso - Microsoft Entra ID for workforce

Overview

 Overview

 Diagnose and solve problems

Manage

 All applications

 Private Network connectors

 User settings

 App launchers

 Custom authentication
extensions (Preview)

Security

 Conditional Access

 Consent and permissions

Activity



 New application

Overview

Tutorials

Search your tenant

Basic information

Total applications

Enterprise applications

Microsoft applications

My feed



Conditional Access

Control user access policy to bring and enforce controls.

3. Select New application.

The screenshot shows the Microsoft Entra Connect | Connect Sync interface. At the top, there's a breadcrumb navigation: Home > Microsoft Entra Connect | Connect Sync >. Below it is a header with a blue circular icon containing a white 'i', followed by the text 'Enterprise applications | Overview' and 'Contoso - Microsoft Entra ID for workforce'. To the right of the header are three dots (...). On the far right of the header bar are two buttons: '+ New application' with a plus sign and 'Got feedback?' with a person icon.

The main content area has a left sidebar with two sections: 'Overview' and 'Manage'. Under 'Overview', the 'Overview' item is selected and highlighted with a grey background. Other items include 'Diagnose and solve problems'. Under 'Manage', the 'All applications' item is listed. To the right of the sidebar is a central panel. At the top of the central panel are two tabs: 'Overview' (which is underlined) and 'Tutorials'. Below the tabs is a search bar with a magnifying glass icon and the placeholder text 'Search your tenant'. Further down is a section titled 'Basic information'.

4. Select **Create your own application**.
5. On the menu, name the Okta app.
6. Select **Register an application you're working on to integrate with Microsoft Entra ID**.
7. Select **Create**.
8. Select **Accounts in any organizational directory (Any Microsoft Entra Directory - Multitenant)**.
9. Select **Register**.

Register an application

* Name

The user-facing display name for this application (this can be changed later).

Okta Application Access



Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (Contoso only - Single tenant)
- Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)
- Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

[Help me choose...](#)

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web



e.g. https://example.com/auth



Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

By proceeding, you agree to the [Microsoft Platform Policies](#)

[Register](#)

10. On the Microsoft Entra ID menu, select **App registrations**.

11. Open the created registration.

Display name	Application (client) ID	Created on	Certificates & secrets
Okta Application Access	11111111-1111-1111-1111-111111111111	3/29/2024	-

12. Record the Tenant ID and Application ID.

ⓘ Note

You need the Tenant ID and Application ID to configure the identity provider in Okta.

The screenshot shows the Microsoft Entra admin center interface. On the left, there's a navigation menu with sections like Home, Favorites, Identity, Applications, App registrations, Roles & admins, Billing, Settings, Protection, Identity governance, External Identities, User experiences, Hybrid management, and Learn & support. The 'App registrations' section is currently selected. In the main content area, the title is 'Okta Application Access'. Under the 'Overview' tab, there's a 'Essentials' section with fields for Display name (Okta Application Access), Application (client) ID (1111111-1111-1111-1111-111111111111), Object ID (1111111-1111-1111-1111-111111111111), and Directory (tenant) ID (1111111-1111-1111-1111-111111111111). Below these, it says 'Supported account types: Multiple organizations'. At the bottom of the overview page, there are 'Get Started' and 'Documentation' links, and a section titled 'Build your application with the Microsoft identity platform'.

13. On the left menu, select **Certificates & secrets**.
14. Select **New client secret**.
15. Enter a secret name.
16. Enter its expiration date.
17. Record the secret value and ID.

(!) Note

The value and ID don't appear later. If you don't record the information, you must regenerate a secret.

18. On the left menu, select **API permissions**.
19. Grant the application access to the OpenID Connect (OIDC) stack.
20. Select **Add a permission**.
21. Select **Microsoft Graph**
22. Select **Delegated permissions**.
23. In the OpenID permissions section, add **email**, **openid**, and **profile**.
24. Select **Add permissions**.

25. Select **Grant admin consent** for <tenant domain name>.

26. Wait for the **Granted** status to appear.

The screenshot shows the Okta Application Access interface. The left sidebar has a 'Manage' section with 'API permissions' selected. The main area displays 'Configured permissions' for 'Microsoft Graph'. A table lists four permissions: 'email', 'openid', 'profile', and 'User.Read', all marked as 'Granted for MW Traders'. A success message at the top states 'Successfully granted admin consent for the requested permissions.'

API / Permissions name	Type	Description	Admin consent req...	Status
email	Delegated	View users' email address	-	Granted for MW Traders
openid	Delegated	Sign users in	-	Granted for MW Traders
profile	Delegated	View users' basic profile	-	Granted for MW Traders
User.Read	Delegated	Sign in and read user profile	-	Granted for MW Traders

27. On the left menu, select **Branding**.

28. For **Home page URL**, add your user application home page.

29. In the Okta administration portal, to add a new identity provider, select **Security** then **Identity Providers**.

30. Select **Add Microsoft**.

Dashboard

Directory

Applications

Security

General

HealthInsight

Authentication

Multifactor

Identity Providers

Delegated Authentication

Networks

🔒 Identity Providers

Identity Providers

Routing R

[+ Add Identity Provider](#)[Add Facebook](#)[Add Google](#)[Add LinkedIn](#)[Add Microsoft](#)[Add Apple](#)[Add OpenID Connect](#)[IdP](#)[Add SAML 2.0 IdP](#)

31. On the **Identity Provider** page, enter the Application ID in the **Client ID** field.
32. Enter the client secret in the **Client Secret** field.
33. Select **Show Advanced Settings**. By default, this configuration ties the user principal name (UPN) in Okta to the UPN in Microsoft Entra ID for reverse-federation access.

ⓘ Important

If UPNs in Okta and Microsoft Entra ID don't match, select an attribute that's common between users.

34. Complete autoprovisioning selections.
35. By default, if no match appears for an Okta user, the system attempts to provision the user in Microsoft Entra ID. If you migrated provisioning away from Okta, select **Redirect to Okta sign-in page**.

General Settings

View the Microsoft [Setup Instructions](#)

Name

MW Traders Tenant

Client ID

MySuperNotSoSecretClientID

Client Secret

.....

Scopes

<https://graph.microsoft.com/User.Read> x

[email](#) x

[openid](#) x

[profile](#) x

[Microsoft Scopes](#)

Protocol

OIDC

[Hide Advanced Settings](#)

IdP Username [?](#)

idpuser.userPrincipalName

[Expression Language Reference](#)

Match against [?](#)

Okta Username

Choose the user attribute to match against the IdP username.

Account Link Policy [?](#)

Automatic

Auto-Link Restrictions [?](#)

None

If no match is found [?](#)

Create new user (JIT)

Redirect to Okta sign-in page

[Update Identity Provider](#)

[Cancel](#)

You created the identity provider (IDP). Send users to the correct IDP.

1. On the **Identity Providers** menu, select **Routing Rules** then **Add Routing Rule**.
2. Use one of the available attributes in the Okta profile.
3. To direct sign-ins from devices and IPs to Microsoft Entra ID, set up the policy seen in following image. In this example, the **Division** attribute is unused on all Okta

profiles. It's a good choice for IDP routing.

4. Record the redirect URI to add it to the application registration.



A screenshot of the Okta Application Registration settings page. The application is named "MW Traders Tenant" and is configured for Microsoft SSO. The "Active" status is shown with a dropdown arrow, and there is a "Configure" button. The "IdP ID" is listed as "Ooawrefm1EYIf8nOh7". The "Authorize URL" field contains a placeholder URL with parameters like {client_id}, {response_type}, {scopes}, and {state}. The "Redirect URI" field is highlighted with a yellow box and contains the URL "https://[REDACTED].apreview.com/oauth2/v1/authorize/callback".

5. On the application registration, on the left menu, select **Authentication**.

6. Select **Add a platform**

7. Select **Web**.

8. Add the redirect URI you recorded in the IDP in Okta.

9. Select **Access tokens and ID tokens**.

10. In the admin console, select **Directory**.

11. Select **People**.

12. To edit the profile, select a test user.

13. In the profile, add **ToAzureAD**. See the following image.

14. Select **Save**.

Cost center	<input type="text"/>
costCenter	<input type="text"/>
Organization	<input type="text"/>
organization	<input type="text"/>
Division	<input type="text"/>
division	<input type="text"/> ToAzureAD
Department	<input type="text"/>
department	<input type="text"/>
ManagerId	<input type="text"/>
managerId	<input type="text"/>
Manager	<input type="text"/>
manager	<input type="text"/>

Save Cancel

15. Sign in to the [Microsoft 356 portal](#) as the modified user. If your user isn't in the managed authentication pilot, your action enters a loop. To exit the loop, add the user to the managed authentication experience.

Test Okta app access on pilot members

After you configure the Okta app in Microsoft Entra ID and configure the IDP in the Okta portal, assign the application to users.

1. In the Microsoft Entra admin center, browse to **Identity > Applications > Enterprise applications**.
2. Select the app registration you created.
3. Go to **Users and groups**.
4. Add the group that correlates with the managed authentication pilot.

! **Note**

You can add users and groups from the **Enterprise applications** page. You can't add users from the **App registrations** menu.

Home > MW Traders > Enterprise applications > Okta Application Access

Okta Application Access | Users and groups

Enterprise Application

Overview Deployment Plan Manage Properties Owners Roles and administrators (Preview) **Users and groups** Single sign-on Provisioning Security Permissions Token encryption Activity Sign-ins Usage & insights (Preview) Audit logs Provisioning logs (Preview) Access reviews

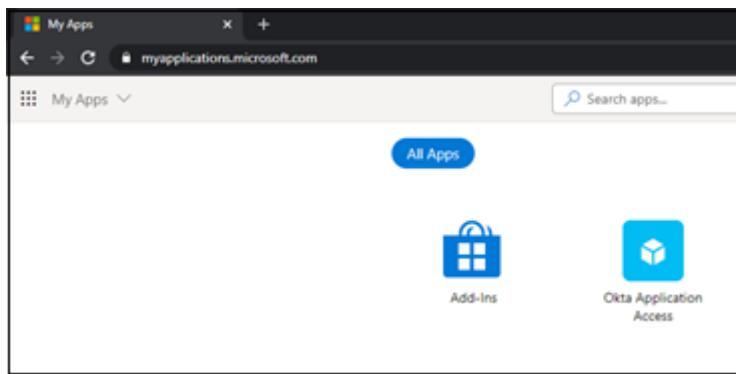
Add user/group Edit Remove Update Credentials Columns Got feedback?

The application will appear on the Access Panel for assigned users. Set 'visible to users?' to no in properties to prevent this.

First 100 shown, to search all users & groups, enter a display name.

Display Name	Object Type
<input type="checkbox"/> SA System Administrator	User
<input type="checkbox"/> MA Managed Authentication Staging Group	Group

5. Wait about 15 minutes.
6. Sign in as a managed authentication pilot user.
7. Go to [My Apps](#).



8. To return to the Okta home page, select the **Okta Application Access** tile.

Test managed authentication on pilot members

After you configure the Okta reverse-federation app, ask users to conduct testing on the managed authentication experience. We recommend you configure company branding to help users recognize the tenant.

Learn more: [Configure your company branding](#).

Important

Before you defederate the domains from Okta, identify needed Conditional Access policies. You can secure your environment before cut-off. See, [Tutorial: Migrate Okta sign-on policies to Microsoft Entra Conditional Access](#).

Defederate Office 365 domains

When your organization is comfortable with the managed authentication experience, you can defederate your domain from Okta. To begin, use the following commands to connect to Microsoft Graph PowerShell. If you don't have the Microsoft Graph PowerShell module, download it by entering `Install-Module Microsoft.Graph`.

1. In PowerShell, sign in to Microsoft Entra ID by using a Hybrid Identity Administrator account.

PowerShell

```
Connect-MgGraph -Scopes "Domain.ReadWrite.All",  
"Directory.AccessAsUser.All"
```

2. To convert the domain, run the following command:

PowerShell

```
Update-MgDomain -DomainId yourdomain.com -AuthenticationType "Managed"
```

3. Verify that the domain is converted to managed by running the following command. The Authentication type should be set to managed.

PowerShell

```
Get-MgDomain -DomainId yourdomain.com
```

After you set the domain to managed authentication, you defederate your Office 365 tenant from Okta while maintaining user access to the Okta home page.

Next steps

- [Tutorial: Migrate Okta sync provisioning to Microsoft Entra Connect-based synchronization](#)
- [Tutorial: Migrate Okta sign-on policies to Microsoft Entra Conditional Access](#)
- [Tutorial: Migrate your applications from Okta to Microsoft Entra ID](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Secure hybrid access: Protect legacy apps with Microsoft Entra ID

Article • 09/20/2024

In this article, learn to protect your on-premises and cloud legacy authentication applications by connecting them to Microsoft Entra ID.

- [Application Proxy](#):
 - Remote access to on-premises applications through Microsoft Entra application proxy
 - Protect users, apps, and data in the cloud and on-premises
 - [Use it to publish on-premises web applications externally](#)
- [Secure hybrid access through Microsoft Entra ID partner integrations](#):
 - Pre-built solutions
 - [Apply Conditional Access policies per application](#)

In addition to Application Proxy, you can strengthen your security posture with [Microsoft Entra Conditional Access](#) and [Microsoft Entra ID Protection](#).

Single sign-on and multifactor authentication

With Microsoft Entra ID as an identity provider (IdP), you can use modern authentication and authorization methods like [single sign-on \(SSO\)](#) and [Microsoft Entra multifactor authentication](#) to secure legacy, on-premises applications.

Secure hybrid access with Application Proxy

Use Application Proxy to protect users, apps, and data in the cloud, and on premises. Use this tool for secure remote access to on-premises web applications. Users don't need to use a virtual private network (VPN); they connect to applications from devices with SSO.

Learn more:

- [Remote access to on-premises applications through Microsoft Entra application proxy](#)
- [Tutorial: Add an on-premises application for remote access through Application Proxy in Microsoft Entra ID](#)
- [How to configure SSO to an Application Proxy application](#)

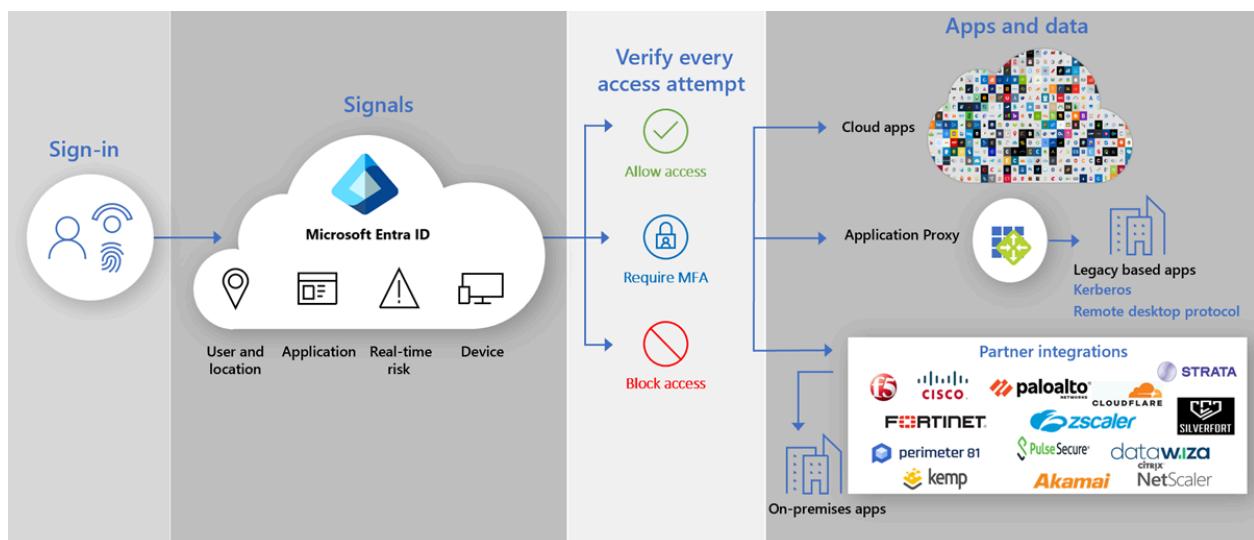
- Using Microsoft Entra application proxy to publish on-premises apps for remote users

Application publishing and access management

Use Application Proxy remote access as a service to publish applications to users outside the corporate network. Help improve your cloud access management without requiring modification to your on-premises applications. Plan a [Microsoft Entra application proxy deployment](#).

Partner integrations for apps: on-premises and legacy authentication

Microsoft partners with various companies that deliver pre-built solutions for on-premises applications, and applications that use legacy authentication. The following diagram illustrates a user flow from sign-in to secure access to apps and data.



Secure hybrid access through Microsoft Entra ID partner integrations

The following partners offer solutions to support [Conditional Access policies per application](#). Use the tables in the following sections to learn about the partners and Microsoft Entra integration documentation.

 Expand table

Partner	Integration documentation
Akamai Technologies	Tutorial: Microsoft Entra SSO integration with Akamai

Partner	Integration documentation
Citrix Systems, Inc.	Tutorial: Microsoft Entra SSO integration with Citrix ADC SAML Connector for Microsoft Entra ID (Kerberos-based authentication)
Cloudflare, Inc.	Tutorial: Configure Cloudflare with Microsoft Entra ID for secure hybrid access
Datawiza	Tutorial: Configure Secure Hybrid Access with Microsoft Entra ID and Datawiza
F5, Inc.	Integrate F5 BIG-IP with Microsoft Entra ID Tutorial: Configure F5 BIG-IP SSL-VPN for Microsoft Entra SSO
Progress Software Corporation, Progress Kemp	Tutorial: Microsoft Entra SSO integration with Kemp LoadMaster Microsoft Entra integration
Perimeter 81 Ltd.	Tutorial: Microsoft Entra SSO integration with Perimeter 81
Silverfort	Tutorial: Configure Secure Hybrid Access with Microsoft Entra ID and Silverfort
Strata Identity, Inc.	Integrate Microsoft Entra SSO with Mavericks Identity Orchestrator SAML Connector

Partners with pre-built solutions and integration documentation

[+] [Expand table](#)

Partner	Integration documentation
Amazon Web Service, Inc.	Tutorial: Microsoft Entra SSO integration with AWS ClientVPN
Check Point Software Technologies Ltd.	Tutorial: Microsoft Entra single SSO integration with Check Point Remote Secure Access VPN
Cisco Systems, Inc.	Tutorial: Microsoft Entra SSO integration with Cisco Secure Firewall - Secure Client
Fortinet, Inc.	Tutorial: Microsoft Entra SSO integration with FortiGate SSL VPN
Palo Alto Networks	Tutorial: Microsoft Entra SSO integration with Palo Alto Networks Admin UI
Pulse Secure	Tutorial: Microsoft Entra SSO integration with Pulse Connect Secure (PCS) Tutorial: Microsoft Entra SSO integration with Pulse Secure Virtual Traffic Manager
Zscaler, Inc.	Tutorial: Integrate Zscaler Private Access with Microsoft Entra ID

Next steps

Select a partner in the tables mentioned to learn how to integrate their solution with Microsoft Entra ID.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Secure hybrid access with Microsoft Entra integration

Article • 10/23/2023

Microsoft Entra ID supports modern authentication protocols that help keep applications secure. However, many business applications work in a protected corporate network, and some use legacy authentication methods. As companies build Zero Trust strategies and support hybrid and cloud environments, there are solutions that connect apps to Microsoft Entra ID and provide authentication for legacy applications.

Learn more: [Zero Trust security](#)

Microsoft Entra ID natively supports modern protocols:

- Security Assertion Markup Language (SAML)
- Web Service Federation (WS-Fed)
- OpenID Connect (OIDC)

Microsoft Entra application proxy, or Microsoft Entra application proxy supports Kerberos and header-based authentication. Other protocols, like Secure Shell (SSH), (Microsoft Windows NT LAN Manager) NTLM, Lightweight Directory Access Protocol (LDAP), and cookies, aren't supported. But, independent software vendors (ISVs) can create solutions to connect these applications with Microsoft Entra ID.

ISVs can help customers discover and migrate software as a service (SaaS) applications into Microsoft Entra ID. They can connect apps that use legacy authentication methods with Microsoft Entra ID. Customers can consolidate onto Microsoft Entra ID to simplify their app management and implement Zero Trust principles.

Solution overview

The solution that you build can include the following parts:

- **App discovery** - Often, customers aren't aware of every application in use
 - Application discovery finds applications, facilitating app integrating with Microsoft Entra ID
- **App migration** - Create a workflow to integrate apps with Microsoft Entra ID without using the Microsoft Entra admin center
 - Integrate apps that customers use today
- **Legacy authentication support** - Connect apps with legacy authentication methods and single sign-on (SSO)

- **Conditional Access** - Enable customers to apply Microsoft Entra policies to apps in your solution without using the Microsoft Entra admin center

Learn more: [What is Conditional Access?](#)

See the following sections for technical considerations and recommendations.

Publishing applications to Azure Marketplace

Azure Marketplace is a trusted source of applications for IT admins. Applications are compatible with Microsoft Entra ID and support SSO, automate user provisioning, and integrate into customer tenants with automated app registration.

You can pre-integrate your application with Microsoft Entra ID to support SSO and automated provisioning. See, [Submit a request to publish your application in Microsoft Entra application gallery](#).

We recommend you become a verified publisher, so customers know you're the trusted publisher. See, [Publisher verification](#).

Enable single sign-on for IT admins

There are several ways to enable SSO for IT administrators to your solution. See, [Plan a single sign-on deployment, SSO options](#).

Microsoft Graph uses OIDC/OAuth. Customers use OIDC to sign in to your solution. Use the JSON Web Token (JWT) Microsoft Entra ID issues to interact with Microsoft Graph. See, [OpenID Connect on the Microsoft identity platform](#).

If your solution uses SAML for IT administrator SSO, the SAML token won't enable your solution to interact with Microsoft Graph. You can use SAML for IT administrator SSO, but your solution needs to support OIDC integration with Microsoft Entra ID, so it can get a JWT from Microsoft Entra ID to interact with Microsoft Graph. See, [How the Microsoft identity platform uses the SAML protocol](#).

You can use one of the following SAML approaches:

- **Recommended SAML approach:** Create a new registration in Azure Marketplace, which is an OIDC app. Customers add the SAML and OIDC apps to their tenant. If your application isn't in the Microsoft Entra gallery, you can start with a non-gallery multi-tenant app.
 - [Configure an OpenID Connect OAuth application from Microsoft Entra app gallery](#)

- [Making your application multi-tenant](#)
- **Alternate SAML approach:** Customers can create an OIDC application registration in their Microsoft Entra tenant and set the URLs, endpoints, and permissions

Use the client credentials grant type, which requires the solution to allow customers to enter a client ID and secret. The solution also requires you store this information. Get a JWT from Microsoft Entra ID, and then use it to interact with Microsoft Graph. See, [Get a token](#). We recommend you reparse customer documentation about how to create application registration in their Microsoft Entra tenant. Include endpoints, URLs, and permissions.

 **Note**

Before applications are used for IT administrator or user SSO, the customer IT administrator must consent to the application in their tenant. See, [Grant tenant-wide admin consent to an application](#).

Authentication flows

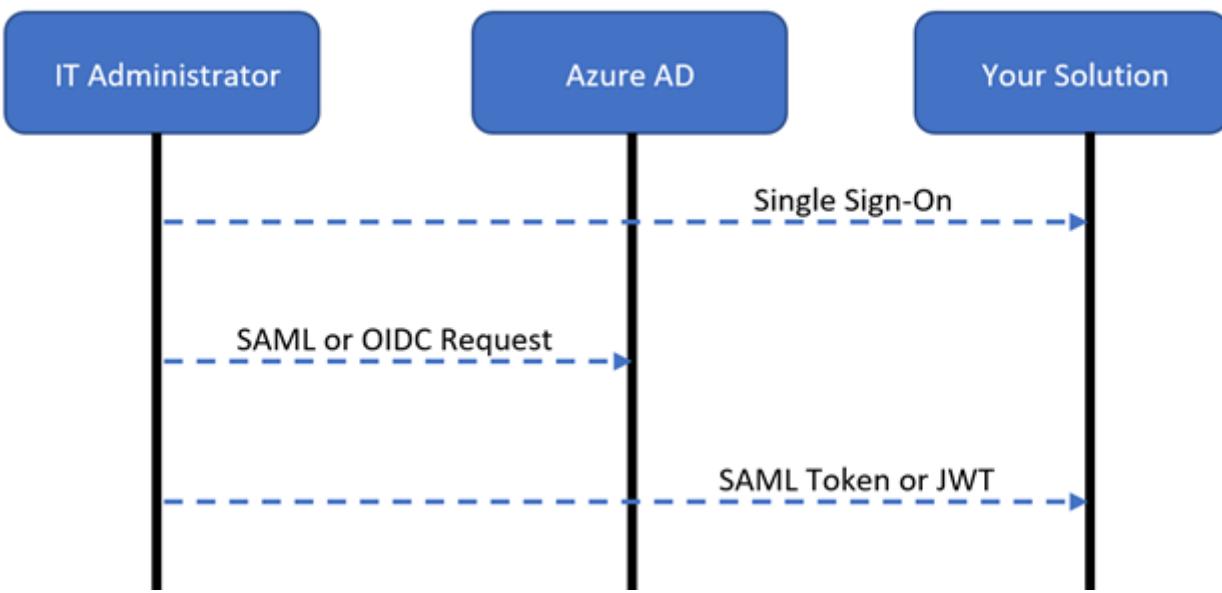
The solution authentication flows support the following scenarios:

- The customer IT administrator signs in with SSO to administer your solution
- The customer IT administrator uses your solution to integrate applications with Microsoft Entra ID with Microsoft Graph
- Users sign in to legacy applications secured by your solution and Microsoft Entra ID

Your customer IT administrator does single sign-on to your solution

Your solution can use SAML or OIDC for SSO, when the customer IT administrator signs in. We recommend the IT administrator signs in to your solution with their Microsoft Entra credentials, which enables use of current security controls. Integrate your solution with Microsoft Entra ID for SSO through SAML or OIDC.

The following diagram illustrates the user authentication flow:

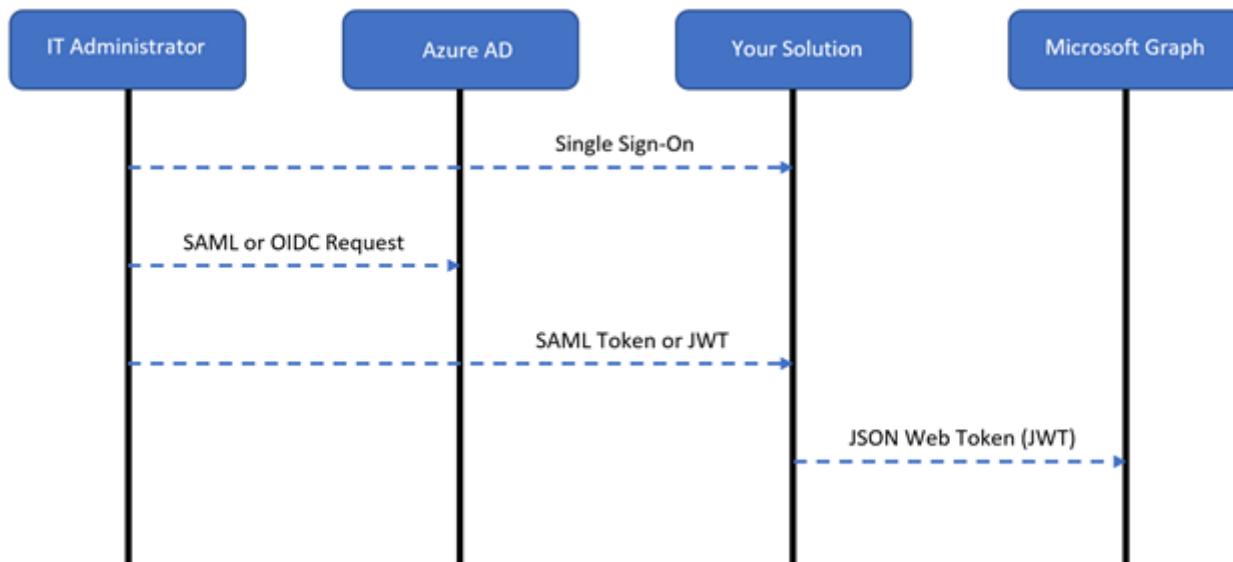


1. The IT administrator signs in to your solution with their Microsoft Entra credentials
2. The solution redirects the IT administrator to Microsoft Entra ID with a SAML or an OIDC sign-in request
3. Microsoft Entra authenticates the IT administrator and redirects them to your solution, with a SAML token or JWT to be authorized in your solution

IT administrators integrate applications with Microsoft Entra ID

IT administrators integrate applications with Microsoft Entra ID by using your solution, which employs Microsoft Graph to create application registrations and Microsoft Entra Conditional Access policies.

The following diagram illustrates the user authentication flow:



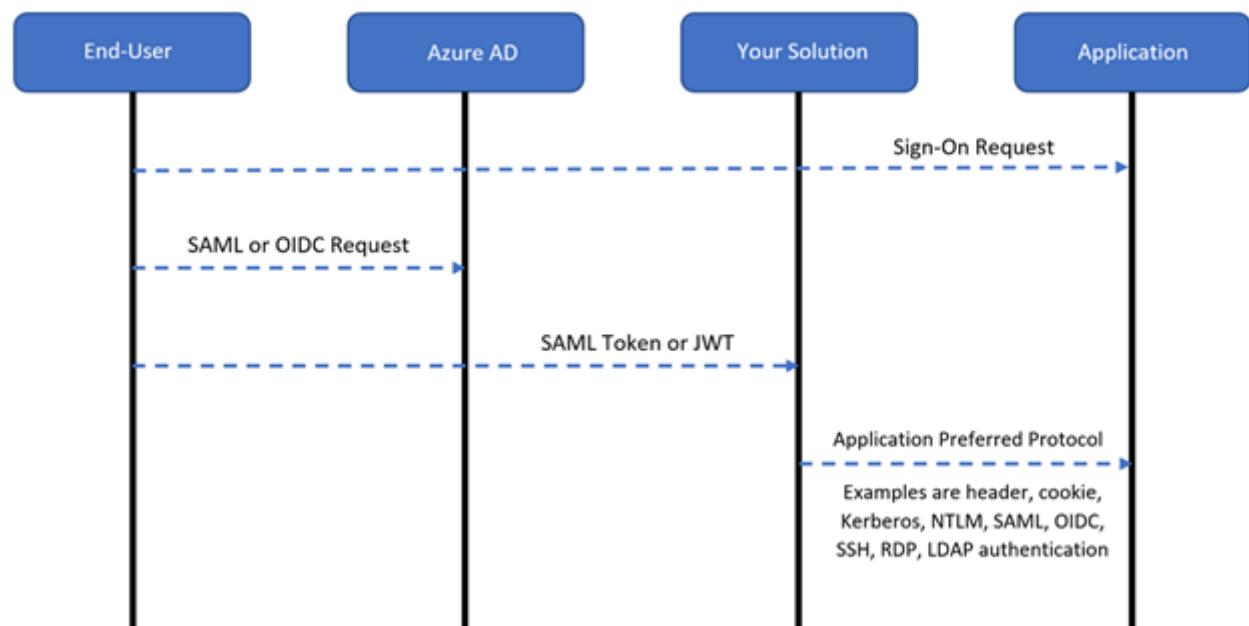
1. The IT administrator signs in to your solution with their Microsoft Entra credentials

2. The solution redirects the IT administrator to Microsoft Entra ID with a SAML or an OIDC sign-in request
3. Microsoft Entra authenticates the IT administrator and redirects them to your solution with a SAML token or JWT for authorization
4. When the IT administrator integrates an application with Microsoft Entra ID, the solution calls Microsoft Graph with their JWT to register applications, or apply Microsoft Entra Conditional Access policies

Users sign in to the applications

When users sign in to applications, they use OIDC or SAML. If the applications need to interact with Microsoft Graph or Microsoft Entra protected API, we recommend you configure them to use OICD. This configuration ensures the JWT is applied to interact with Microsoft Graph. If there's no need for applications to interact with Microsoft Graph, or Microsoft Entra protected APIs, then use SAML.

The following diagram shows user authentication flow:



1. The user signs in to an application
2. The solution redirects the user to Microsoft Entra ID with a SAML or an OIDC sign-in request
3. Microsoft Entra authenticates the user and redirects them to your solution with a SAML token or JWT for authorization
4. The solution allows the request by using the application protocol

Microsoft Graph API

We recommend use of the following APIs. Use Microsoft Entra ID to configure delegated permissions or application permissions. For this solution, use delegated permissions.

- **Applications templates API** - In Azure Marketplace, use this API to find a matching application template
 - Permissions required: Application.Read.All
- **Application registration API** - Create OIDC or SAML application registrations for users to sign in to applications secured with your solution
 - Permissions required: Application.Read.All, Application.ReadWrite.All
- **Service principal API** - After you register the app, update the service principal object to set SSO properties
 - Permissions required: Application.ReadWrite.All, Directory.AccessAsUser.All, AppRoleAssignment.ReadWrite.All (for assignment)
- **Conditional Access API** - Apply Microsoft Entra Conditional Access policies to user applications
 - Permissions required: Policy.Read.All, Policy.ReadWrite.ConditionalAccess, and Application.Read.All

Learn more [Use the Microsoft Graph API](#)

Microsoft Graph API scenarios

Use the following information to implement application registrations, connect legacy applications, and enable Conditional Access policies. Learn to automate admin consent, get the token-signing certificate, and assign users and groups.

Use Microsoft Graph API to register apps with Microsoft Entra ID

Add apps in Azure Marketplace

Some applications your customers use are in the [Azure Marketplace](#). You can create a solution that adds applications to the customer tenant. Use the following example with Microsoft Graph API to search Azure Marketplace for a template.

Note

In Application Templates API, the display name is case-sensitive.

Authorization: Required with a valid Bearer token

Method: Get

```
https://graph.microsoft.com/v1.0/applicationTemplates?$filter=displayname eq "Salesforce.com"
```

If you find a match from the API call, capture the ID. Make the following API call and provide a display name for the application in the JSON body:

https

Authorization: Required with a valid Bearer token

Method: POST

Content-type: application/json

```
https://graph.microsoft.com/v1.0/applicationTemplates/cd3ed3de-93ee-400b-8b19-b61ef44a0f29/instantiate
{
    "displayname": "Salesforce.com"
}
```

After you make the API call, you generate a service principal object. Capture the application ID and the service principal ID to use in the next API calls.

Patch the service principal object with the SAML protocol and a login URL:

https

Authorization: Required with a valid Bearer token

Method: PATCH

Content-type: servicePrincipal/json

```
https://graph.microsoft.com/v1.0/servicePrincipals/3161ab85-8f57-4ae0-82d3-7a1f71680b27
{
    "preferredSingleSignOnMode": "saml",
    "loginURL": "https://www.salesforce.com"
}
```

Patch the application object with redirect URLs and the identifier URLs:

https

Authorization: Required with a valid Bearer token

Method: PATCH

Content-type: application/json

```
https://graph.microsoft.com/v1.0/applications/54c4806b-b260-4a12-873c-
```

```
967116983792
```

```
{
  "web": {
    "redirectUris": ["https://www.salesforce.com"],
    "identifierUris": ["https://www.salesforce.com"]
  }
}
```

Add apps not in Azure Marketplace

If there's no match in Azure Marketplace, or to integrate a custom application, register a custom application in Microsoft Entra ID with the template ID: 8adf8e6e-67b2-4cf2-a259-e3dc5476c621. Then, make the following API call and provide an application display name in the JSON body:

```
https
```

```
Authorization: Required with a valid Bearer token
```

```
Method: POST
```

```
Content-type: application/json
```

```
https://graph.microsoft.com/v1.0/applicationTemplates/8adf8e6e-67b2-4cf2-a259-e3dc5476c621/instantiate
{
  "displayname": "Custom SAML App"
}
```

After you make the API call, you generate a service principal object. Capture the application ID and the service principal ID to use in the next API calls.

Patch the service principal object with the SAML protocol and a login URL:

```
https
```

```
Authorization: Required with a valid Bearer token
```

```
Method: PATCH
```

```
Content-type: servicePrincipal/json
```

```
https://graph.microsoft.com/v1.0/servicePrincipals/3161ab85-8f57-4ae0-82d3-7a1f71680b27
{
  "preferredSingleSignOnMode": "saml",
  "loginURL": "https://www.samlapp.com"
}
```

Patch the application object with redirect URIs and identifier URIs:

```
https
```

Authorization: Required with a valid Bearer token

Method: PATCH

Content-type: application/json

```
https://graph.microsoft.com/v1.0/applications/54c4806b-b260-4a12-873c-  
967116983792  
{  
    "web": {  
        "redirectUris": ["https://www.samlapp.com"],  
        "identifierUris": ["https://www.samlapp.com"]  
    }  
}
```

Use Microsoft Entra single sign-on

After the SaaS applications are registered in Microsoft Entra ID, the applications need to start using Microsoft Entra ID as the identity provider (IdP):

- **Applications support one-click SSO** - Microsoft Entra ID enables the applications. In the Microsoft Entra admin center, the customer performs one-click SSO with the administrative credentials for the supported SaaS applications.
 - Learn more: [One-click app configuration of single sign-on](#)
- **Applications don't support one-click SSO** - The customer enables the applications to use Microsoft Entra ID.
 - [Tutorials for integrating SaaS applications with Microsoft Entra ID](#)

Connect apps to Microsoft Entra ID with legacy authentication

Your solution can enable the customer to use SSO and Microsoft Entra features, even unsupported applications. To allow access with legacy protocols, your application calls Microsoft Entra ID to authenticate the user and apply [Microsoft Entra Conditional Access policies](#). Enable this integration from your console. Create a SAML or an OIDC application registration between your solution and Microsoft Entra ID.

Create a SAML application registration

Use the following custom application template ID: 8adf8e6e-67b2-4cf2-a259-e3dc5476c621. Then, make the following API call and provide a display name in the JSON body:

```
https
```

Authorization: Required with a valid Bearer token

Method: POST

Content-type: application/json

```
https://graph.microsoft.com/v1.0/applicationTemplates/8adf8e6e-67b2-4cf2-a259-e3dc5476c621/instantiate
```

```
{  
    "displayname": "Custom SAML App"  
}
```

After you make the API call, you generate a service principal object. Capture the application ID and the service principal ID to use in the next API calls.

Patch the service principal object with the SAML protocol and a login URL:

https

Authorization: Required with a valid Bearer token

Method: PATCH

Content-type: servicePrincipal/json

```
https://graph.microsoft.com/v1.0/servicePrincipals/3161ab85-8f57-4ae0-82d3-7a1f71680b27
```

```
{  
    "preferredSingleSignOnMode": "saml",  
    "loginURL": "https://www.samlapp.com"  
}
```

Patch the application object with redirect URIs and identifier URIs:

https

Authorization: Required with a valid Bearer token

Method: PATCH

Content-type: application/json

```
https://graph.microsoft.com/v1.0/applications/54c4806b-b260-4a12-873c-967116983792
```

```
{  
    "web": {  
        "redirectUris": ["https://www.samlapp.com"],  
        "identifierUris": ["https://www.samlapp.com"]  
    }  
}
```

Create an OIDC application registration

Use the following template ID for a custom application: 8adf8e6e-67b2-4cf2-a259-e3dc5476c621. Make the following API call and provide a display name in the JSON body:

```
https
```

Authorization: Required with a valid Bearer token

Method: POST

Content-type: application/json

```
https://graph.microsoft.com/v1.0/applicationTemplates/8adf8e6e-67b2-4cf2-a259-e3dc5476c621/instantiate
{
    "displayname": "Custom OIDC App"
}
```

From the API call, capture the application ID and the service principal ID to use in the next API calls.

```
https
```

Authorization: Required with a valid Bearer token

Method: PATCH

Content-type: application/json

```
https://graph.microsoft.com/v1.0/applications/{Application Object ID}
{
    "web": {
        "redirectUris": ["https://www.samlapp.com"],
        "identifierUris": ["https://www.samlapp.com"],
        "requiredResourceAccess": [
            {
                "resourceAppId": "00000003-0000-0000-c000-000000000000",
                "resourceAccess": [
                    {
                        "id": "7427e0e9-2fba-42fe-b0c0-848c9e6a8182",
                        "type": "Scope"
                    },
                    {
                        "id": "e1fe6dd8-ba31-4d61-89e7-88639da4683d",
                        "type": "Scope"
                    },
                    {
                        "id": "37f7f235-527c-4136-accd-4a02d197296e",
                        "type": "Scope"
                    }
                ]
            }
        }
}
```

ⓘ Note

The API permissions in the `resourceAccess` node grant the application the `openid`, `User.Read`, and `offline_access` permissions, which enable sign-in. See, [Overview of Microsoft Graph permissions](#).

Apply Conditional Access policies

Customers and partners can use the Microsoft Graph API to create or apply per application [Conditional Access policies](#). For partners, customers can apply these policies from your solution without using the Microsoft Entra admin center. There are two options to apply Microsoft Entra Conditional Access policies:

- Assign the application to a Conditional Access policy
- Create a new Conditional Access policy and assign the application to it

Use a Conditional Access policy

For a list of Conditional Access policies, run the following query. Get the policy object ID to modify.

```
https

Authorization: Required with a valid Bearer token
Method:GET

https://graph.microsoft.com/v1.0/identity/conditionalAccess/policies
```

To patch the policy, include the application object ID to be in scope of `includeApplications`, in the JSON body:

```
https

Authorization: Required with a valid Bearer token
Method: PATCH

https://graph.microsoft.com/v1.0/identity/conditionalAccess/policies/{policy
id}
{
    "displayName": "Existing Conditional Access Policy",
    "state": "enabled",
    "conditions": {
        "applications":
```

```
{  
    "includeApplications": [  
        "00000003-0000-0ff1-ce00-000000000000",  
        "{Application Object ID}"  
    ]  
,  
    "users": {  
        "includeUsers": [  
            "All"  
        ]  
    }  
,  
    "grantControls":  
    {  
        "operator": "OR",  
        "builtInControls": [  
            "mfa"  
        ]  
    }  
}
```

Create a new Conditional Access policy

Add the application object ID to be in scope of `includeApplications`, in the JSON body:

https

Authorization: Required with a valid Bearer token

Method: POST

```
https://graph.microsoft.com/v1.0/identity/conditionalAccess/policies/  
{  
    "displayName": "New Conditional Access Policy",  
    "state": "enabled",  
    "conditions":  
    {  
        "applications": {  
            "includeApplications": [  
                "{Application Object ID}"  
            ]  
        },  
        "users": {  
            "includeUsers": [  
                "All"  
            ]  
        }  
    },  
    "grantControls": {  
        "operator": "OR",  
        "builtInControls": [  
            "mfa"  
        ]  
    }  
}
```

```
    }  
}
```

To create new Microsoft Entra Conditional Access policies, see [Conditional Access: Programmatic access](#).

```
https
```

```
#Policy Template for Requiring Compliant Device
```

```
{  
    "displayName": "Enforce Compliant Device",  
    "state": "enabled",  
    "conditions": {  
        "applications": {  
            "includeApplications": [  
                "{Application Object ID}"  
            ]  
        },  
        "users": {  
            "includeUsers": [  
                "All"  
            ]  
        }  
    },  
    "grantControls": {  
        "operator": "OR",  
        "builtInControls": [  
            "compliantDevice",  
            "domainJoinedDevice"  
        ]  
    }  
}
```

```
#Policy Template for Block
```

```
{  
    "displayName": "Block",  
    "state": "enabled",  
    "conditions": {  
        "applications": {  
            "includeApplications": [  
                "{Application Object ID}"  
            ]  
        },  
        "users": {  
            "includeUsers": [  
                "All"  
            ]  
        }  
    },  
    "grantControls": {  
        "operator": "OR",  
        "builtInControls": [  
            "blockDevice",  
            "blockUser"  
        ]  
    }  
}
```

```
    "builtInControls": [
        "block"
    ]
}
```

Automate admin consent

If the customer is adding applications from your solution to Microsoft Entra ID, you can automate administrator consent with Microsoft Graph. You need the application service principal object ID you created in API calls, and the Microsoft Graph service principal object ID from the customer tenant.

Get the Microsoft Graph service principal object ID by making the following API call:

```
https
```

Authorization: Required with a valid Bearer token

Method: GET

```
https://graph.microsoft.com/v1.0/serviceprincipals/?$filter=appId eq  
'00000003-0000-0000-c000-000000000000'&$select=id,displayName
```

To automate admin consent, make the following API call:

```
https
```

Authorization: Required with a valid Bearer token

Method: POST

Content-type: application/json

```
https://graph.microsoft.com/v1.0/oauth2PermissionGrants  
{  
    "clientId":"{Service Principal Object ID of Application}",  
    "consentType":"AllPrincipals",  
    "principalId":null,  
    "resourceId":"{Service Principal Object ID Of Microsoft Graph}",  
    "scope":"openid user.read offline_access"}  
}
```

Get the token-signing certificate

To get the public portion of the token-signing certificate, use **GET** from the Microsoft Entra metadata endpoint for the application:

```
https
```

Method:GET

```
https://login.microsoftonline.com/{Tenant_ID}/federationmetadata/2007-06/federationmetadata.xml?appid={Application_ID}
```

Assign users and groups

After you publish the application to Microsoft Entra ID, you can assign the app to users and groups to ensure it appears on the My Apps portal. This assignment is on the service principal object generated when you created the application. See, [My Apps portal overview](#).

Get `AppRole` instances the application might have associated with it. It's common for SaaS applications to have various `AppRole` instances associated with them. Typically, for custom applications, there's one default `AppRole` instance. Get the `AppRole` instance ID you want to assign:

```
https
```

`Authorization`: Required with a valid Bearer token

Method:GET

```
https://graph.microsoft.com/v1.0/servicePrincipals/3161ab85-8f57-4ae0-82d3-7a1f71680b27
```

From Microsoft Entra ID, get the user or group object ID that you want to assign to the application. Take the app role ID from the previous API call and submit it with the patch body on the service principal:

```
https
```

`Authorization`: Required with a valid Bearer token

Method: PATCH

`Content-type`: servicePrincipal/json

```
https://graph.microsoft.com/v1.0/servicePrincipals/3161ab85-8f57-4ae0-82d3-7a1f71680b27
{
    "principalId": "{Principal Object ID of User -or- Group}",
    "resourceId": "{Service Principal Object ID}",
    "appRoleId": "{App Role ID}"
}
```

Partnerships

To help protect legacy applications, while using networking and delivery controllers, Microsoft has partnerships with the following application delivery controller (ADC) providers.

- **Akamai Enterprise Application Access**
 - [Tutorial: Microsoft Entra SSO integration with Akamai](#)
- **Citrix ADC**
 - [Tutorial: Microsoft Entra SSO integration with Citrix ADC SAML Connector for Microsoft Entra ID \(Kerberos-based authentication\)](#)
- **F5 BIG-IP Access Policy Manager**
 - [Tutorial: Microsoft Entra SSO integration with Citrix ADC SAML Connector for Microsoft Entra ID \(Kerberos-based authentication\)](#)
- **Kemp LoadMaster**
 - [Tutorial: Microsoft Entra SSO integration with Kemp LoadMaster Microsoft Entra integration](#)
- **Pulse Secure Virtual Traffic Manager**
 - [Tutorial: Microsoft Entra SSO integration with Pulse Secure Virtual Traffic Manager](#)

The following VPN solution providers connect with Microsoft Entra ID to enable modern authentication and authorization methods like SSO and multifactor authentication (MFA).

- **Cisco AnyConnect**
 - [Tutorial: Microsoft Entra SSO integration with Cisco AnyConnect](#)
- **Fortinet FortiGate**
 - [Tutorial: Microsoft Entra SSO integration with FortiGate SSL VPN](#)
- **F5 BIG-IP Access Policy Manager**
 - [Tutorial: Configure F5 BIG-IP SSL-VPN for Microsoft Entra SSO](#)
- **Palo Alto Networks GlobalProtect**
 - [Tutorial: Microsoft Entra SSO integration with Palo Alto Networks - Admin UI](#)
- **Pulse Connect Secure**
 - [Tutorial: Microsoft Entra SSO integration with Pulse Secure PCS](#)

The following software-defined perimeter (SDP) solutions providers connect with Microsoft Entra ID for authentication and authorization methods like SSO and MFA.

- **Datawiza Access Broker**
 - [Tutorial: Configure Secure Hybrid Access with Microsoft Entra ID and Datawiza](#)
- **Perimeter 81**

- [Tutorial: Microsoft Entra SSO integration with Perimeter 81](#)
- **Silverfort Authentication Platform**
 - [Tutorial: Configure Secure Hybrid Access with Microsoft Entra ID and Silverfort](#)
- **Strata Mavericks Identity Orchestrator**
 - [Integrate Microsoft Entra SSO with Mavericks Identity Orchestrator SAML Connector](#)
- **Zscaler Private Access**
 - [Tutorial: Integrate Zscaler Private Access with Microsoft Entra ID](#)

Tutorial: Configure Cloudflare with Microsoft Entra ID for secure hybrid access

Article • 01/29/2025

In this tutorial, learn to integrate Microsoft Entra ID with Cloudflare Zero Trust. Build rules based on user identity and group membership. Users authenticate with Microsoft Entra credentials and connect to Zero Trust protected applications.

Prerequisites

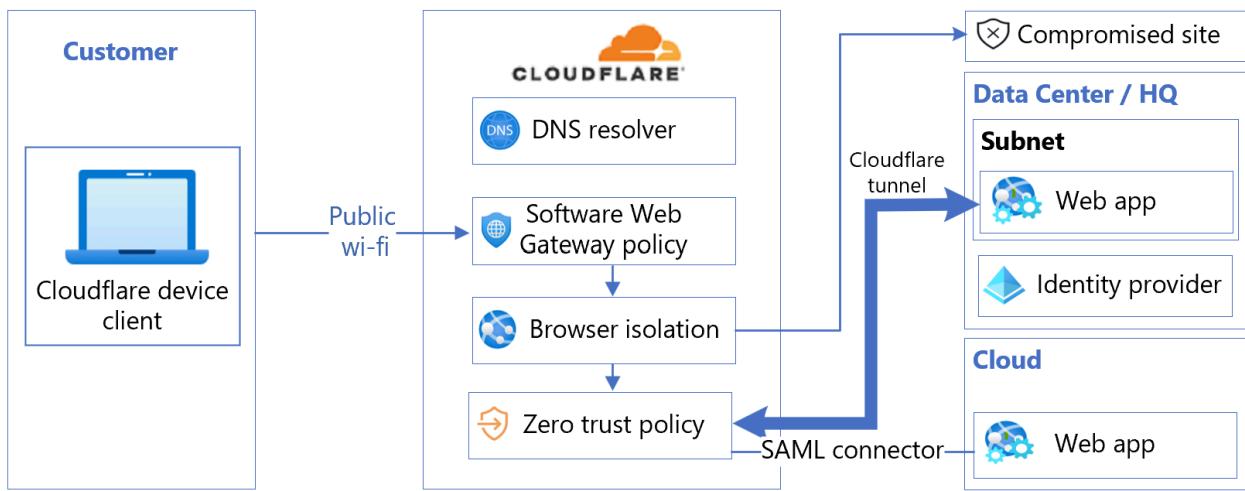
- A Microsoft Entra subscription
 - If you don't have one, get an [Azure free account](#)
- A Microsoft Entra tenant linked to the Microsoft Entra subscription
 - See, [Quickstart: Create a new tenant in Microsoft Entra ID](#)
- A Cloudflare Zero Trust account
 - If you don't have one, go to [Get started with Cloudflare's Zero Trust platform](#)
- One of the following roles: Cloud Application Administrator, or Application Administrator.

Integrate organization identity providers with Cloudflare Access

Cloudflare Zero Trust Access helps enforce default-deny, Zero Trust rules that limit access to corporate applications, private IP spaces, and hostnames. This feature connects users faster and safer than a virtual private network (VPN). Organizations can use multiple identity providers (IdPs), reducing friction when working with partners or contractors.

To add an IdP as a sign-in method, sign in to Cloudflare on the [Cloudflare sign in page](#) and Microsoft Entra ID.

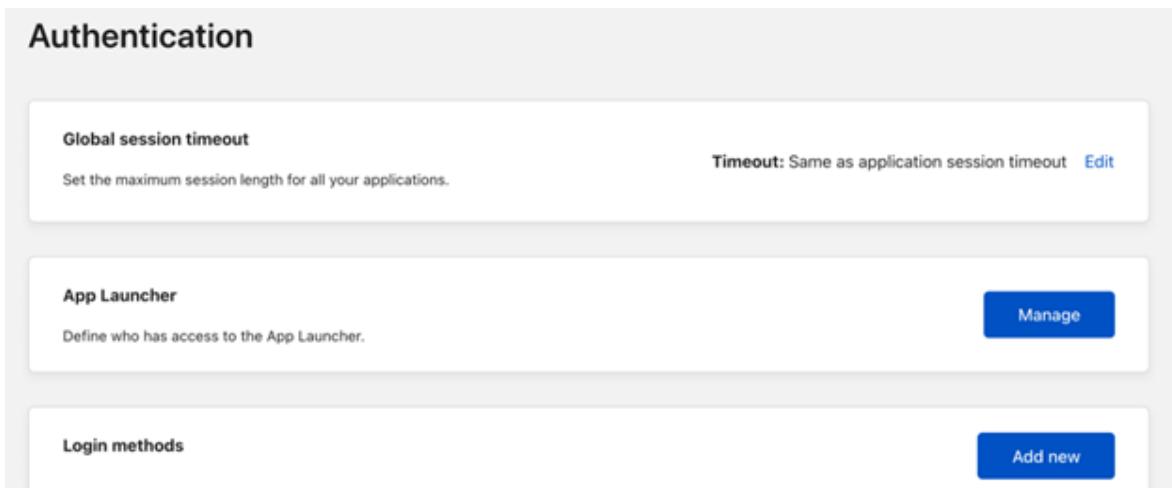
The following architecture diagram shows the integration.



Integrate a Cloudflare Zero Trust account with Microsoft Entra ID

Integrate Cloudflare Zero Trust account with an instance of Microsoft Entra ID.

1. Sign in to the Cloudflare Zero Trust dashboard on the [Cloudflare sign in page](#).
2. Navigate to **Settings**.
3. Select **Authentication**.
4. For **Login methods**, select **Add new**.



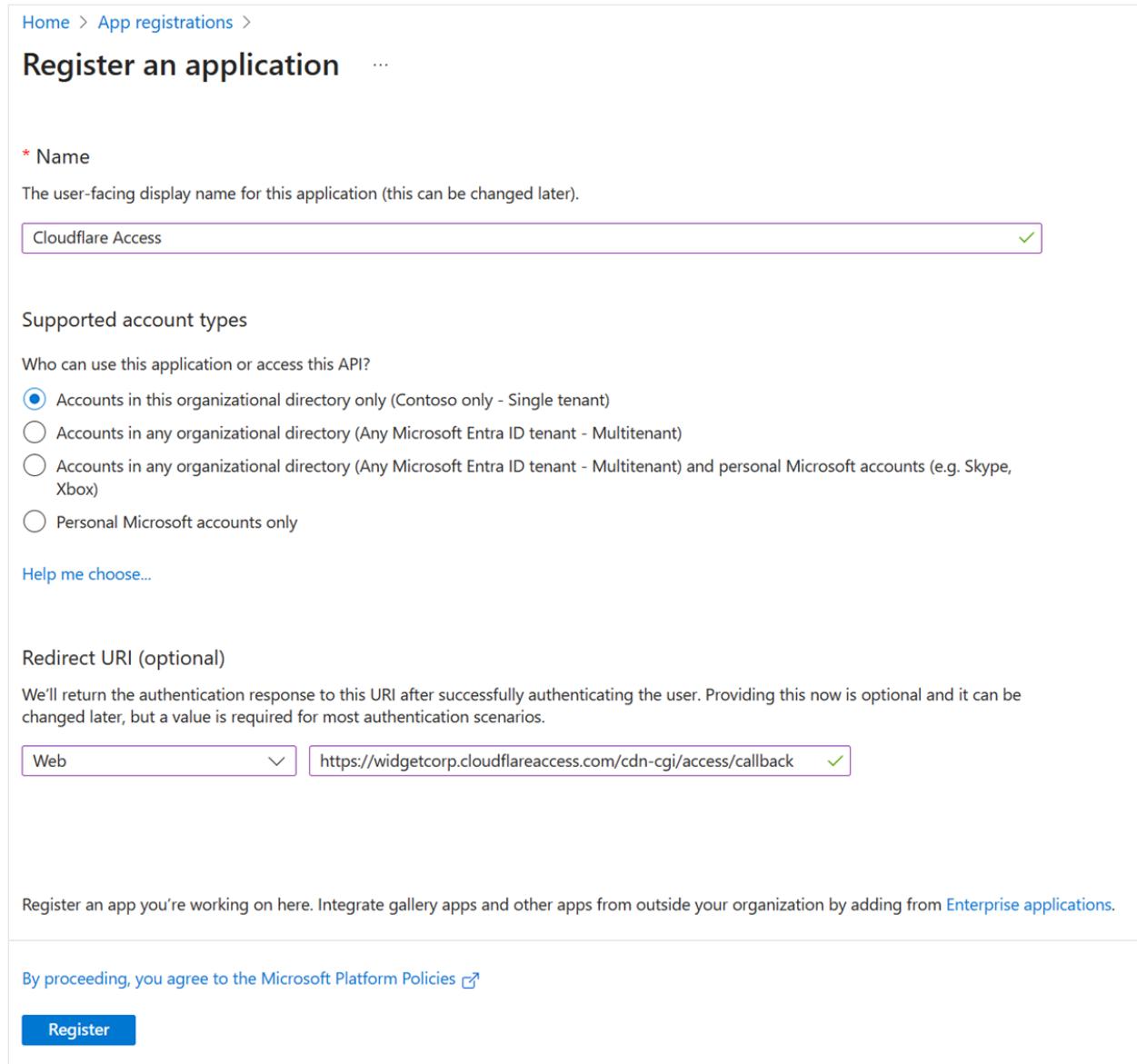
5. Under **Select an identity provider**, select **Microsoft Entra ID**.
6. The **Add Azure ID** dialog appears.
7. Enter Microsoft Entra instance credentials and make needed selections.
8. Select **Save**.

Register Cloudflare with Microsoft Entra ID

Use the instructions in the following three sections to register Cloudflare with Microsoft Entra ID.

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > App registrations**.
3. Select **New registration**.
4. Enter an application **Name**.
5. Enter a team name with **callback** at the end of the path. For example,
`https://<your-team-name>.cloudflareaccess.com/cdn-cgi/access/callback`
6. Select **Register**.

See the [team domain](#) definition in the Cloudflare Glossary.



The screenshot shows the 'Register an application' page in the Microsoft Entra Admin Center. It includes fields for Name, Supported account types, Redirect URI, and a note about Enterprise applications, along with a 'Register' button.

Home > App registrations >

Register an application

* Name
The user-facing display name for this application (this can be changed later).
Cloudflare Access

Supported account types
Who can use this application or access this API?
 Accounts in this organizational directory only (Contoso only - Single tenant)
 Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)
 Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
 Personal Microsoft accounts only
[Help me choose...](#)

Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.
Web https://widgetcorp.cloudflareaccess.com/cdn-cgi/access/callback

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

By proceeding, you agree to the [Microsoft Platform Policies](#)

Register

Certificates & secrets

1. On the Cloudflare Access screen, under **Essentials**, copy and save the Application (Client) ID and the Directory (Tenant) ID.

The screenshot shows the Microsoft Entra admin center interface. The left sidebar has a tree view with categories like Home, Favorites, Identity, Applications, Protection, and Identity governance. Under Applications, 'Enterprise applications' is expanded, and 'App registrations' is selected. In the main content area, the 'Cloudflare Access' application is displayed. The 'Overview' tab is selected. The 'Essentials' section contains the following information:

- Display name: Cloudflare Access
- Application (client) ID: 1111111-1111-1111-1111-111111111111
- Object ID: 1111111-1111-1111-1111-111111111111
- Directory (tenant) ID: 1111111-1111-1111-1111-111111111111
- Supported account types: My organization only
- Client credentials: Add a certificate or secret
- Redirect URIs: 1.web.0.spa.0/public-client
- Application ID URI: Add an Application ID URI
- Managed application in local directory: Cloudflare Access

Below the essentials section, there are 'Get Started' and 'Documentation' links, and a 'Build your application with the Microsoft identity platform' section.

2. In the left menu, under **Manage**, select **Certificates & secrets**.

The screenshot shows the Microsoft Entra admin center interface, similar to the previous one but with a different focus. The left sidebar is identical. In the main content area, the 'Cloudflare Access' application is selected, and the 'Certificates & secrets' section is highlighted. A modal window titled 'Add a client secret' is open over the page. The modal has fields for 'Description' (set to 'Cloudflare Access') and 'Expires'. The 'Expires' dropdown shows several options: Custom, Recommended: 180 days (6 months), 90 days (3 months), 365 days (12 months), 545 days (18 months), 730 days (24 months), and another Custom option.

3. Under **Certificates & secrets**, select **+ New client secret**.

4. In **Description**, enter the Client Secret.

5. Under **Expires**, select an expiration.

6. Select **Add**.

7. Under **Certificates & secrets**, from the **Value** field, copy the value. Consider the value an application password. The example value appears, Azure values appear in the Cloudflare Access configuration.

Permissions

1. In the left menu, select **API permissions**.

2. Select **+ Add a permission**.

3. Under **Select an API**, select **Microsoft Graph**.

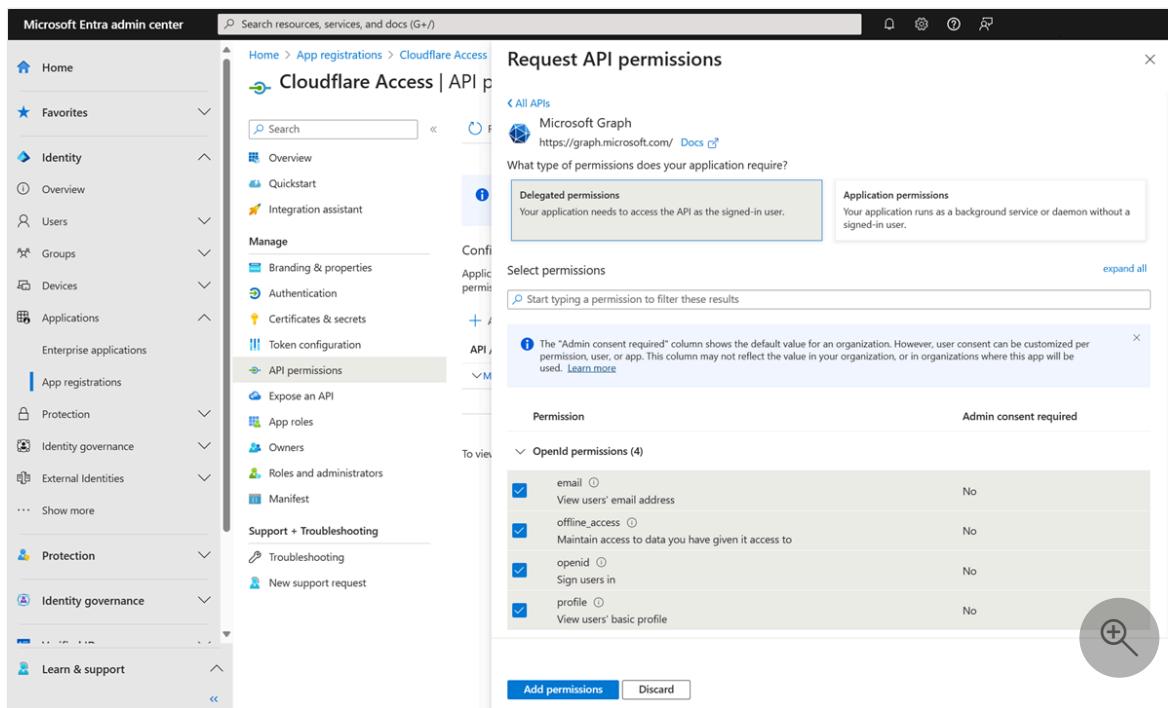
The screenshot shows the Microsoft Entra admin center interface. On the left, the navigation menu is visible with various sections like Home, Favorites, Identity, Overview, Users, Groups, Devices, Applications, Protection, Identity governance, External Identities, and Learn & support. The 'Applications' section is expanded, showing 'Enterprise applications' and 'App registrations'. The 'App registrations' section is also expanded, showing 'Cloudflare Access | API p...' which is currently selected. On the right, a modal window titled 'Request API permissions' is open. It has a search bar at the top and tabs for 'Microsoft APIs', 'APIs my organization uses', and 'My APIs'. The 'Microsoft APIs' tab is selected. Below it, a section titled 'Commonly used Microsoft APIs' lists several services with their icons and brief descriptions. The 'Microsoft Graph' service is highlighted with a blue border, indicating it is the selected API.

Service	Description
Azure Rights Management Services	Allow validated users to read and write protected content
Azure Service Management	Programmatic access to much of the functionality available through the Azure portal
Dynamics CRM	Access the capabilities of CRM business software and ERP systems
Intune	Programmatic access to Intune data
Office 365 Management APIs	Retrieve information about user, admin, system, and policy actions and events from Office 365 and Microsoft Entra ID activity logs
OneNote	Create and manage notes, lists, pictures, files, and more in OneNote notebooks
Power Automate	Embed flow templates and manage flows
Power BI Service	Programmatic access to Dashboard resources such as Datasets, Tables, and Rows in Power BI
SharePoint	Interact remotely with SharePoint data

4. Select **Delegated permissions** for the following permissions:

- Email
- openid
- profile
- offline_access
- user.read
- directory.read.all
- group.read.all

5. Under **Manage**, select **+ Add permissions**.



6. Select Grant Admin Consent for

API / Permissions name	Type	Description	Admin consent req...	Status
Directory.Read.All	Delegated	Read directory data	Yes	Not granted for Town L...
email	Delegated	View users' email address	-	Not granted for Town L...
Group.Read.All	Delegated	Read all groups	Yes	Not granted for Town L...
offline.access	Delegated	Maintain access to data you have given it access to	-	Not granted for Town L...
openid	Delegated	Sign users in	-	Not granted for Town L...
profile	Delegated	View users' basic profile	-	Not granted for Town L...
User.Read	Delegated	Sign in and read user profile	-	Not granted for Town L...
User.Read.All	Delegated	Read all users' full profiles	Yes	Not granted for Town L...

7. On the Cloudflare Zero Trust dashboard, navigate to Settings > Authentication.

8. Under Login methods, select Add new.

9. Select Microsoft Entra ID.

10. Enter values for Application ID, Application Secret, and Directory ID.

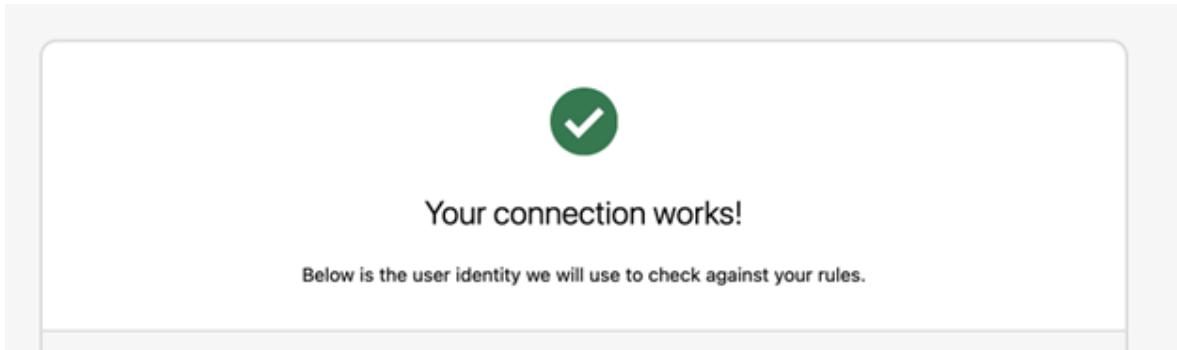
11. Select Save.

⚠ Note

For Microsoft Entra groups, in Edit your Microsoft Entra identity provider, for Support Groups select On.

Test the integration

1. On the Cloudflare Zero Trust dashboard, navigate to **Settings > Authentication**.
2. Under **Login methods**, for Microsoft Entra ID select **Test**.
3. Enter Microsoft Entra credentials.
4. The **Your connection works** message appears.



Next steps

- Go to developer.cloudflare.com for [Integrate SSO ↗](#)
- [Tutorial: Configure Conditional Access policies for Cloudflare Access](#)
- [Tutorial: Configure Cloudflare Web Application Firewall with Azure AD B2C](#)

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Tutorial: Configure Conditional Access policies in Cloudflare Access

Article • 01/29/2025

With Conditional Access, administrators enforce policies on application and user policies in Microsoft Entra ID. Conditional Access brings together identity-driven signals, to make decisions, and enforce organizational policies. Cloudflare Access creates access to self-hosted, software as a service (SaaS), or nonweb applications.

Learn more: [What is Conditional Access?](#)

Prerequisites

- A Microsoft Entra subscription
 - If you don't have one, get an [Azure free account](#)
- A Microsoft Entra tenant linked to the Microsoft Entra subscription
 - See, [Quickstart: Create a new tenant in Microsoft Entra ID](#)
- One of the following roles: Cloud Application Administrator, or Application Administrator.
- Configured users in the Microsoft Entra subscription
- A Cloudflare account
 - Go to dash.cloudflare.com to [Get started with Cloudflare](#)

Scenario architecture

- **Microsoft Entra ID** - Identity Provider (IdP) that verifies user credentials and Conditional Access
- **Application** - You created for IdP integration
- **Cloudflare Access** - Provides access to applications

Set up an identity provider

Go to developers.cloudflare.com to [set up Microsoft Entra ID as an IdP](#).

Note

It's recommended you name the IdP integration in relation to the target application. For example, **Microsoft Entra ID - Customer management portal**.

Configure Conditional Access

1. Sign in to the Microsoft Entra admin center [↗](#) as at least a Cloud Application Administrator.
2. Browse to Identity > Applications > App registrations > All applications
3. Select the application you created.
4. Go to Branding & properties.
5. For Home page URL, enter the application hostname.

The screenshot shows the 'Customer management portal' branding & properties page in the Microsoft Entra admin center. The left sidebar lists various management options like Overview, Quickstart, Integration assistant, and several tabs under 'Manage'. The 'Branding & properties' tab is selected. On the right, there are fields for Name (set to 'Customer management portal'), Logo (None provided), Upload new logo (Select a file), Home page URL (http://crm...com), Terms of service URL, Privacy statement URL, Service management reference, and Internal notes (Add information relevant to the management of this application). At the bottom are 'Save' and 'Discard' buttons.

6. Browse to Identity > Applications > Enterprise applications > All applications.
7. Select your application.
8. Select Properties.
9. For Visible to users, select Yes. This action enables the app to appear in App Launcher and in [My Apps](#) [↗](#).
10. Under Security, select Conditional Access.
11. See, [Building a Conditional Access policy](#).
12. Create and enable other policies for the application.

Create a Cloudflare Access application

Enforce Conditional Access policies on a Cloudflare Access application.

1. Go to dash.cloudflare.com to sign in to Cloudflare ↗.
2. In **Zero Trust**, go to **Access**.
3. Select **Applications**.
4. See, [Add a self-hosted application](#) ↗.
5. In **Application domain**, enter the protected application target URL.
6. For **Identity providers**, select the IdP integration.
7. Create an Access policy. See, [Access policies](#) ↗ and the following example.

 **Note**

Reuse the IdP integration for other applications if they require the same Conditional Access policies. For example, a baseline IdP integration with a Conditional Access policy requiring multifactor authentication and a modern authentication client. If an application requires specific Conditional Access policies, set up a dedicated IdP instance for that application.

Next steps

- [What is Conditional Access?](#)
- [Secure Hybrid Access with Microsoft Entra ID partner integrations](#)
- [Tutorial: Configure Cloudflare with Microsoft Entra ID for secure hybrid access](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) ↗

Tutorial: Configure Secure Hybrid Access with Microsoft Entra ID and Datawiza

Article • 01/30/2024

In this tutorial, learn how to integrate Microsoft Entra ID with [Datawiza](#) for [hybrid access](#). [Datawiza Access Proxy \(DAP\)](#) extends Microsoft Entra ID to enable single sign-on (SSO) and provide access controls to protect on-premises and cloud-hosted applications, such as Oracle E-Business Suite, Microsoft IIS, and SAP. With this solution, enterprises can transition from legacy web access managers (WAMs), such as Symantec SiteMinder, NetIQ, Oracle, and IBM, to Microsoft Entra ID without rewriting applications. Enterprises can use Datawiza as a no-code, or low-code, solution to integrate new applications to Microsoft Entra ID. This approach enables enterprises to implement their Zero Trust strategy while saving engineering time and reducing costs.

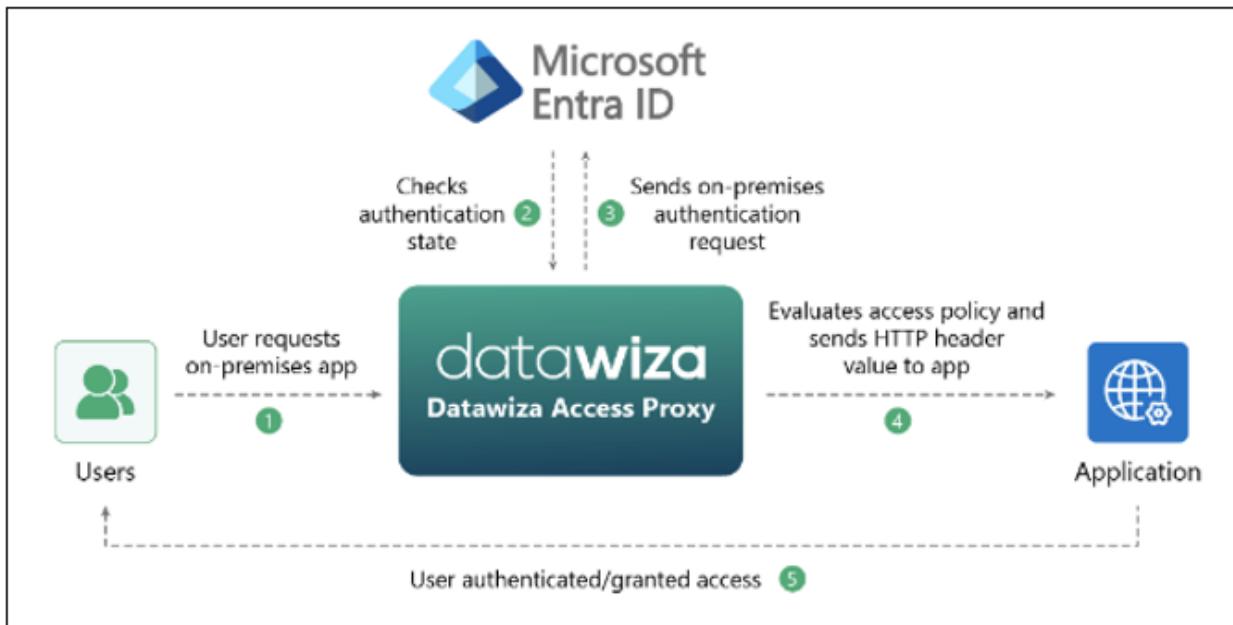
Learn more: [Zero Trust security](#)

Datawiza with Microsoft Entra authentication Architecture

Datawiza integration includes the following components:

- **Microsoft Entra ID** - Identity and access management service that helps users sign in and access external and internal resources
- **Datawiza Access Proxy (DAP)** - This service transparently passes identity information to applications through HTTP headers
- **Datawiza Cloud Management Console (DCMC)** - UI and RESTful APIs for administrators to manage the DAP configuration and access control policies

The following diagram illustrates the authentication architecture with Datawiza in a hybrid environment.



1. The user requests access to the on-premises or cloud-hosted application. DAP proxies the request to the application.
2. DAP checks user authentication state. If there's no session token, or the session token is invalid, DAP sends the user request to Microsoft Entra ID for authentication.
3. Microsoft Entra ID sends the user request to the endpoint specified during DAP registration in the Microsoft Entra tenant.
4. DAP evaluates policies and attribute values to be included in HTTP headers forwarded to the application. DAP might call out to the identity provider to retrieve the information to set the header values correctly. DAP sets the header values and sends the request to the application.
5. The user is authenticated and is granted access.

Prerequisites

To get started, you need:

- An Azure subscription
 - If you don't have one, you can get an [Azure free account](#)
- An [Microsoft Entra tenant](#) linked to the Azure subscription
- [Docker](#) and [docker-compose](#) are required to run DAP
 - Your applications can run on platforms, such as a virtual machine (VM) or bare metal
- An on-premises or cloud-hosted application to transition from a legacy identity system to Microsoft Entra ID
 - In this example, DAP is deployed on the same server as the application

- The application runs on localhost: 3001. DAP proxies traffic to the application via localhost: 9772
- The traffic to the application reaches DAP, and is proxied to the application

Configure Datawiza Cloud Management Console

1. Sign in to [Datawiza Cloud Management Console ↗](#) (DCMC).
2. Create an application on DCMC and generate a key pair for the app:
`PROVISIONING_KEY` and `PROVISIONING_SECRET`.
3. To create the app and generate the key pair, follow the instructions in [Datawiza Cloud Management Console ↗](#).
4. Register your application in Microsoft Entra ID with [One Click Integration With Microsoft Entra ID ↗](#).

Configure IdP

Configure the IdP with the following fields. Each IdP config in Datawiza maps to one application inside an IdP's tenant.

Name *

Organization Domain ⓘ

Protocol *

Identity Provider *

Automatic Generator

Click the "Create" button and it will ask you to consent permissions to automatically create a new application on your behalf in your Azure tenant. If you'd like to integration with an existing application, please choose the "Off" of this option

Supported account types *

Account in this organizational directory only (Single tenant)

Accounts in any organizational directory (Any AD directory - Multitenant)

Accounts in any organizational directory (Any AD directory - Multitenant) and personal Microsoft account

Personal Microsoft accounts only

Who can use this application or access this API?

Tenant ID ⓘ

If you want to manipulate the tenant which is not your default tenant, you can choose to input tenant id here

Previous Cancel Create

5. To use a web application, manually populate form fields: **Tenant ID**, **Client ID**, and **Client Secret**.

Learn more: To create a web application and obtain values, go to docs.datawiza.com for [Microsoft Entra ID](#) documentation.

X Create a New Deployment (Step 3 of 4)

Configure IdP

Configure the IdP with the following fields. Each IdP config in Datawiza maps to one application inside an IdP's tenant.

Name *

Organization Domain ⓘ

Protocol *

Identity Provider *

Automatic Generator

Click the "Create" button and it will ask you to consent permissions to automatically create a new application on your behalf in your Azure tenant. If you'd like to integration with an existing application, please choose the "Off" of this option

Tenant ID * ⓘ

Client ID * ⓘ

Client Secret * ⓘ

Issuer * ⓘ

Scopes * ⓘ

[Previous](#)

[Cancel](#)

[Create](#)

6. Run DAP using either Docker or Kubernetes. The docker image is needed to create a sample header-based application.

- For Kubernetes, see [Deploy Datawiza Access Proxy with a Web App using Kubernetes ↗](#)
- For Docker, see [Deploy Datawiza Access Proxy With Your App ↗](#)
 - You can use the following sample docker image docker-compose.yml file:

YAML

```
services:
  datawiza-access-broker:
    image: registry.gitlab.com/datawiza/access-broker
    container_name: datawiza-access-broker
    restart: always
    ports:
      - "9772:9772"
    environment:
      PROVISIONING_KEY: #####
      PROVISIONING_SECRET: #####
  header-based-app:
    image: registry.gitlab.com/datawiza/header-based-app
    restart: always
ports:
  - "3001:3001"
```

7. Sign in to the container registry.
8. Download the DAP images and the header-based application in this [Important Step](#).
9. Run the following command: `docker-compose -f docker-compose.yml up`.
10. The header-based application has SSO enabled with Microsoft Entra ID.
11. In a browser, go to `http://localhost:9772/`.
12. A Microsoft Entra sign-in page appears.
13. Pass user attributes to the header-based application. DAP gets user attributes from Microsoft Entra ID and passes attributes to the application via a header or cookie.
14. To pass user attributes such as email address, first name, and last name to the header-based application, see [Pass User Attributes](#).
15. To confirm configured user attributes, observe a green check mark next to each attribute.

Datawiza Access Broker Demo Application - Home Page

This page uses the Datawiza Access Broker to demonstrate header and cookie-based app integration and URL level authorization

Attributes				Raw
#	Attribute	Location	Description	Value
✓	host	Header	Application Host	localhost:9772
✓	email	Header	User email	xxx@datawiza.com
✓	firstname	Header	First Name	John
✓	lastname	Header	Last Name	Doe

Test the flow

1. Go to the application URL.
2. DAP redirects you to the Microsoft Entra sign-in page.
3. After authentication, you're redirected to DAP.
4. DAP evaluates policies, calculates headers, and sends you to the application.
5. The requested application appears.

Next steps

- Tutorial: [Configure Azure Active Directory B2C with Datawiza to provide secure hybrid access](#)
- Tutorial: [Configure Datawiza to enable Microsoft Entra multifactor authentication and SSO to Oracle JD Edwards](#)
- Tutorial: [Configure Datawiza to enable Microsoft Entra multifactor authentication and SSO to Oracle PeopleSoft](#)
- Tutorial: [Configure Datawiza to enable Microsoft Entra multifactor authentication and SSO to Oracle Hyperion EPM](#)
- Go to docs.datawiza.com for Datawiza [User Guides](#)

Tutorial: Configure Datawiza to enable Microsoft Entra multifactor authentication and single sign-on to Oracle JD Edwards

Article • 01/30/2024

In this tutorial, learn how to enable Microsoft Entra single sign-on (SSO) and Microsoft Entra multifactor authentication for an Oracle JD Edwards (JDE) application using Datawiza Access Proxy (DAP).

Learn more [Datawiza Access Proxy](#)

Benefits of integrating applications with Microsoft Entra ID using DAP:

- Embrace proactive security with Zero Trust - a security model that adapts to modern environments and embraces hybrid workplace, while it protects people, devices, apps, and data
- Microsoft Entra single sign-on - secure and seamless access for users and apps, from any location, using a device
- How it works: Microsoft Entra multifactor authentication - users are prompted during sign-in for forms of identification, such as a code on their cellphone or a fingerprint scan
- What is Conditional Access? - policies are if-then statements, if a user wants to access a resource, then they must complete an action
- Easy authentication and authorization in Microsoft Entra ID with no-code Datawiza - use web applications such as: Oracle JDE, Oracle E-Business Suite, Oracle Sibel, and home-grown apps
- Use the Datawiza Cloud Management Console (DCMC) - manage access to applications in public clouds and on-premises

Scenario description

This scenario focuses on Oracle JDE application integration using HTTP authorization headers to manage access to protected content.

In legacy applications, due to the absence of modern protocol support, a direct integration with Microsoft Entra SSO is difficult. DAP can bridge the gap between the

legacy application and the modern ID control plane, through protocol transitioning. DAP lowers integration overhead, saves engineering time, and improves application security.

Scenario architecture

The scenario solution has the following components:

- **Microsoft Entra ID** - identity and access management service that helps users sign in and access external and internal resources
- **Oracle JDE application** - legacy application protected by Microsoft Entra ID
- **Datawiza Access Proxy (DAP)** - container-based reverse-proxy that implements OpenID Connect (OIDC), OAuth, or Security Assertion Markup Language (SAML) for user sign-in flow. It passes identity transparently to applications through HTTP headers.
- **Datawiza Cloud Management Console (DCMC)** -a console to manage DAP. Administrators use UI and RESTful APIs to configure DAP and access control policies.

Learn more: [Datawiza and Microsoft Entra authentication Architecture](#)

Prerequisites

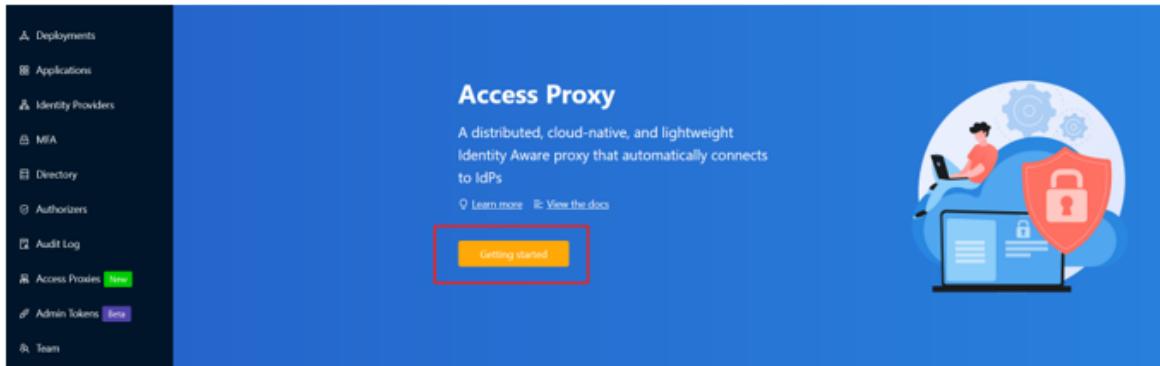
Ensure the following prerequisites are met.

- An Azure subscription.
 - If you don't have one, you can get an [Azure free account](#)
- A Microsoft Entra tenant linked to the Azure subscription
 - See, [Quickstart: Create a new tenant in Microsoft Entra ID](#).
- Docker and Docker Compose
 - Go to docs.docker.com to [Get Docker](#) and [Install Docker Compose](#)
- User identities synchronized from an on-premises directory to Microsoft Entra ID, or created in Microsoft Entra ID and flowed back to an on-premises directory
 - See, [Microsoft Entra Connect Sync: Understand and customize synchronization](#)
- An account with Microsoft Entra ID and a global administrator role. See, [Microsoft Entra built-in roles, all roles](#)
- An Oracle JDE environment
- (Optional) An SSL web certificate to publish services over HTTPS. You can also use default Datawiza self-signed certs for testing

Getting started with DAB

To integrate Oracle JDE with Microsoft Entra ID:

1. Sign in to [Datawiza Cloud Management Console](#). ↗
2. The Welcome page appears.
3. Select the orange **Getting started** button.



4. In the **Name** and **Description** fields, enter information.
5. Select **Next**.

X Create a New Deployment (Step 1 of 4)

Deployment Name

Setting up Deployment is quite easy and simple, you just need to follow the instructions. After several steps, the setup will be finished in a few minutes.

Name	<input type="text" value="Oracle JDE demo"/>
Description	<input type="text" value="Add a description in less than 250 characters"/>
<input type="button" value="Cancel"/>	<input type="button" value="Next"/>

6. On the **Add Application** dialog, for **Platform**, select **Web**.
7. For **App Name**, enter a unique application name.
8. For **Public Domain**, for example enter <https://jde-external.example.com>. For testing the configuration, you can use localhost DNS. If you aren't deploying DAP

behind a load balancer, use the **Public Domain** port.

9. For **Listen Port**, select the port that DAP listens on.
10. For **Upstream Servers**, select the Oracle JDE implementation URL and port to be protected.
11. Select **Next**.

X Create a New Deployment (Step 2 of 4)

Add Application

Configure your application with the following fields. All these configurations can be changed later.

Only one application can be added at this time, but you can add more later

* Platform	WEB
* App Name	Oracle JDE App
Description	Description
* Public Domain ⓘ	https://jde.datawiza.net
* Listen Port ⓘ	443
If not using Load Balancer, sync from public domain	
Upstream Servers ⓘ	https://10.0.0.122
Add URL	

Previous Cancel Next

12. On the **Configure IdP** dialog, enter information.

ⓘ Note

Use DCMC one-click integration to help complete Microsoft Entra configuration. DCMC calls the Graph API to create an application registration on your behalf in your Microsoft Entra tenant. Go to docs.datawiza.com for **One Click Integration With Microsoft Entra ID**.

13. Select **Create**.

Configure IdP

Configure the IdP with the following fields. Each IdP config in Datawiza maps to one application inside an IdP's tenant.

Name *

Organization Domain ⓘ

Protocol *

Identity Provider *

Automatic Generator

Click the "Create" button and it will ask you to consent permissions to automatically create a new application on your behalf in your Azure tenant. If you'd like to integration with an existing application, please choose the "Off" of this option

Supported account types * Account in this organizational directory only (Single tenant) Accounts in any organizational directory (Any AD directory - Multitenant) Accounts in any organizational directory (Any AD directory - Multitenant) and personal Microsoft account Personal Microsoft accounts only

Who can use this application or access this API?

Tenant ID ⓘ

If you want to manipulate the tenant which is not your default tenant, you can choose to input tenant id here

14. The DAP deployment page appears.

15. Make a note of the deployment Docker Compose file. The file includes the DAP image, Provisioning Key, and Provision Secret, which pulls the latest configuration and policies from DCMC.

Provision Key ✖

Provision Secret ✖

Step 1 Login Docker Registry

```
docker login registry.gitlab.com -u datawiza-deploy-token
```

Step 2 Pull Access Broker docker image

```
docker pull registry.gitlab.com/datawiza/access-broker:latest
```

Step 3 For docker compose env, use the following snippet to deploy Access Broker Image

```
version: "3"
services:
  # Configuration of Datawiza Access Broker image begins here.
  # Unless you know what you are doing, do not modify this block.
  # If you need any assistant, please contact: info@datawiza.com
  datawiza-access-broker:
    image: registry.gitlab.com/datawiza/access-broker
    container_name: datawiza-access-broker
    ports:
      - "443:443"
    restart: always
    environment:
      PROVISIONING_KEY: ::1
      PROVISIONING_SECRET: ::1
```

SSO and HTTP headers

DAP gets user attributes from IdP and passes them to the upstream application with a header or cookie.

The Oracle JDE application needs to recognize the user: using a name, the application instructs DAP to pass the values from the IdP to the application through the HTTP header.

1. In Oracle JDE, from the left navigation, select **Applications**.
2. Select the **Attribute Pass** subtab.
3. For **Field**, select **Email**.
4. For **Expected**, select **JDE_SSO_UID**.

5. For Type, select Header.

The screenshot shows the DataWiza application interface. On the left, a sidebar lists various options: Deployments, Applications (which is selected and highlighted with a red box), IdPs, B2B Connections, Directory, Authorizers, Audit Log, Access Brokers (New), Admin Tokens (Beta), Team, Plan, and Settings. The main area displays the 'Oracle JDE app' details under the 'WEB' category. The 'Attribute Pass' tab is selected (highlighted with a red box). A note below the tabs states: 'Define what user attributes (e.g., firstName and lastName) will be sent to the proxied application via header or cookie.' Below this, there is a table with one row: 'Field' (email), 'Expected' (JDE_SSO_UID), 'Type' (HEADER), and buttons for 'Edit' and 'Delete'. Navigation icons < > and a page number '1' are at the bottom right of the table.

ⓘ Note

This configuration uses the Microsoft Entra user principal name as the sign-in username, used by Oracle JDE. To use another user identity, go to the **Mappings** tab.

The screenshot shows the 'Profile' mapping configuration between 'azure a...' and 'Profile'. It lists several mappings:

Azure AD Field	Profile Expected Field
groups	groups
displayName	username
surname	lastName
userPrincipalName	email
givenName	firstName

A red box highlights the 'userPrincipalName' field in the Azure AD column. At the bottom right, there are 'Cancel' and 'Save mappings' buttons.

6. Select the Advanced tab.

datawiza

Deployments Applications IdPs B2B Connections Directory Authorizers Audit Log Access Brokers **New** Admin Tokens **Beta** Team Plan Settings

< Home / Deployments / Deployment Detail / Application Detail

Oracle JDE App WEB

General IdP Configuration Profile Mappings Attribute Pass Rules Conditional Access Kerberos **Advanced**

Advanced setting is for profession administrator use. If you are not familiar with these settings, you can keep all default.

SSL

Enable SSL **On**

Browser — Access Broker — Application

* Cert Type: Upload (File Based)

Select Option: File Based

* Cert: Upload (jde.datawiza.net_chain.crt size: 4KB)

datawiza

Deployments Applications **IdPs** B2B Connections Directory Authorizers Audit Log Access Brokers **New** Admin Tokens **Beta** Team Plan Settings

< Home / Applications / Application Detail

Oracle JDE app WEB

General IdP Configuration Profile Mappings **Attribute Pass** Rules Conditional Access Kerberos Advanced

Define what user attributes (e.g., firstName and lastName) will be sent to the proxied application via header or cookie.

Add New Attribute Pass

Field	Expected	Type
email	JDE_SSO_UID	HEADER

< 1 >

7. Select **Enable SSL**.

8. From the **Cert Type** dropdown, select a type.

datawiza

Deployments Applications Identity Providers Directory Authorizers Audit Log Access Brokers **New** Admin Tokens **Beta** Team Plan Settings

< Home / Deployments / Deployment Detail / Application Detail

Oracle JDE App WEB

General IdP Configuration Profile Mappings Attribute Pass Rules **Advanced**

Advanced setting is for profession administrator use. If you are not familiar with these settings, you can keep all default.

SSL Advanced Options JWT Conditional Access Kerberos

Enable SSL **Off** Edit

Browser — Access Broker — Application

SSL (jde.datawiza.net_chain.crt size: 4KB)

9. For testing purposes, we'll be providing a self-signed certificate.

Advanced Settings

Advanced setting is for profession administrator use. If you are not familiar with these settings, you can keep all default.

SSL Advanced Options JWT Conditional Access Kerberos

Enable SSL

Cert Type *

Please choose your cert type

Self Signed

Upload

! Note

You have the option to upload a certificate from a file.

Advanced Settings

Advanced setting is for profession administrator use. If you are not familiar with these settings, you can keep all default.

SSL Advanced Options JWT Conditional Access Kerberos

Enable SSL

Cert Type *

Upload

Select Option

File Based

Cert *

Upload

 jde.datawiza.net_chain.crt size: 4KB, MD5: 47ffcd36ab

Private Key *

Upload

 jde.datawiza.net_key.key size: 2KB, MD5: a6c980db8f

Cancel

Save

10. Select Save.

Enable Microsoft Entra multifactor authentication

Tip

Steps in this article might vary slightly based on the portal you start from.

To provide more security for sign-ins, you can enforce MFA for user sign-in.

See, [Tutorial: Secure user sign-in events with Microsoft Entra multifactor authentication](#).

1. Sign in to the [Microsoft Entra admin center](#) as a **Global Administrator**.
2. Browse to **Identity > Overview > Properties** tab.
3. Under **Security defaults**, select **Manage security defaults**.
4. On the **Security defaults** pane, toggle the dropdown menu to select **Enabled**.
5. Select **Save**.

Enable SSO in the Oracle JDE EnterpriseOne Console

To enable SSO in the Oracle JDE environment:

1. Sign in to the Oracle JDE EnterpriseOne Server Manager Management Console as an **Administrator**.
2. In **Select Instance**, select the option above **EnterpriseOne HTML Server**.
3. In the **Configuration** tile, select **View as Advanced**.
4. Select **Security**.
5. Select the **Enable Oracle Access Manager** checkbox.
6. In the **Oracle Access Manager Sign-Off URL** field, enter **datawiza/ab-logout**.
7. In the **Security Server Configuration** section, select **Apply**.
8. Select **Stop**.

ⓘ Note

If a message states the web server configuration (jas.ini) is out-of-date, select **Synchronize Configuration**.

9. Select **Start**.

Test an Oracle JDE-based application

To test an Oracle JDE application, validate application headers, policy, and overall testing. If needed, use header and policy simulation to validate header fields and policy execution.

To confirm Oracle JDE application access occurs, a prompt appears to use a Microsoft Entra account for sign-in. Credentials are checked and the Oracle JDE appears.

Next steps

- [Video Enable SSO and MFA for Oracle JDE\) with Microsoft Entra ID via Datawiza ↗](#)
- [Tutorial: Configure Secure Hybrid Access with Microsoft Entra ID and Datawiza](#)
- [Tutorial: Configure Azure AD B2C with Datawiza to provide secure hybrid access](#)
- Go to docs.datawiza.com for Datawiza [User Guides ↗](#)

Tutorial: Configure Datawiza to enable Microsoft Entra multifactor authentication and single sign-on to Oracle PeopleSoft

Article • 01/30/2024

In this tutorial, learn how to enable Microsoft Entra single sign-on (SSO) and Microsoft Entra multifactor authentication for an Oracle PeopleSoft application using Datawiza Access Proxy (DAP).

Learn more: [Datawiza Access Proxy ↗](#)

Benefits of integrating applications with Microsoft Entra ID using DAP:

- Embrace proactive security with Zero Trust ↗ - a security model that adapts to modern environments and embraces hybrid workplace, while it protects people, devices, apps, and data
- Microsoft Entra single sign-on ↗ - secure and seamless access for users and apps, from any location, using a device
- How it works: Microsoft Entra multifactor authentication - users are prompted during sign-in for forms of identification, such as a code on their cellphone or a fingerprint scan
- What is Conditional Access? - policies are if-then statements, if a user wants to access a resource, then they must complete an action
- Easy authentication and authorization in Microsoft Entra ID with no-code Datawiza ↗ - use web applications such as: Oracle JDE, Oracle E-Business Suite, Oracle Sibel, and home-grown apps
- Use the Datawiza Cloud Management Console ↗ (DCMC) - manage access to applications in public clouds and on-premises

Scenario description

This scenario focuses on Oracle PeopleSoft application integration using HTTP authorization headers to manage access to protected content.

In legacy applications, due to the absence of modern protocol support, a direct integration with Microsoft Entra SSO is difficult. Datawiza Access Proxy (DAP) bridges the gap between the legacy application and the modern ID control plane, through

protocol transitioning. DAP lowers integration overhead, saves engineering time, and improves application security.

Scenario architecture

The scenario solution has the following components:

- **Microsoft Entra ID** - identity and access management service that helps users sign in and access external and internal resources
- **Datawiza Access Proxy (DAP)** - container-based reverse-proxy that implements OpenID Connect (OIDC), OAuth, or Security Assertion Markup Language (SAML) for user sign-in flow. It passes identity transparently to applications through HTTP headers.
- **Datawiza Cloud Management Console (DCMC)** - administrators manage DAP with UI and RESTful APIs to configure DAP and access control policies
- **Oracle PeopleSoft application** - legacy application to be protected by Microsoft Entra ID and DAP

Learn more: [Datawiza and Microsoft Entra authentication architecture](#)

Prerequisites

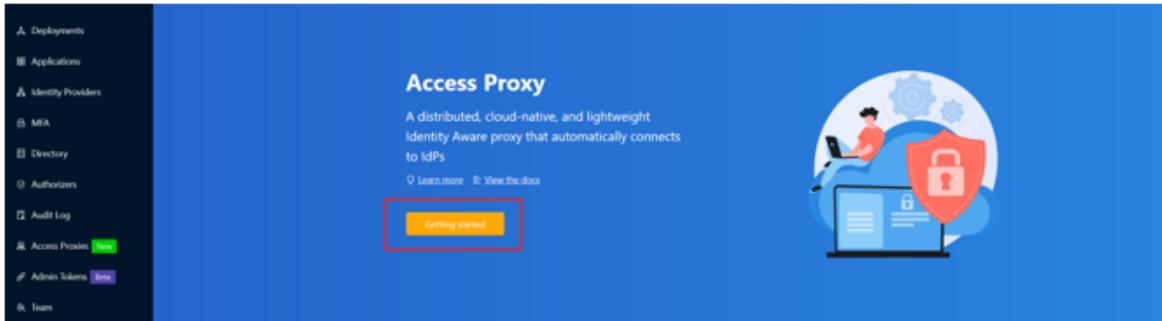
Ensure the following prerequisites are met.

- An Azure subscription
 - If you don't have one, you can get an [Azure free account](#)
- A Microsoft Entra tenant linked to the Azure subscription
 - See, [Quickstart: Create a new tenant in Microsoft Entra ID](#)
- Docker and Docker Compose
 - Go to docs.docker.com to [Get Docker](#) and [Install Docker Compose](#)
- User identities synchronized from an on-premises directory to Microsoft Entra ID, or created in Microsoft Entra ID and flowed back to an on-premises directory
 - See, [Microsoft Entra Connect Sync: Understand and customize synchronization](#)
- An account with Microsoft Entra ID and the Application Administrator role
 - See, [Microsoft Entra built-in roles, all roles](#)
- An Oracle PeopleSoft environment
- (Optional) An SSL web certificate to publish services over HTTPS. You can use default Datawiza self-signed certs for testing.

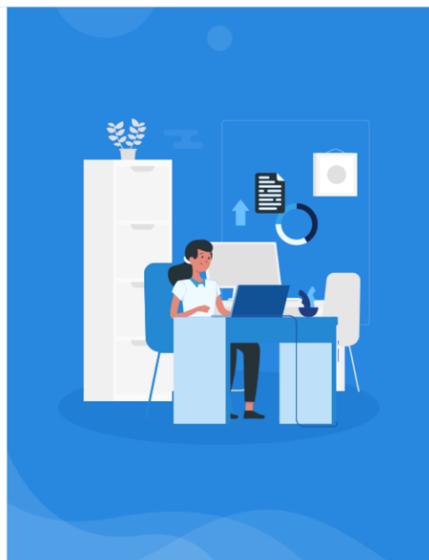
Getting started with DAP

To integrate Oracle PeopleSoft with Microsoft Entra ID:

1. Sign in to [Datawiza Cloud Management Console](#) (DCMC).
2. The Welcome page appears.
3. Select the orange **Getting started** button.



4. In the **Name** and **Description** fields, enter information.



X Create a New Deployment (Step 1 of 4)

Deployment Name

Setting up Deployment is quite easy and simple, you just need to follow the instructions. After several steps, the setup will be finished in a few minutes.

Name
Oracle peoplesoft demo

Description
Add a description in less than 250 characters
0 / 1024

Cancel **Next**

5. Select **Next**.
6. The Add Application dialog appears.
7. For **Platform**, select **Web**.
8. For **App Name**, enter a unique application name.
9. For **Public Domain**, for example use `https://ps-external.example.com`. For testing, you can use localhost DNS. If you aren't deploying DAP behind a load balancer, use the Public Domain port.
10. For **Listen Port**, select the port that DAP listens on.

11. For **Upstream Servers**, select the Oracle PeopleSoft implementation URL and port to be protected.

X Create a New Deployment (Step 2 of 4)

Add Application

Configure your application with the following fields. All these configurations can be changed later.

! Only one application can be added at this time, but you can add more later

* Platform	WEB	
* App Name	peoplesoft demo app	
Description	Description	
* Public Domain ⓘ	https://ps.datawiza.net	
* Listen Port ⓘ	443	
If not using Load Balancer, sync from public domain		
Upstream Servers ⓘ	http://10.0.0.165:8000	
Add URL		
Previous	Cancel	Next

12. Select **Next**.

13. On the **Configure IdP** dialog, enter information.

ⓘ Note

DCMC has one-click integration to help complete Microsoft Entra configuration.
DCMC calls the Microsoft Graph API to create an application registration on your

behalf in your Microsoft Entra tenant. Learn more at docs.datawiza.com in One Click Integration with Microsoft Entra ID ↗

14. Select Create.

Configure IdP

Configure the IdP with the following fields. Each IdP config in Datawiza maps to one application inside an IdP's tenant.

Name *

Organization Domain ⓘ

Protocol *

Identity Provider * Microsoft Entra ID

Automatic Generator

Click the "Create" button and it will ask you to consent permissions to automatically create a new application on your behalf in your Azure tenant. If you'd like to integration with an existing application, please choose the "Off" of this option

Supported account types * Account in this organizational directory only (Single tenant) Accounts in any organizational directory (Any AD directory - Multitenant) Accounts in any organizational directory (Any AD directory - Multitenant) and personal Microsoft account Personal Microsoft accounts only

Who can use this application or access this API?

Tenant ID ⓘ

If you want to manipulate the tenant which is not your default tenant, you can choose to input tenant id here

[Previous](#) [Cancel](#) [Create](#)

15. The DAP deployment page appears.

16. Make a note of the deployment Docker Compose file. The file includes the DAP image, the Provisioning Key and Provision Secret, which pulls the latest configuration and policies from DCMC.

Provisioning Key

KL9



Provisioning Secret

SDB



Step 1 Login Docker Registry

```
docker login registry.gitlab.com -u datawiza-deploy-token -p pZN27
```

Step 2 Pull Access Proxy docker image

```
docker pull registry.gitlab.com/datawiza/access-proxy:latest
```

Step 3 For docker compose env, use the following snippet to deploy Access Proxy Image

```
version: "3"
services:
  # Configuration of Datawiza Access Proxy image begins here.
  # Unless you know what you are doing, do not modify this block.
  # If you need any assistance, please contact: info@datawiza.com
  datawiza-access-proxy:
    image: registry.gitlab.com/datawiza/access-proxy
    container_name: datawiza-access-proxy
    ports:
      - 443:443
    restart: always
    environment:
      PROVISIONING_KEY: KL9
      PROVISIONING_SECRET: SDB
```

Done

SSO and HTTP headers

DAP gets user attributes from the identity provider (IdP) and passes them to the upstream application with a header or cookie.

The Oracle PeopleSoft application needs to recognize the user. Using a name, the application instructs DAP to pass the values from the IdP to the application through the HTTP header.

1. In Oracle PeopleSoft, from the left navigation, select Applications.

2. Select the **Attribute Pass** subtab.

3. For **Field**, select **email**.

4. For **Expected**, select **PS_SSO_UID**.

5. For **Type**, select **Header**.

The screenshot shows the Datawiza application interface. On the left is a sidebar with various navigation options: Deployments (selected), Applications, IdPs, JDBC Connections, Directory, Authorizers, Audit Log, Access Brokers (New), Admin Tokens (864), Team, Plan, and Settings. The main area displays the 'peoplesoft demo app' under 'WEB'. At the top, there are tabs: General, IdP Configuration, Profile, Mappings, Attribute Pass (which is highlighted with a red box), Rules, Conditional Access, Kerberos, and Advanced. Below the tabs, it says 'Define what user attributes (e.g., firstName and lastName) will be sent to the proxied application via header or cookie.' A button 'Add New Attribute Pass' is visible. A table below lists one entry: Field (email), Expected (PS_SSO_UID), Type (HEADER). There are 'Edit' and 'Delete' buttons next to the entry, along with navigation arrows. The bottom of the screen shows a footer with '© 2022 Datawiza Version 2.1.4' and a search icon.

⚠ Note

This configuration uses Microsoft Entra user principal name as the sign-in username for Oracle PeopleSoft. To use another user identity, go to the **Mappings** tab.

The screenshot shows the Azure Active Directory B2B mapping configuration. On the left, under 'azure a...', there are fields: 'groups', 'displayName', 'surname', 'userPrincipalName' (which is highlighted with a red box), and 'givenName'. On the right, under 'Profile', there are corresponding fields: 'groups', 'username', 'lastName', 'email', and 'firstName'. A large green arrow points from the left side to the right side, indicating the mapping. At the bottom right are 'Cancel' and 'Save mappings' buttons.

SSL Configuration

1. Select the Advanced tab.

The screenshot shows the Datawiza application configuration page for a 'peoplesoft demo app'. At the top, there's a navigation bar with links to Home, Deployments, Deployment Detail, Application Detail, and a back arrow. Below the navigation is a section for 'peoplesoft demo app' with a 'WEB' icon. A horizontal menu bar includes General, IdP Configuration, Profile, Mappings, Attribute Pass, Rules, Conditional Access, Kerberos, and Advanced. The 'Advanced' tab is highlighted with a red box. A note below the tabs states: 'Advanced setting is for profession administrator use. If you are not familiar with these settings, you can keep all default.' Under the 'SSL' section, there's a toggle switch labeled 'Enable SSL' which is currently off. To its right is a diagram illustrating the flow from a 'Browser' through an 'Access Broker' to an 'Application', with locks indicating secure connections at each point. Below the SSL section is a 'Advanced Options' group with fields for Custom Logout URL (set to '/ab-logout'), Logout Redirect URI (placeholder 'Please input your logout redirect uri'), Cookie Domain (placeholder 'Please input your cookie domain, e.g., yourcompany.com'), and Proxy Url Prefix (set to '/datawiza/').

2. Select Enable SSL.

3. From the Cert Type dropdown, select a type.

This screenshot shows the same Datawiza interface as the previous one, but with a different focus. The left sidebar shows various navigation items like Deployments, Applications, Identity Providers, etc. The main content area has the 'Advanced' tab selected again. Within the 'Advanced' tab, the 'SSL' sub-tab is highlighted with a red box. Below it, there's a field labeled 'Enable SSL' with a dropdown menu showing 'Off'. To the right of this field is a large 'Edit' button, also highlighted with a red box. The central part of the screen contains the same network diagram as before, showing the flow from Browser to Access Broker to Application with locks.

4. For testing the configuration, there's a self-signed certificate.

Enable SSL



* Cert Type

Self Signed

! Note

You can upload a certificate from a file.

Advanced Settings

Advanced setting is for professional administrator use. If you are not familiar with these settings, you can keep all default.

[SSL](#) [Advanced Options](#) [JWT](#) [Conditional Access](#) [Kerberos](#)

Enable SSL

Cert Type *

Upload

Select Option

File Based

Cert *

Upload

jde.datawiza.net_chain.crt size: 4KB, MD5: 47ffcd36ab

Private Key *

Upload

jde.datawiza.net_key.key size: 2KB, MD5: a6c980db8f

Cancel

Save

5. Select Save.

Enable Microsoft Entra multifactor authentication

Tip

Steps in this article might vary slightly based on the portal you start from.

To provide more security for sign-ins, you can enforce Microsoft Entra multifactor authentication.

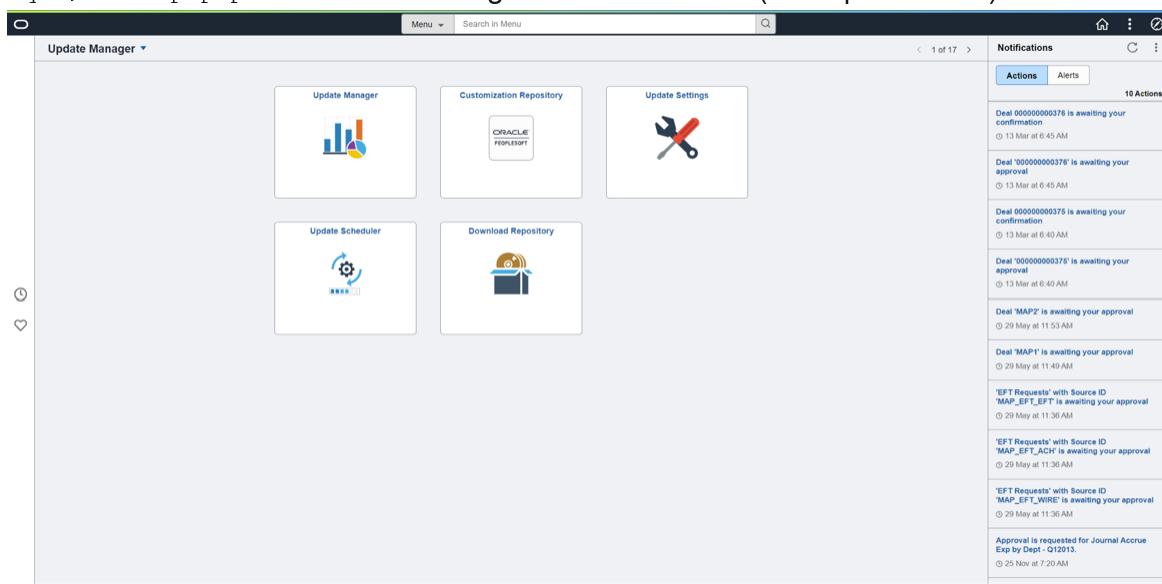
Learn more: [Tutorial: Secure user sign-in events with Microsoft Entra multifactor authentication](#)

1. Sign in to the [Microsoft Entra admin center](#) as a **Global Administrator**.
2. Browse to **Identity > Overview > Properties tab**.
3. Under **Security defaults**, select **Manage security defaults**.
4. On the **Security defaults** pane, toggle the dropdown menu to select **Enabled**.
5. Select **Save**.

Enable SSO in the Oracle PeopleSoft console

To enable SSO in the Oracle PeopleSoft environment:

1. Sign in to the PeopleSoft Console <http://{your-peoplesoft-fqdn}:8000/psp/ps/?cmd=start> using Admin credentials, for example, PS/PS.



2. Add a default public access user to PeopleSoft.
3. From the main menu, navigate to **PeopleTools > Security > User Profiles > User Profiles > Add a New Value**.
4. Select **Add a new value**.
5. Create user **PSPUBUSER**.
6. Enter the password.

User Profiles

User ID PSPUBUSER

Account Locked Out?

Description

Logon Information

Symbolic ID SYSADM1

Change Password? New Password

Password Expired? Confirm Password

User ID Alias

Edit Email Addresses Instant Messaging Information

General Attributes

Language English

Enable Expert Entry

Currency

Default Mobile Page

Permission Lists

Navigator Homepage

Primary

Process Profile

Row Security

Action Buttons

Save Return to Search Add Update/Display

General | ID | Roles | Workflow | Audit | Links | User ID Queries

7. Select the **ID** tab.

8. For **ID Type**, select **None**.

User Profiles

User ID PSPUBUSER

Description

ID Types and Values

*ID Type:

Attribute Name **Attribute Value** **Description**

User Description

Description:

Set Description or type in User Description.

Action Buttons

Save Return to Search Add Update/Display

General | ID | Roles | Workflow | Audit | Links | User ID Queries

9. Navigate to PeopleTools > Web Profile > Web Profile Configuration > Search > PROD > Security.

10. Under Public Users, select the Allow Public Access box.

11. For User ID, enter PSPUBUSER.

12. Enter the password.

The screenshot shows the 'Web Profile Config' page with the 'Profile Name' set to 'PROD'. In the 'Public Users' section, the 'Allow Public Access' checkbox is checked, and the 'User ID' is set to 'PSPUBUSER'. A red box highlights this section. Below it, the 'Web Server Jolt Settings' and 'XML Link' sections are visible. At the bottom, there are 'Save', 'Return to Search', 'Notify', 'Previous tab', 'Next tab', 'Add', and 'Update/Display' buttons.

13. Select Save.

14. To enable SSO, navigate to **PeopleTools > Security > Security Objects > Signon PeopleCode**.

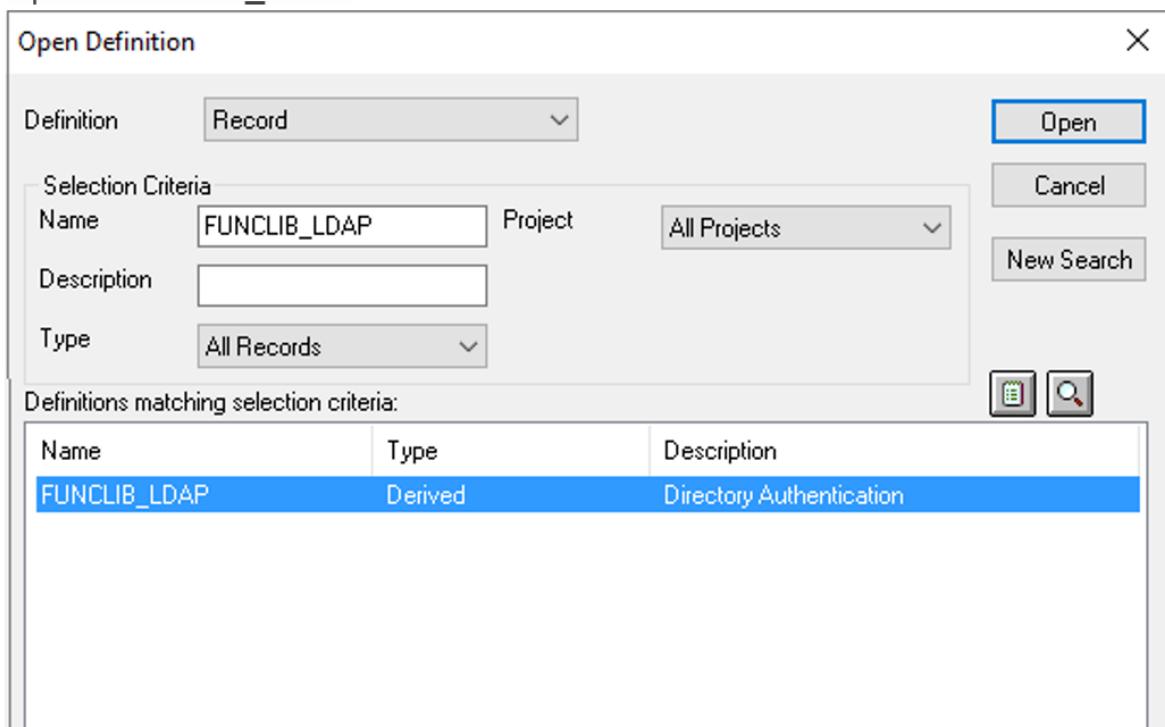
15. Select the **Sign on PeopleCode** page.

16. Enable **OAMSSO_AUTHENTICATION**.

17. Select Save.

18. To configure PeopleCode using the PeopleTools application designer, navigate to **File > Open > Definition: Record > Name: FUNCLIB_LDAP**.

19. Open **FUNCLIB_LDAP**.



20. Select the record.
21. Select **LDAPAUTH > View PeopleCode**.
22. Search for the `getWWWAuthConfig()` function `Change &defaultUserId = "";` to
`&defaultUserId = PSPUBUSER.`
23. Confirm the user Header is `PS_SSO_UID` for `OAMSSO_AUTHENTICATION` function.
24. Save the record definition.

LDAPAUTH [field] FieldDefault

```

/////////////////////////////
Function OAMSSO_AUTHENTICATION()
  If %PSAuthResult = True And
    &authMethod <> "LDAP" And
    &authMethod <> "WWW" And
    &authMethod <> "OSO" And
    &authMethod <> "SSO" Then
    getWWWAuthConfig();
  If &signonUserId = &defaultUserId Then
    &userID = %Request.GetHeader("PS_SSO_UID"); /*This header is used by IDCS*/
    REM &userID = %Request.GetHeader("OAM_REMOTE_USER"); /*This header is used by and del:
    If &userID <> "" Then
      If &bConfigRead = False Then
        getLDAPConfig();
      End-If;
      SetAuthenticationResult( True, Upper(&userID), "", False);
      &authMethod = "OAMSSO";
    End-If;
  End-If;
End-If;

End-Function;

```

Test an Oracle PeopleSoft application

To test an Oracle PeopleSoft application, validate application headers, policy, and overall testing. If needed, use header and policy simulation to validate header fields and policy execution.

To confirm Oracle PeopleSoft application access occurs correctly, a prompt appears to use a Microsoft Entra account for sign-in. Credentials are checked and the Oracle PeopleSoft appears.

Next steps

- Video: Enable SSO and MFA for Oracle JD Edwards with Microsoft Entra ID via [Datawiza ↗](#)
- Tutorial: Configure Secure Hybrid Access with Microsoft Entra ID and Datawiza
- Tutorial: Configure Azure AD B2C with Datawiza to provide secure hybrid access
- Go to docs.datawiza.com for Datawiza [User Guides ↗](#)

Configure Datawiza for Microsoft Entra multifactor authentication and single sign-on to Oracle EBS

Article • 01/31/2024

In this article, learn how to enable Microsoft Entra multifactor authentication and single sign-on (SSO) for an Oracle E-Business Suite (Oracle EBS) application via Datawiza.

Here are some benefits of integrating applications with Microsoft Entra ID via Datawiza:

- A [Zero Trust](#) security model adapts to modern environments and embraces a hybrid workplace while it helps protect people, devices, apps, and data.
- [Single sign-on](#) provides secure and seamless access for device users and apps from any location.
- [Multifactor authentication](#) prompts users during sign-in for forms of identification, such as a code on their device or a fingerprint scan.
- [Conditional Access](#) provides policies as if/then statements. If a user wants to access a resource, then they must complete an action.
- [Datawiza](#) provides authentication and authorization in Microsoft Entra ID with no code. Use web applications such as Oracle JDE, Oracle EBS, Oracle Siebel, and home-grown apps.
- Use the [Datawiza Cloud Management Console](#) (DCMC) to manage access to applications in public clouds and on-premises.

This article focuses on modern identity providers (IdPs) integrating with the legacy Oracle EBS application. The application requires a set of Oracle EBS service account credentials and an Oracle EBS database container (DBC) file.

Architecture

The solution has the following components:

- **Microsoft Entra ID:** Microsoft's cloud-based identity and access management service, which helps users sign in and access external and internal resources.
- **Oracle EBS:** The legacy application that Microsoft Entra ID will help protect.
- **Datawiza Access Proxy (DAP):** A lightweight container-based reverse proxy that implements OIDC/OAuth or SAML for user sign-on flow. It transparently passes identity to applications through HTTP headers.

- **DCMC:** A centralized management console that manages DAP. The console provides UI and RESTful APIs for administrators to manage the configurations of DAP and its granular access control policies.

Prerequisites

To complete the steps in this article, you need:

- An Azure subscription. If you don't have one, you can get an [Azure free account](#).
- A Microsoft Entra tenant linked to the Azure subscription.
- A [Global Administrator](#) role.
- Docker and Docker Compose, to run DAP. For more information, see [Get Docker](#) and [Docker Compose Overview](#).
- User identities synchronized from an on-premises directory to Microsoft Entra ID, or created in Microsoft Entra ID and flowed back to your on-premises directory. For more information, see [Microsoft Entra Connect Sync: Understand and customize synchronization](#).
- An Oracle EBS environment.

Configure the Oracle EBS environment for SSO and create the DBC file

To enable SSO in the Oracle EBS environment:

1. Sign in to the Oracle EBS management console as an administrator.
2. Scroll down the navigation pane, expand **User Management**, and then select **Users**.

Home

The screenshot shows the Oracle E-Business Suite Home page. On the left, there is a navigation tree with various modules like Knowledge Base Worker, Mobile Applications Manager, Oracle Trade Management User, etc. Under User Management, the 'Users' node is selected and highlighted with a red box. The main area displays a 'Worklist' table with columns: From, Type, Subject, Sent, and Due. The table lists numerous entries, mostly from System: Mailer, related to User Notification Preference Update Reports and Request Approvals. A 'Full List (557)' link is at the top right of the table, and a 'Rows 1 to 75' link is at the bottom right.

3. Add a user account. Select Create User > User Account.

The screenshot shows the Oracle User Management interface. The top navigation bar includes links for Home, Logout, and Logged In As SYSADMIN. The main menu has tabs for Users, Roles & Role Inheritance, Role Categories, Registration Processes, Security Report, Proxy Configuration, and Responsibility. The 'User Maintenance' section contains search fields for User Name, Email, Last Name, First Name, Organization, and Role. Below the search is a toolbar with Register, Go, and several icons. A dropdown menu shows 'Last Name' selected. To the right, there is a 'User Management' sidebar with two sections: 'Maintain User Accounts' (with options for Register new people, create/disable user accounts, and reset passwords) and 'Control Access' (with options for Grant access to different parts of the system by assigning/revoke roles). A magnifying glass icon is in the bottom right corner.

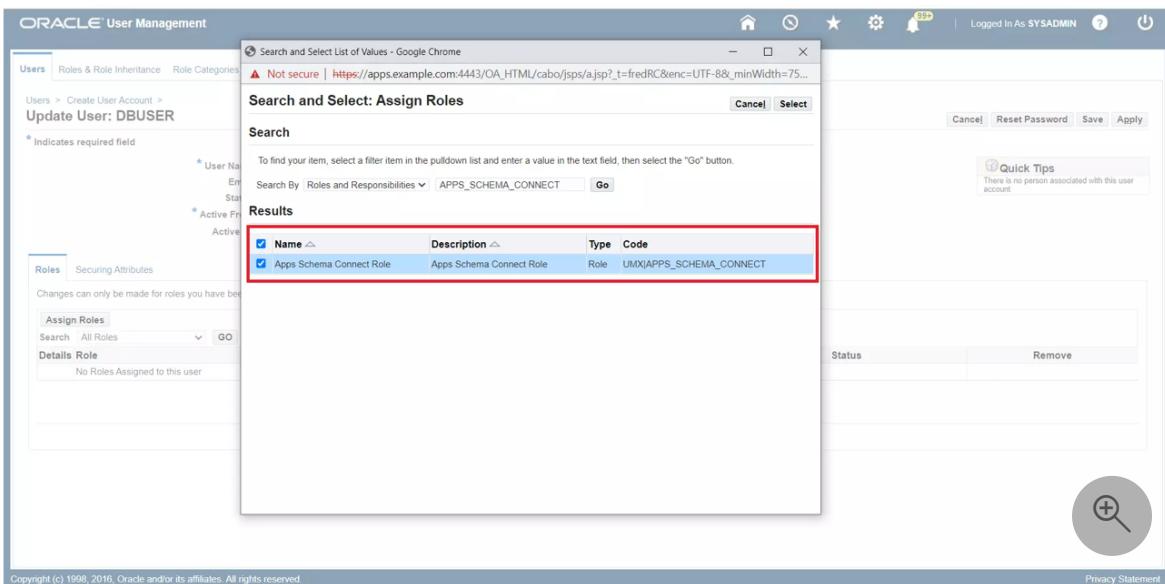
4. For User Name, enter DWSSOUSER.

5. For Password, enter a password.

6. For Description, enter DW User account for SSO.

7. For Password Expiration, select None.

8. Assign Apps Schema Connect Role to the user.



Register DAP with Oracle EBS

In the Oracle EBS Linux environment, generate a new DBC file for DAP. You need the app's user credentials and the default DBC file (under `$FND_SECURE`) that the application tier uses.

1. Configure the environment for Oracle EBS by using a command similar to

```
./u01/install/APPS/EBSapps.env run.
```

2. Use the AdminDesktop utility to generate the new DBC file. Specify the name of a new desktop node for this DBC file:

```
java oracle.apps.fnd.security.AdminDesktop apps/apps CREATE NODE_NAME=\<ebs
domain name>
DBC=/u01/install/APPS/fs1/inst/apps/EBSDB_apps/appl/fnd/12.0.0/secure/EBSDB.db
c
```

This action generates a file called `ebsdb_\<ebs domain name>.dbc` in the location where you ran the command.

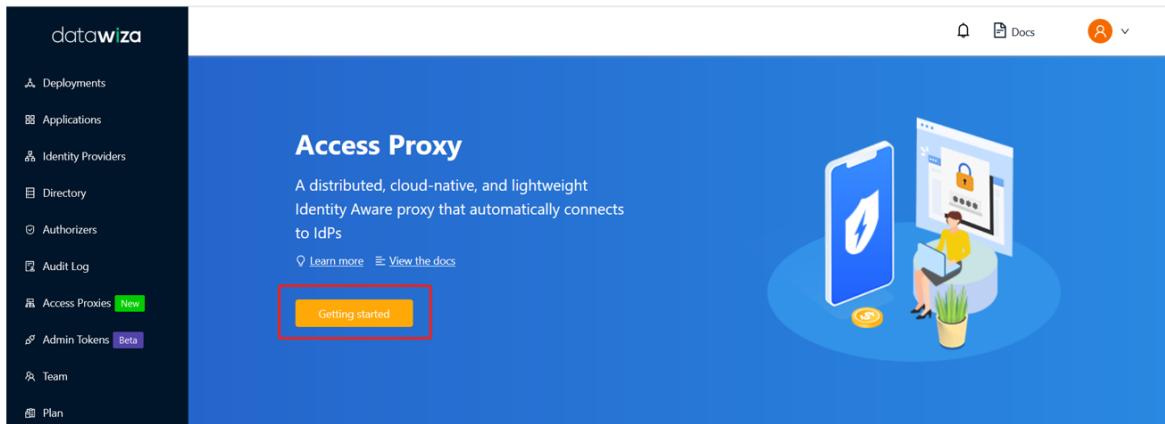
3. Copy the DBC file's content to a notebook. You'll use the content later.

Enable Oracle EBS for SSO

1. To integrate JDE with Microsoft Entra ID, sign in to the [Datawiza Cloud Management Console](#).

The welcome page appears.

2. Select the orange **Getting started** button.



3. For **Name**, enter a name for the deployment.

X Create a New Deployment (Step 1 of 4)

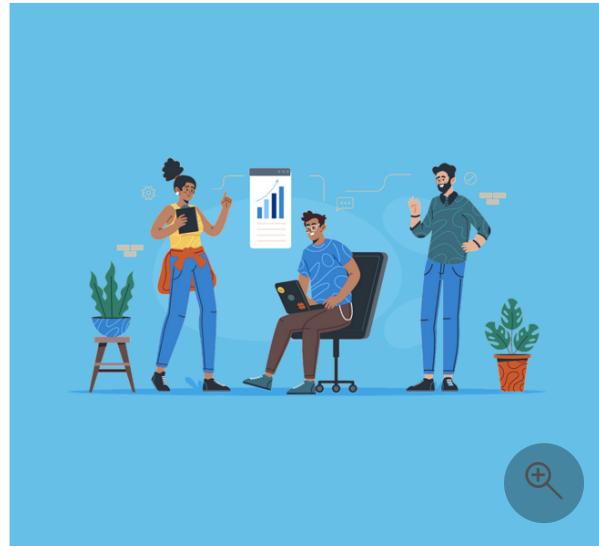
Deployment Name

Setting up Deployment is quite easy and simple, you just need to follow the instructions.
After several steps, the setup will be finished in a few minutes.

Name
Oracle EBS demo

Description
Add a description in less than 250 characters
0 / 1024

Cancel **Next**

A screenshot of a deployment creation form. The 'Name' field contains 'Oracle EBS demo' and is highlighted with a red box. The 'Description' field is empty. At the bottom are 'Cancel' and 'Next' buttons.

4. For **Description**, enter a description of the deployment.

5. Select **Next**.

6. On **Add Application**, for **Platform**, select **Oracle E-Business Suite**.

7. For **App Name**, enter the app name.

8. For **Public Domain**, enter the external-facing URL of the application. For example, enter `https://ebs-external.example.com`. You can use localhost DNS for testing.

9. For **Listen Port**, select the port that DAP listens on. You can use the port in **Public Domain** if you aren't deploying the DAP behind a load balancer.

10. For **Upstream Servers**, enter the URL and port combination of the Oracle EBS implementation that you want to protect.

11. For **EBS Service Account**, enter the username from the service account (**DWSSOUSER**).

12. For **EBS Account Password**, enter the password for the service account.
13. For **EBS User Mapping**, the product decides the attribute to be mapped to the Oracle EBS username for authentication.
14. For **EBS DBC Content**, use the content that you copied.

15. Select **Next**.

Only one application can be added at this time, but you can add more later

Platform * Oracle E-Business Suite

App Name * EBS App

Description

Public Domain * https://ebs.datawiza.net

Listen Port * 443

Upstream Servers * http://10.0.0.122:8000

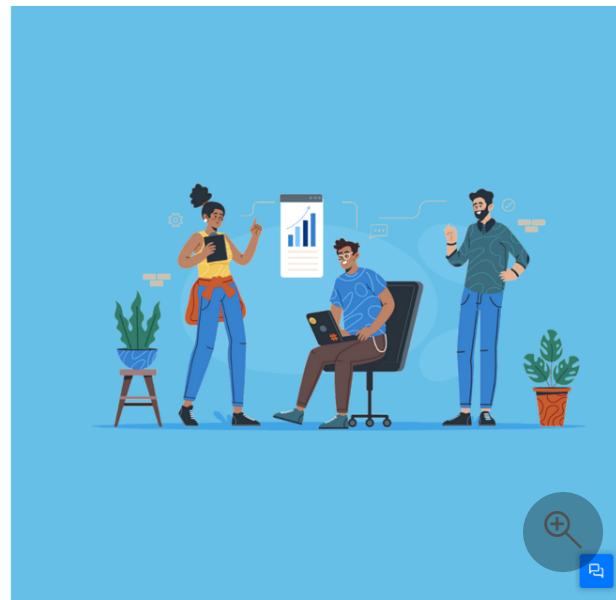
EBS Service Account * DWSSOUSER

EBS Account Password * *****

EBS User Mapping * email

EBS DBC Content * #Desktop DB Settings
#Tue Nov 22 05:34:09 GMT 2022
FNDNAM=APPS

Previous Cancel Next



IdP configuration

Use the DCMC one-click integration to help you complete Microsoft Entra configuration. With this feature, you can reduce management costs and the likelihood of configuration errors.

Configure IdP

Configure the IdP with the following fields. Each IdP config in Datawiza maps to one application inside an IdP's tenant.

Name * EBS demo IdP

Organization Domain ⓘ Example: datawiza.cloud

Protocol * OIDC

Identity Provider * Microsoft Entra ID

Automatic Generator

Click the "Create" button and it will ask you to consent permissions to automatically create a new application on your behalf in your Azure tenant. If you'd like to integration with an existing application, please choose the "Off" of this option

Supported account types *

- Account in this organizational directory only (Single tenant)
- Accounts in any organizational directory (Any AD directory - Multitenant)
- Accounts in any organizational directory (Any AD directory - Multitenant) and personal Microsoft account
- Personal Microsoft accounts only

Who can use this application or access this API?

Tenant ID ⓘ (OPTIONAL) Your Tenant ID

If you want to manipulate the tenant which is not your default tenant, you can choose to input tenant id here

[Previous](#) [Cancel](#) [Create !\[\]\(37c3088836d5e548bcac39ac06994438_img.jpg\)](#)

Docker Compose file

Configuration on the management console is complete. You're prompted to deploy DAP with your application. Make a note of the deployment Docker Compose file. The file includes the DAP image, `PROVISIONING_KEY`, and `PROVISIONING_SECRET`. DAP uses this information to pull the latest configuration and policies from DCMC.

Provisioning Key

KL9



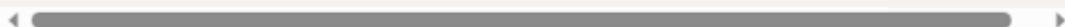
Provisioning Secret

SDB



Step 1 Login Docker Registry

```
docker login registry.gitlab.com -u datawiza-deploy-token -p pZN27
```



Step 2 Pull Access Proxy docker image

```
docker pull registry.gitlab.com/datawiza/access-proxy:latest
```



Step 3 For docker compose env, use the following snippet to deploy Access Proxy Image

```
version: "3"
services:
  # Configuration of Datawiza Access Proxy image begins here.
  # Unless you know what you are doing, do not modify this block.
  # If you need any assistance, please contact: info@datawiza.com
  datawiza-access-proxy:
    image: registry.gitlab.com/datawiza/access-proxy
    container_name: datawiza-access-proxy
    ports:
      - 443:443
    restart: always
    environment:
      PROVISIONING_KEY: KL9
      PROVISIONING_SECRET: SDB
```



Done

SSL configuration

1. For certificate configuration, select the **Advanced** tab on your application page. Then select **SSL > Edit**.

2. Turn on the **Enable SSL** toggle.
3. For **Cert Type**, select a certificate type.

There's a self-signed certificate for localhost. To use that certificate for testing, select **Self Signed**.

Optionally, you can upload a certificate from a file. For **Cert Type**, select **Upload**. Then, for **Select Option**, select **File Based**.

Advanced Settings

Advanced setting is for profession administrator use. If you are not familiar with these settings, you can keep all default.

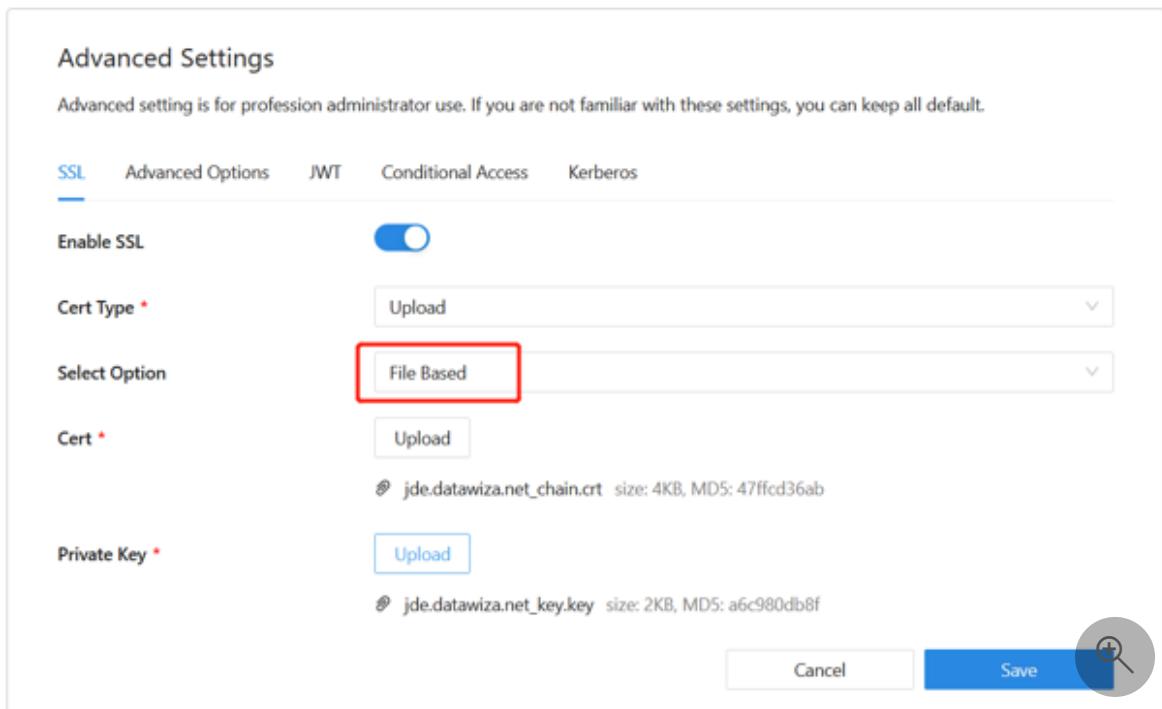
SSL Advanced Options JWT Conditional Access Kerberos

Enable SSL

Cert Type * Select Option File Based

Cert *  jde.datawiza.net_chain.crt size: 4KB, MD5: 47ffcd36ab

Private Key *  jde.datawiza.net_key.key size: 2KB, MD5: a6c980db8f



4. Select Save.

Optional: Enable multifactor authentication on Microsoft Entra ID

💡 Tip

Steps in this article might vary slightly based on the portal you start from.

To provide more security for sign-ins, you can enable multifactor authentication in the Microsoft Entra admin center:

1. Sign in to the [Microsoft Entra admin center](#) as a [Global Administrator](#).
2. Browse to **Identity > Overview > Properties tab**.
3. Under **Security defaults**, select **Manage security defaults**.
4. On the **Security defaults** pane, toggle the dropdown menu to select **Enabled**.
5. Select **Save**.

Next steps

- Video: [Enable SSO and MFA for Oracle JD Edwards with Microsoft Entra ID via Datawiza](#)
- Tutorial: [Configure Secure Hybrid Access with Microsoft Entra ID and Datawiza](#)
- Tutorial: [Configure Azure AD B2C with Datawiza to provide secure hybrid access](#)
- [Datawiza user guides](#)

Tutorial: Configure Datawiza to enable Microsoft Entra multifactor authentication and single sign-on to Oracle Hyperion EPM

Article • 11/02/2023

Use this tutorial to enable Microsoft Entra multifactor authentication and single sign-on (SSO) for Oracle Hyperion Enterprise Performance Management (EPM) using Datawiza Access Proxy (DAP).

Learn more on [datawiza.com](#).

Benefits of integrating applications with Microsoft Entra ID by using DAP:

- Embrace proactive security with Zero Trust - a security model that adapts to modern environments and embraces hybrid workplace, while it protects people, devices, apps, and data
- Microsoft Entra single sign-on - secure and seamless access for users and apps, from any location, using a device
- How it works: Microsoft Entra multifactor authentication - users are prompted during sign-in for forms of identification, such as a code on their cellphone, or a fingerprint scan
- What is Conditional Access? - policies are if-then statements, if a user wants to access a resource, then they must complete an action
- Easy authentication and authorization in Microsoft Entra ID with no-code Datawiza - use web applications such as: Oracle JDE, Oracle E-Business Suite, Oracle Siebel, and home-grown apps
- Use the Datawiza Cloud Management Console (DCMC) - manage access to applications in public clouds and on-premises

Scenario description

This scenario focuses on Oracle Hyperion EPM integration using HTTP authorization headers to manage access to protected content.

Due to the absence of modern protocol support in legacy applications, a direct integration with Microsoft Entra ID SSO is challenging. Datawiza Access Proxy (DAP) bridges the gap between the legacy application and the modern identity control plane,

through protocol transitioning. DAP lowers integration overhead, saves engineering time, and improves application security.

Scenario architecture

The solution has the following components:

- **Microsoft Entra ID** - identity and access management service that helps users sign in and access external and internal resources
- **Datawiza Access Proxy (DAP)** - container-based reverse-proxy that implements OpenID Connect (OIDC), OAuth, or Security Assertion Markup Language (SAML) for user sign-in flow. It passes identity transparently to applications through HTTP headers.
- **Datawiza Cloud Management Console (DCMC)** - administrators manage DAP with UI and RESTful APIs to configure DAP and access control policies
- **Oracle Hyperion EPM** - legacy application to be protected by Microsoft Entra ID and DAP

Learn about the service provider-initiated flow in [Datawiza with Microsoft Entra authentication architecture](#).

Prerequisites

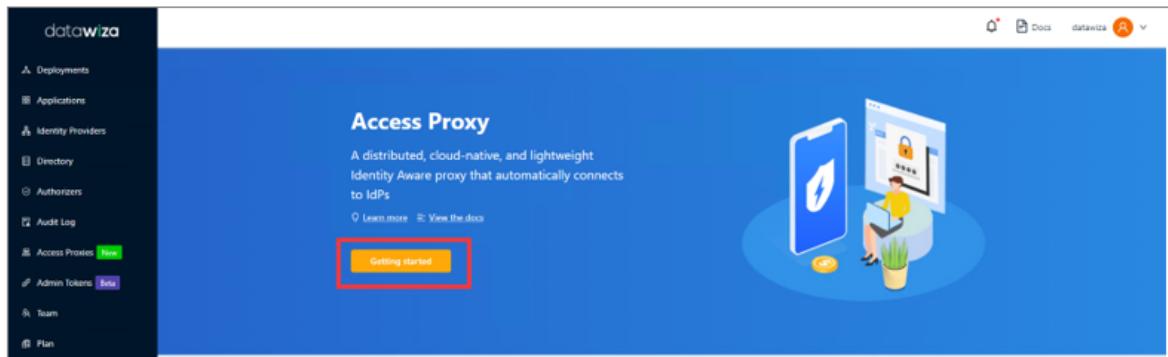
Ensure the following prerequisites are met:

- An Azure subscription
 - If you don't have one, you can get an [Azure free account](#)
- A Microsoft Entra ID tenant linked to the Azure subscription
 - See, [Quickstart: Create a new tenant in Microsoft Entra ID](#)
- Docker and Docker Compose
 - Go to docs.docker.com to [Get Docker](#) and [Install Docker Compose](#)
- User identities synchronized from an on-premises directory to Microsoft Entra ID, or created in Microsoft Entra ID and flowed back to an on-premises directory
 - See, [Microsoft Entra Connect Sync: Understand and customize synchronization](#)
- An account with Microsoft Entra ID and the Application Administrator role
 - See, [Microsoft Entra built-in roles, all roles](#)
- An Oracle Hyperion EPM environment
 - (Optional) An SSL web certificate to publish services over HTTPS. You can use default Datawiza self-signed certs for testing.

Getting started with DAP

To integrate Oracle Hyperion EMP with Microsoft Entra ID:

1. Sign in to [Datawiza Cloud Management Console](#) (DCMC).
2. The Welcome page appears.
3. Select the orange **Getting started** button.



4. Under Deployment Name in the Name and Description fields, enter information.

X Create a New Deployment (Step 1 of 4)

Deployment Name

Setting up Deployment is quite easy and simple, you just need to follow the instructions. After several steps, the setup will be finished in a few minutes.

Name *

Description 0 / 1024

5. Select Next.
6. The Add Application dialog appears.
7. For Platform, select Web.
8. For App Name, enter a unique application name.
9. For Public Domain, for example use `https://hyperion.example.com`. For testing, you can use localhost DNS. If you aren't deploying DAP behind a load balancer, use the Public Domain port.
10. For Listen Port, select the port that DAP listens on.

11. For **Upstream Servers**, select the Oracle Hyperion implementation URL and port to be protected.

12. Select **Next**.

13. On **Add Application**, enter information. Note the example entries for **Public Domain**, **Listen Port**, and **Upstream Servers**.

14. Select **Next**.

15. On the **Configure IdP** dialog, enter relevant information.

 **Note**

Use Datawiza Cloud Management Console (DCMC) [One Click Integration](#) to help complete configuration. DCMC calls the Microsoft Graph API to create an application registration on your behalf, in your Microsoft Entra ID tenant.

16. Select **Create**.

17. The DAP deployment page appears.

18. Make a note of the deployment Docker Compose file. The file includes the DAP image, also the Provisioning Key and Provisioning Secret, which pull the latest configuration and policies from DCMC.

Provisioning Key 

Provisioning Secret 

Step 1 Login Docker Registry

```
docker login registry.gitlab.com -u datawiza-deploy-token -p pZN2
```

Step 2 Pull Access Proxy docker image

```
docker pull registry.gitlab.com/datawiza/access-proxy:latest
```

Step 3 For docker compose env, use the following snippet to deploy Access Proxy Image

```
version: "3"
services:
  # Configuration of Datawiza Access Proxy image begins here.
  # Unless you know what you are doing, do not modify this block.
  # If you need any assistance, please contact: info@datawiza.com
  datawiza-access-proxy:
    image: registry.gitlab.com/datawiza/access-proxy
    container_name: datawiza-access-proxy
    ports:
      - 443:443
    restart: always
    environment:
      PROVISIONING_KEY: KL9
      PROVISIONING_SECRET: SDB
```

 Done



19. Select Done.

SSO and HTTP headers

DAP gets user attributes from the identity provider (IdP) and passes them to the upstream application with a header or cookie.

The following instructions enable Oracle Hyperion EPM application to recognize the user. Using a name, it instructs DAP to pass the values from the IdP to the application

through the HTTP header.

1. In the left navigation, select **Applications**.
2. Locate the application you created.
3. Select the **Attribute Pass** subtab.
4. For **Field**, select **email**.
5. For **Expected**, select **HYPLOGIN**.
6. For **Type**, select **Header**.

The screenshot shows the Oracle Hyperion Application Detail page for the 'Hyperion demo app'. The top navigation bar includes links for Home, Deployments, Deployment Detail, and Application Detail. Below the navigation is a card for the 'Hyperion demo app' with a globe icon and the text 'WEB'. The main content area has tabs: General, IdP Configuration, Profile, Mappings, Attribute Pass (which is highlighted with a red box), Rules, and Advanced. Under the Attribute Pass tab, there is a sub-section titled 'Define what user attributes (e.g., firstName and lastName) will be sent to the proxied application via header or cookie.' A blue button labeled 'Add New Attribute Pass' is visible. Below this, there is a table with three columns: Field, Expected, and Type. The first row shows 'email' in the Field column, 'HYPLOGIN' in the Expected column (which is also highlighted with a red box), and 'HEADER' in the Type column. To the right of the table are 'Edit' and 'Delete' buttons, and navigation arrows for the list. The table has a total count of 1 item.

(!) Note

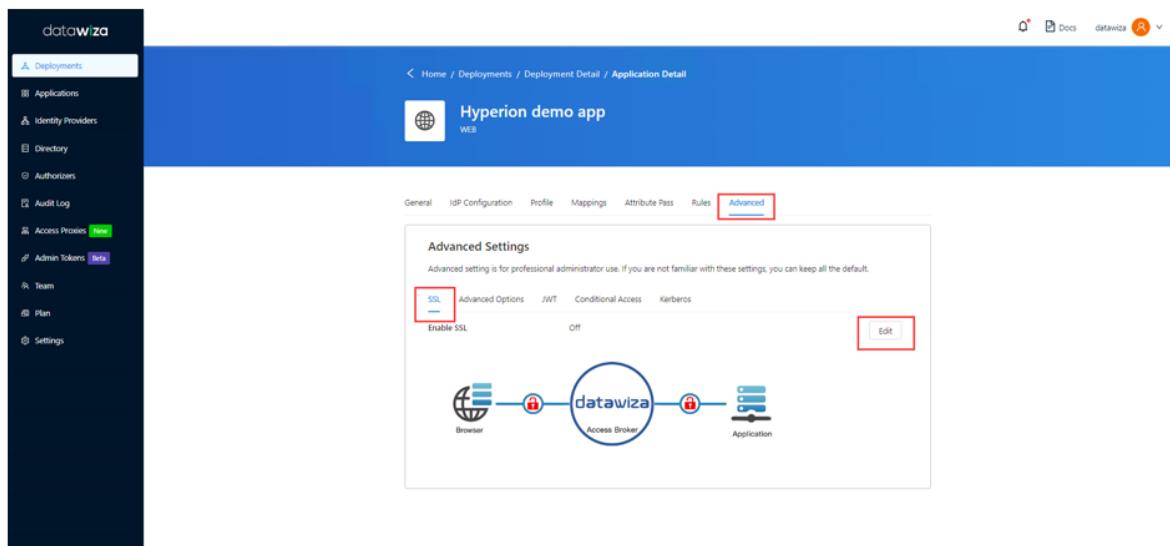
This configuration uses the Microsoft Entra ID user principal name for the sign in username, which is used by Oracle Hyperion. For another user identity, go to the **Mappings** tab.

The screenshot shows a mapping configuration screen. On the left, there are five input fields: 'groups', 'displayName', 'surname', 'userPrincipalName', and 'givenName'. The 'userPrincipalName' field is highlighted with a red border. On the right, there are five corresponding output fields: 'groups', 'username', 'lastName', 'email', and 'firstName'. Below the fields are two buttons: 'Cancel' and 'Save mappings'.

SSL configuration

Use the following instructions for SSL configuration.

1. Select the **Advanced** tab.



2. On the **SSL** tab, select **Enable SSL**.

3. From the **Cert Type** dropdown, select the type. For testing, there's a self-signed certificate.

Advanced Settings

Advanced setting is for professional administrator use. If you are not familiar with these settings, you can keep all the default.

SSL

Advanced Options

JWT

Conditional Access

Kerberos

Enable SSL



Cert Type *

Please choose your cert type

Self Signed

Upload

! Note

You can upload a certificate from a file.

Advanced Settings

Advanced setting is for profession administrator use. If you are not familiar with these settings, you can keep all default.

SSL

Advanced Options

JWT

Conditional Access

Kerberos

Enable SSL



Cert Type *

Upload

Select Option

File Based

Cert *

Upload

jde.datawiza.net_chain.crt size: 4KB, MD5: 47ffcd36ab

Private Key *

Upload

jde.datawiza.net_key.key size: 2KB, MD5: a6c980db8f

Cancel

Save

4. Select Save.

Login and Logout Redirect URI

Use the following instructions to indicate Login Redirect URI and Logout Redirect URI.

1. Select the Advanced Options tab.

2. For Login Redirect URI and Logout Redirect URI, enter `/workspace/index.jsp`.

The screenshot shows the Datawiza platform interface. On the left, there's a sidebar with various navigation options like Deployments, Applications, Identity Providers, Directory, Authorizers, Audit Log, Access Proxies (marked as New), Admin Tokens (Beta), Team, Plan, and Settings. The main area is titled 'Hyperion demo app WEB' and shows a 'Deployment Detail / Application Detail' view. At the top, there are tabs for General, IdP Configuration, Profile, Mappings, Attribute Pass, Rules, and Advanced. The Advanced tab is selected, leading to the 'Advanced Settings' page. This page has tabs for SSL, Advanced Options (which is selected), JWT, Conditional Access, and Kerberos. Under 'Advanced Options', there's a 'Single Sign Out' toggle switch followed by two input fields: 'Login Redirect URI' containing '/workspace/index.jsp' and 'Logout Redirect URI' also containing '/workspace/index.jsp'. Below these is a 'Logout DAP URI' field with the value '/datawiza/ab-logout'. A 'Skip Login Page' toggle switch is also present. A red box highlights the 'Logout Redirect URI' field.

3. Select Save.

Enable Microsoft Entra multifactor authentication

To provide more security for sign-ins, you can enforce Microsoft Entra multifactor authentication.

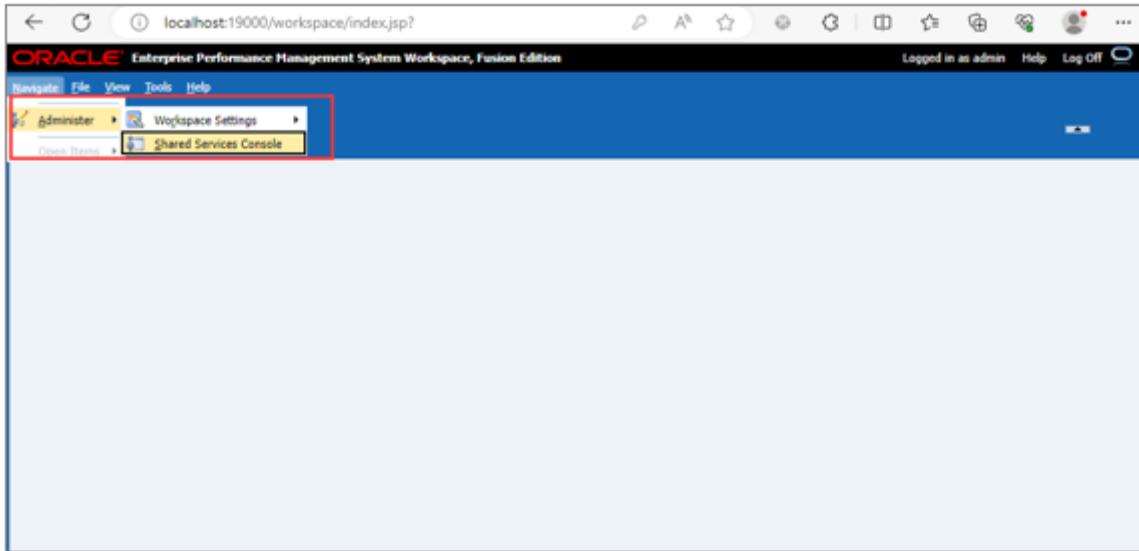
Learn more in the [Tutorial: Secure user sign-in events with Microsoft Entra multifactor authentication](#)

1. Sign in to the [Azure portal](#) as a [Global Administrator](#) role.
2. Select [Microsoft Entra ID](#) > [Manage](#) > [Properties](#).
3. Under [Properties](#) select [Manage security defaults](#).
4. Under [Enable Security Defaults](#), select [Yes](#).
5. Select [Save](#).

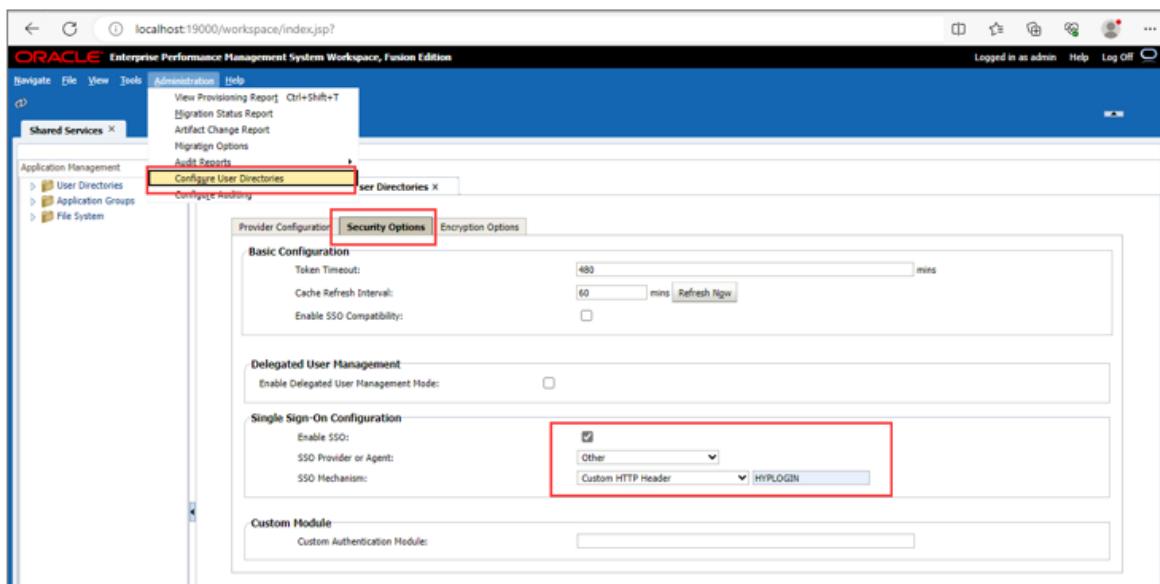
Enable SSO in the Oracle Hyperion Shared Services Console

Use the following instructions to enable SSO in the Oracle Hyperion environment.

1. Sign in to the Hyperion Shared Service Console with administrator permissions. For example, `http://{your-hyperion-fqdn}:19000/workspace/index.jsp`.
2. Select [Navigate](#), then [Shared Services Console](#).



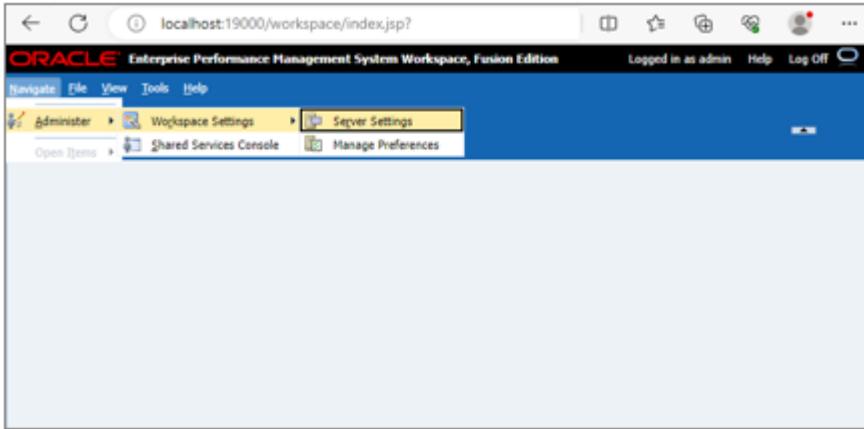
3. Select Administration and then **Configure User Directories**.
4. Select the **Security Options** tab.
5. In **Single Sign-On Configuration**, select the **Enable SSO** checkbox.
6. From the **SSO Provider or Agent** dropdown, select **Other**.
7. From the **SSO Mechanism** dropdown, select **Custom HTTP Header**.
8. In the following field, enter **HYPLOGIN**, the header name the security agent passes to EMP.
9. Select **OK**.



Update Post Log off URL settings in EMP Workspace

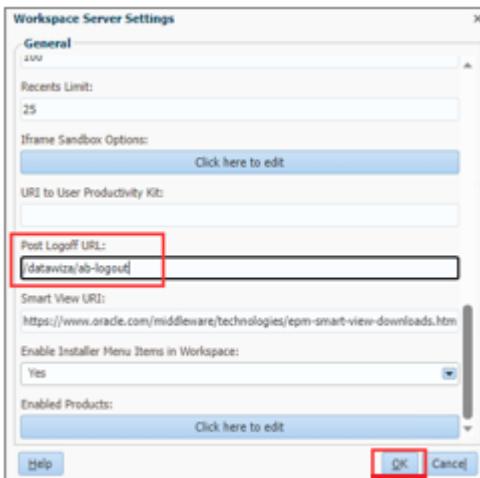
1. Select Navigate.

2. In Administer, select Workspace Settings then Server Settings.



3. On the **Workspace Server Settings** dialog, for Post Logoff URL, select the URL users see when they sign out of EPM, `/datawiza/ab-logout`.

4. Select OK.



Test an Oracle Hyperion EPM application

To confirm Oracle Hyperion application access, a prompt appears to use a Microsoft Entra ID account for sign-in. Credentials are checked and the Oracle Hyperion EPM home page appears.

Next steps

- Tutorial: Configure Secure Hybrid Access with Microsoft Entra ID and Datawiza
- Tutorial: Configure Azure AD B2C with Datawiza to provide secure hybrid access
- Go to Datawiza for [Add SSO and MFA to Oracle Hyperion EPM in minutes](#)
- Go to docs.datawiza.com for Datawiza [User Guides](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#) | [Get help at Microsoft Q&A](#)

Configure Datawiza Access Proxy for Microsoft Entra SSO and MFA for Outlook Web Access

Article • 07/02/2024

In this tutorial, learn how to configure Datawiza Access Proxy (DAP) to enable Microsoft Entra single sign-on (SSO) and Microsoft Entra multifactor authentication for Outlook Web Access (OWA). Help solve issues when modern identity providers (IdPs) integrate with legacy OWA, which supports Kerberos token authentication to identify users.

Often, legacy app and modern SSO integration are a challenge because there's no modern protocol support. Datawiza Access Proxy removes the protocol support gap, reduces integration overhead, and improves application security.

Integration benefits:

- Improved Zero Trust security with SSO, MFA, and Conditional Access:
 - See, [Embrace proactive security with Zero Trust ↗](#)
 - See, [What is Conditional Access?](#)
- No-code integration with Microsoft Entra ID and web apps:
 - OWA
 - Oracle JD Edwards
 - Oracle E-Business Suite
 - Oracle Siebel
 - Oracle PeopleSoft
 - Your apps
 - See, [Easy authentication and authorization in Microsoft Entra ID with no-code Datawiza ↗](#)
- Use the Datawiza Cloud Management Console (DCMC) to manage access to cloud and on-premises apps:
 - Go to [login.datawizwa.com ↗](https://login.datawizwa.com) to sign in or sign up for an account

Architecture

DAP integration architecture includes the following components:

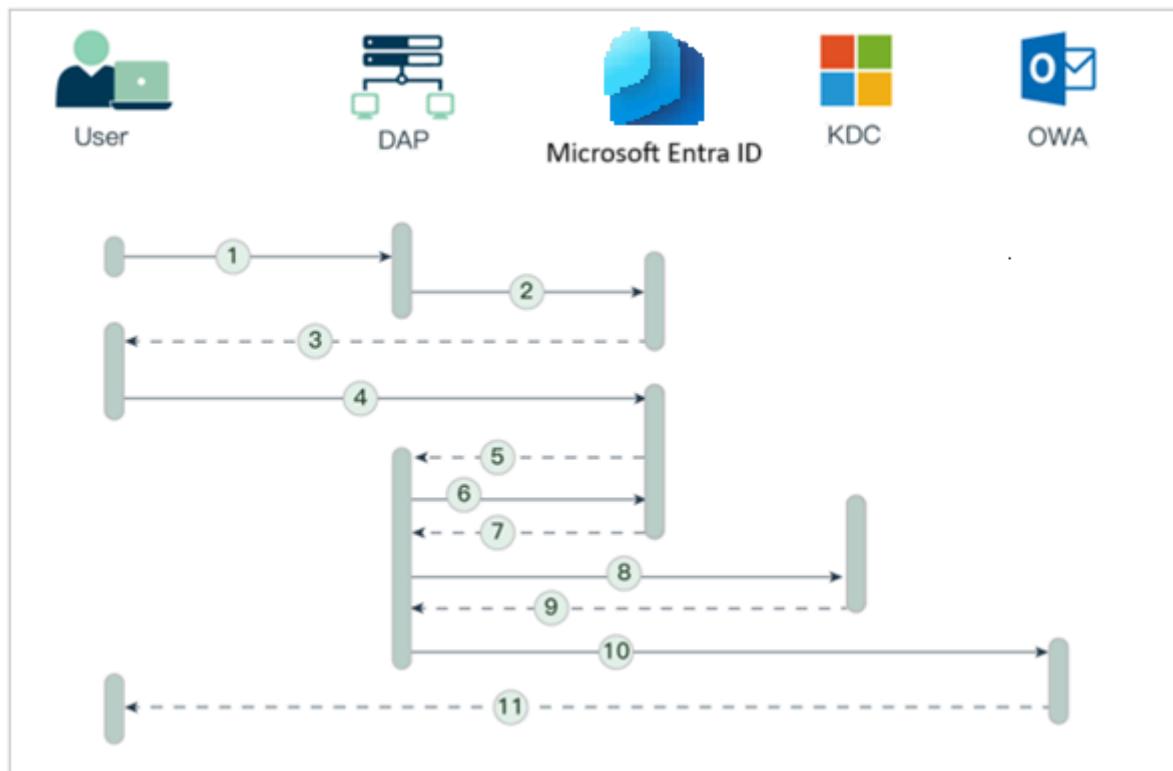
- **Microsoft Entra ID** - identity and access management service that helps users sign in and access external and internal resources

- **OWA** - the legacy, Exchange Server component to be protected by Microsoft Entra ID
- **Domain controller** - a server that manages user authentication and access to network resources in a Windows-based network
- **Key distribution center (KDC)** - distributes and manages secret keys and tickets in a Kerberos authentication system
- **DAP** - a reverse-proxy that implements OpenID Connect (OIDC), OAuth, or Security Assertion Markup Language (SAML) for user sign in. DAP integrates with protected applications by using:
 - HTTP headers
 - Kerberos
 - JSON web token (JWT)
 - other protocols
- **DCMC** - the DAP management console with UI and RESTful APIs to manage configurations and access control policies

The following diagram illustrates a user flow with DAP in a customer network.



The following diagram illustrates the user flow from user browser to OWA.



! Note

Subsequent user browser requests contain the Kerberos token, which enables access to OWA via DAP.

Prerequisites

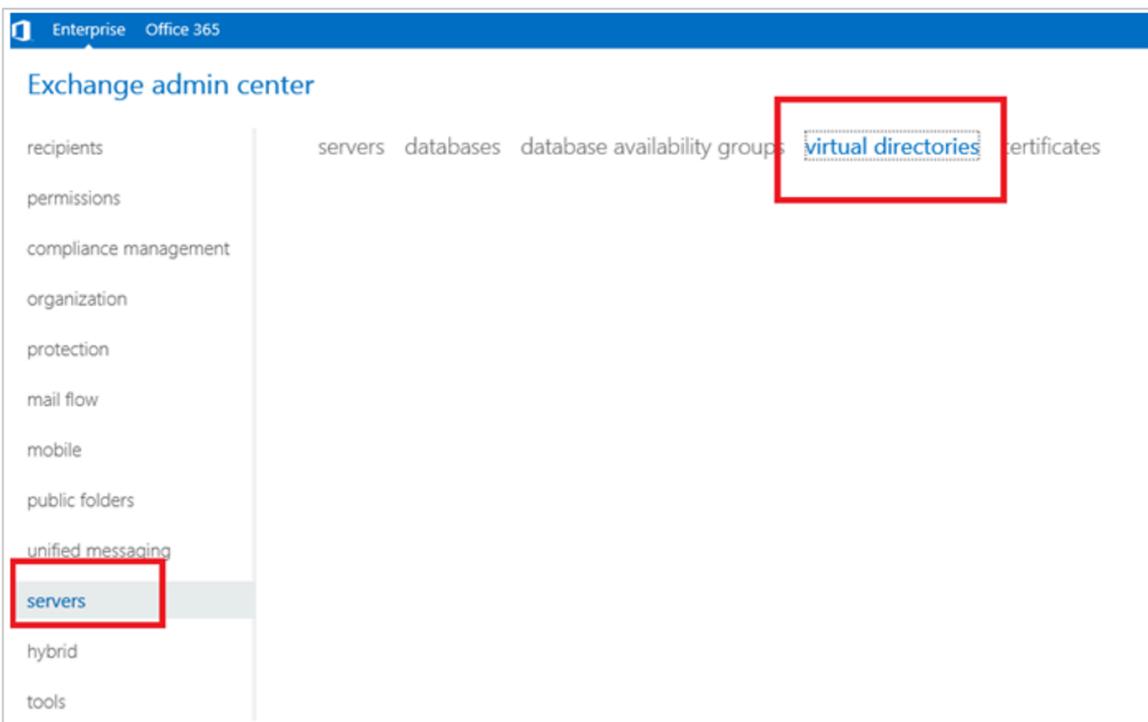
You need the following components. Prior DAP experience isn't necessary.

- An Azure account
 - If you don't have one, get an [Azure free account](#)

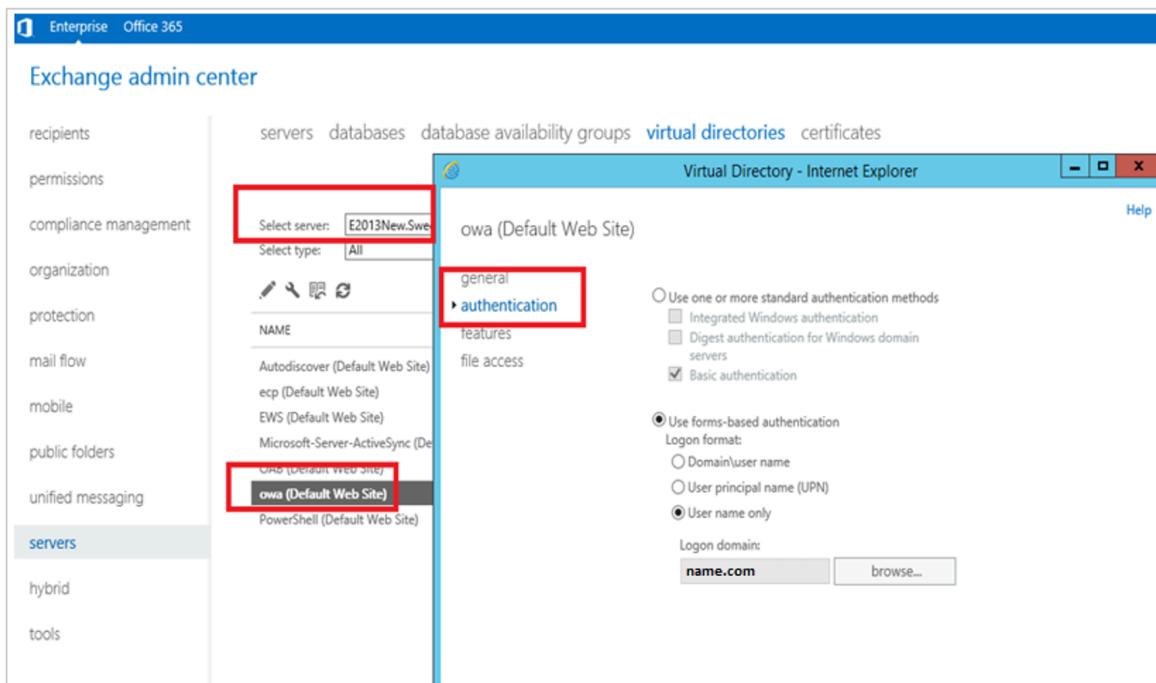
- A Microsoft Entra tenant linked to the Azure account
 - See, [Quickstart: Create a new tenant in Microsoft Entra ID](#)
- Docker and Docker Compose are required to run DAP
 - See, [Get Docker](#)
 - See, [Install Docker Compose](#), [Overview](#)
- User identities synchronized from an on-premises directory to Microsoft Entra ID, or created in Microsoft Entra ID and flowed back to your on-premises directory
 - See, [Microsoft Entra Connect Sync: Understand and customize synchronization](#)
- An account with Microsoft Entra Application Administrator permissions
 - See, Application Administrator and other roles on, [Microsoft Entra built-in roles](#)
- An Exchange Server environment. Supported versions:
 - Microsoft Internet Information Services (IIS) Integrated Windows Authentication (IWA) - IIS 7 or later
 - Microsoft OWA IWA - IIS 7 or later
- A Windows Server instance configured with IIS and Microsoft Entra services running as a domain controller (DC) and implementing Kerberos (IWA) SSO
 - It's unusual for large production environments to have an application server (IIS) that also functions as a DC.
- **Optional:** an SSL Web certificate to publish services over HTTPS, or DAP self-signed certificates, for testing.

Enable Kerberos authentication for OWA

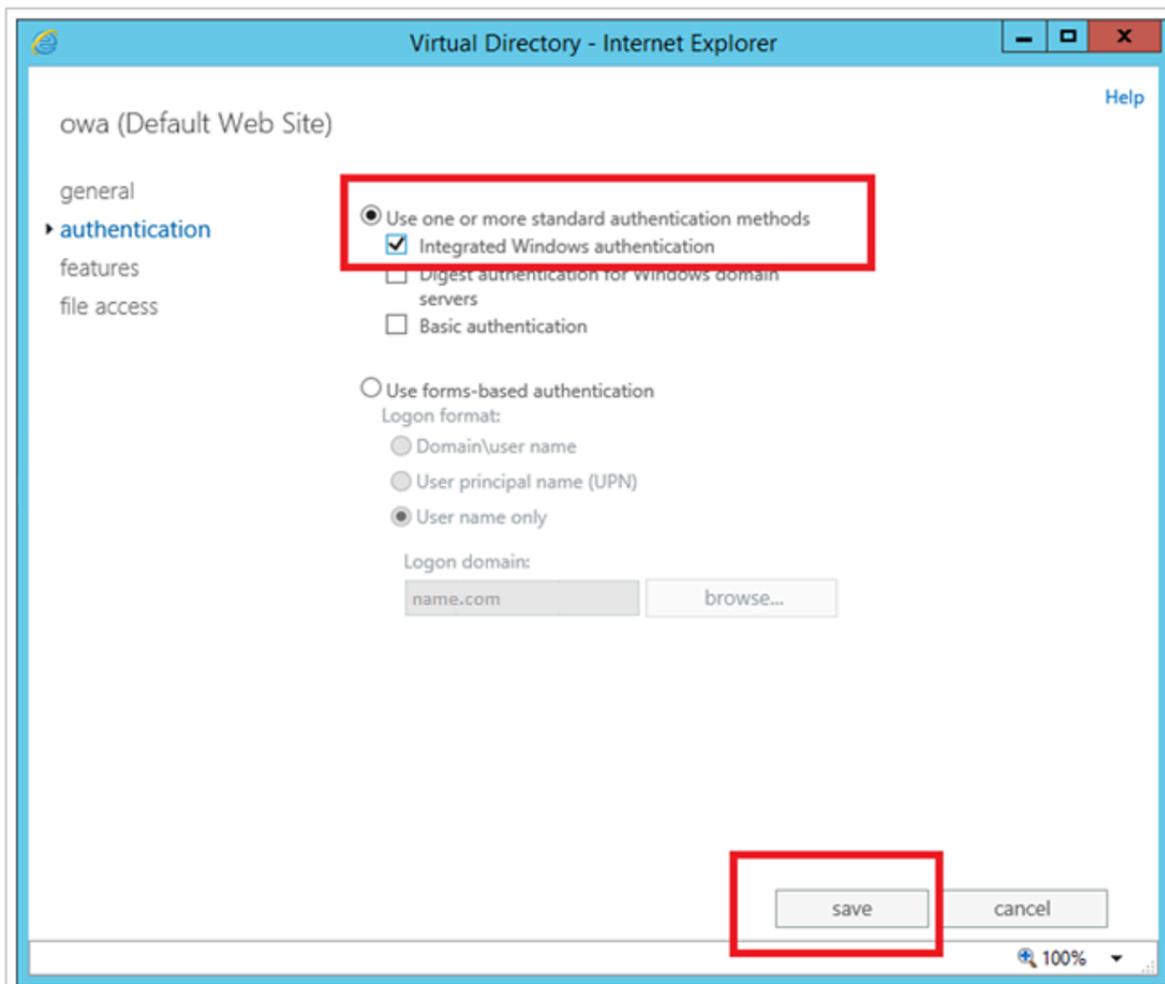
1. Sign in to the [Exchange admin center](#).
2. In the Exchange admin center, left navigation, select **servers**.
3. Select the **virtual directories** tab.



4. From the **select server** dropdown, select a server.
5. Double-click **owa (Default Web Site)**.
6. In the **Virtual Directory**, select the **authentication** tab.



7. On the authentication tab, select **Use one or more standard authentication methods**, and then select **Integrated Windows authentication**.
8. Select **save**



9. Open a command prompt.

10. Execute the **iisreset** command.

```
Administrator: Command Prom...
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\jack>iisreset

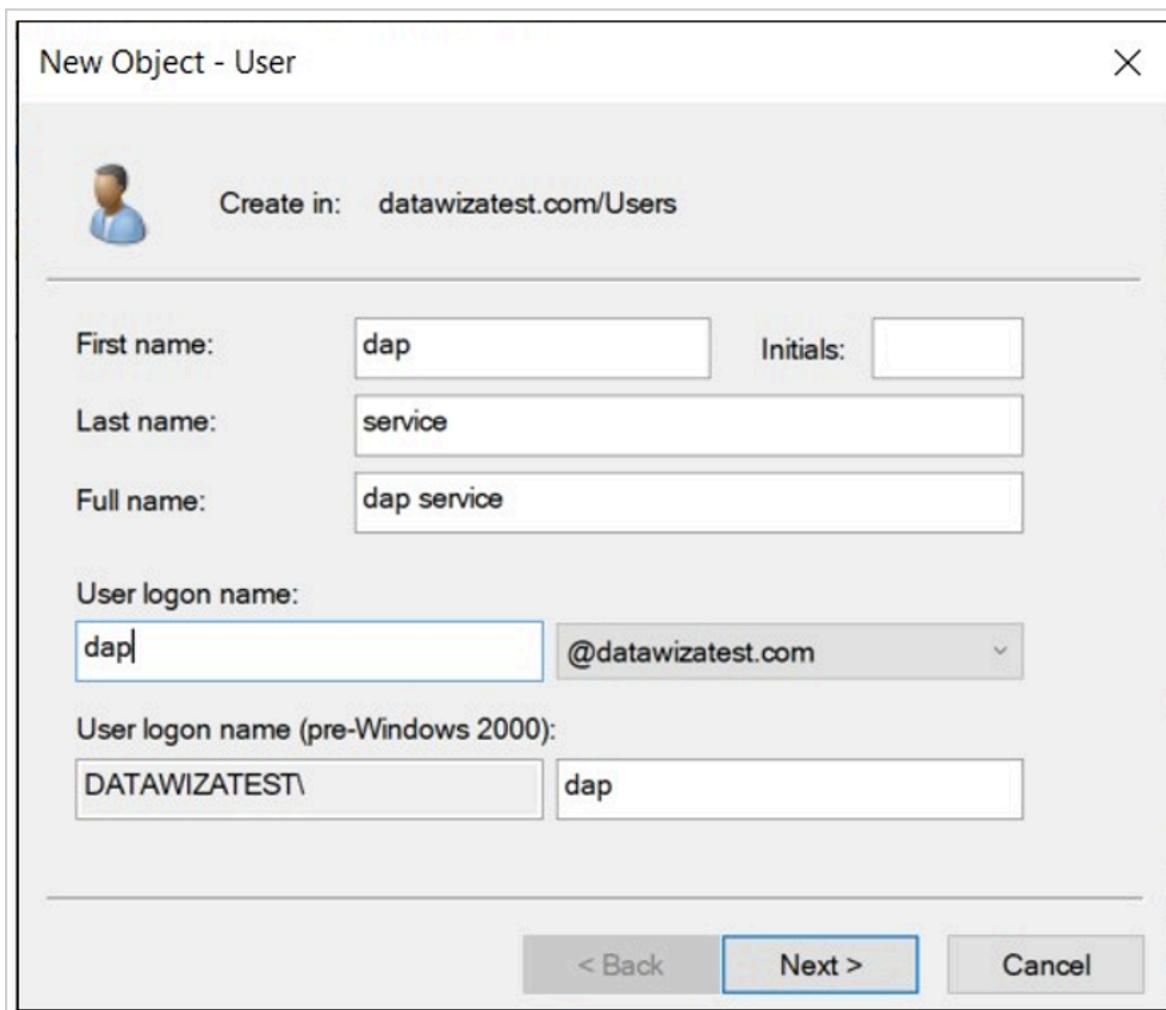
Attempting stop...
Internet services successfully stopped
Attempting start...
Internet services successfully restarted

C:\Users\jack>_
```

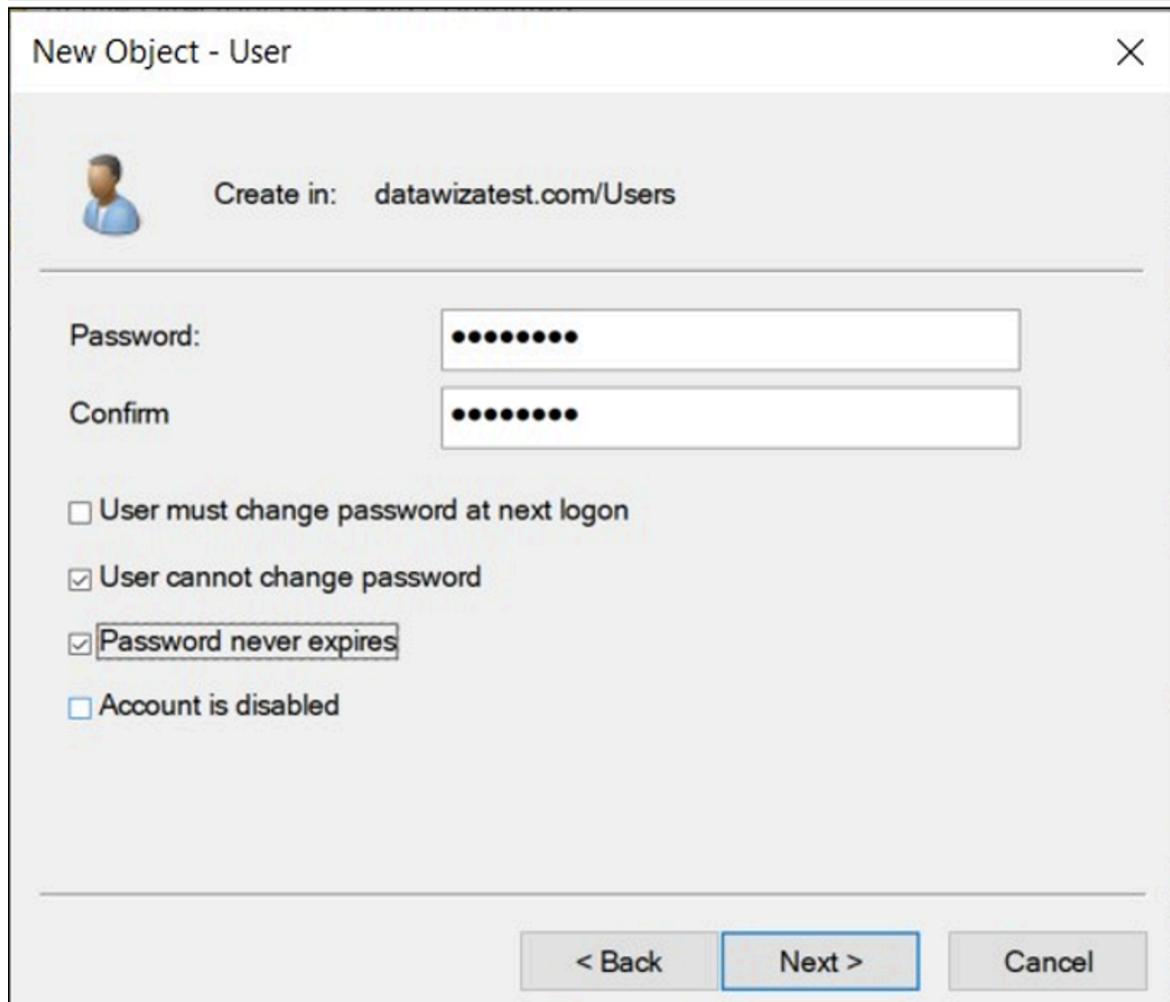
Create a DAP service account

DAP requires known Windows credentials that are used by the instance to configure the Kerberos service. The user is the DAP service account.

1. Sign in to the Windows Server instance.
2. Select **Users and Computers**.
3. Select the DAP instance down-arrow. The example is **datawizatest.com**.
4. In the list, right-click **Users**.
5. From the menu, select **New**, then select **User**.
6. On **New Object--User**, enter a **First name** and **Last name**.
7. For **User logon name**, enter **dap**.
8. Select **Next**.



9. In **Password**, enter a password.
10. Enter it again in **Confirm**.
11. Check the boxes for **User cannot change password** and **Password never expires**.



12. Select **Next**.

13. Right-click the new user to see the configured properties.

Create a service principal name for the service account

Before you create the service principal name (SPN), you can list SPNs and confirm the http SPN is among them.

1. To list SPNs, use the following syntax on the Windows command line.

```
setspn -Q \*/\<**domain.com**
```

2. Confirm the http SPN is among them.

3. To register the host SPN for the account, use the following syntax on the Windows command line.

```
setspn -A host/dap.datawizatest.com dap
```

 **Note**

host/dap.datawizatest.com is the unique SPN, and dap is the service account you created.

Configure Windows Server IIS for Constrained Delegation

1. Sign in to a domain controller (DC).
2. Select **Users and Computers**.
3. In your organization, locate and select the **Users** object.
4. Locate the service account you created.
5. Right-click the account.
6. From the list, select **Properties**.
7. Select the **Delegation** tab.
8. Select **Trust this user for delegation to specified services only**.
9. Select **Use any authentication protocol**.
10. Select **Add**.

dap service Properties

?

X

Organization	Member Of	Dial-in	Environment	Sessions
Remote control	Remote Desktop Services Profile			COM+
General	Address	Account	Profile	Telephones
Delegation				

Delegation is a security-sensitive operation, which allows services to act on behalf of another user.

- Do not trust this user for delegation
- Trust this user for delegation to any service (Kerberos only)
- Trust this user for delegation to specified services only
 - Use Kerberos only
 - Use any authentication protocol

Services to which this account can present delegated credentials:

Service Type	User or Computer	Port	Service Na

Expanded

Add...

Remove

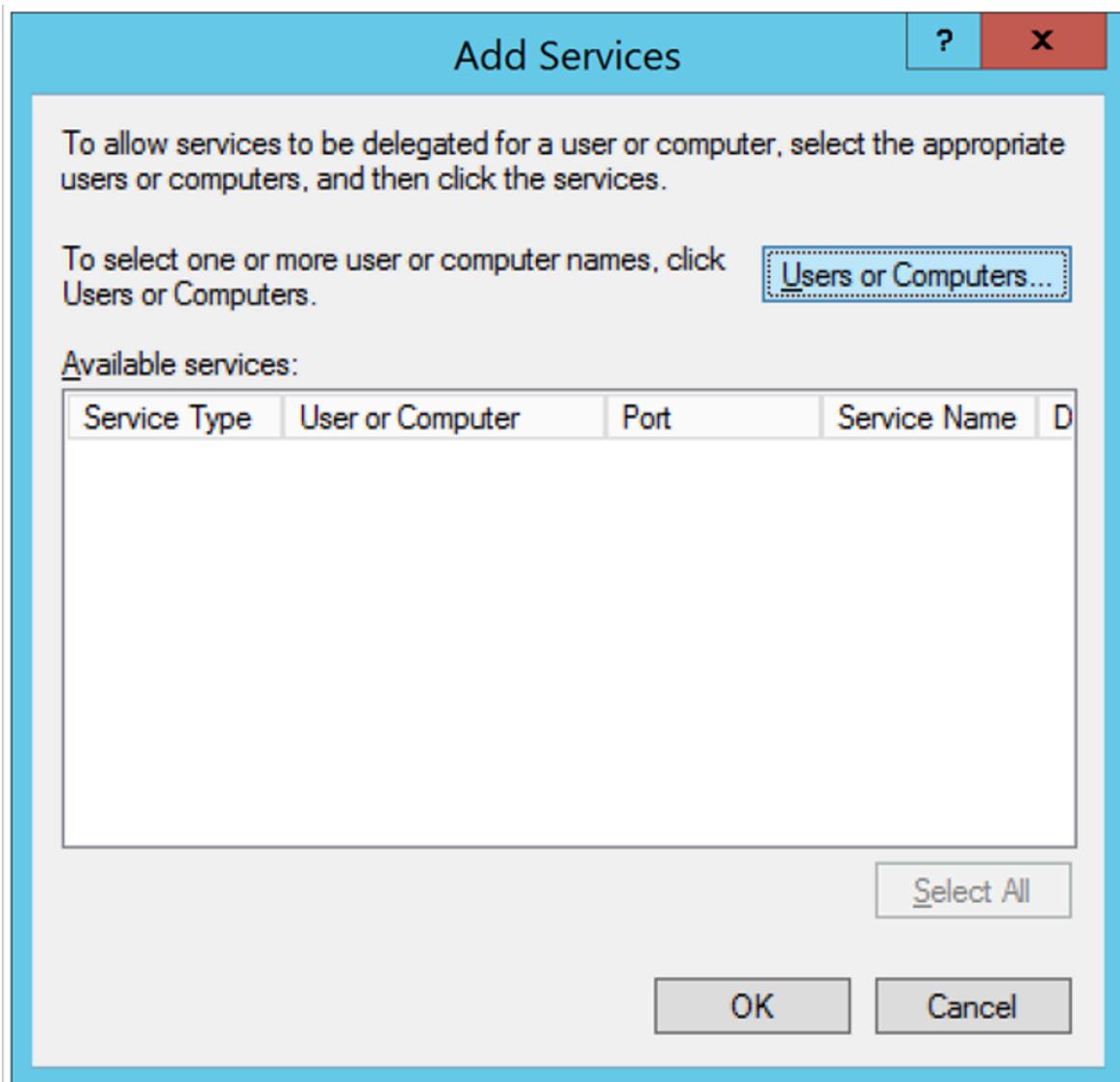
OK

Cancel

Apply

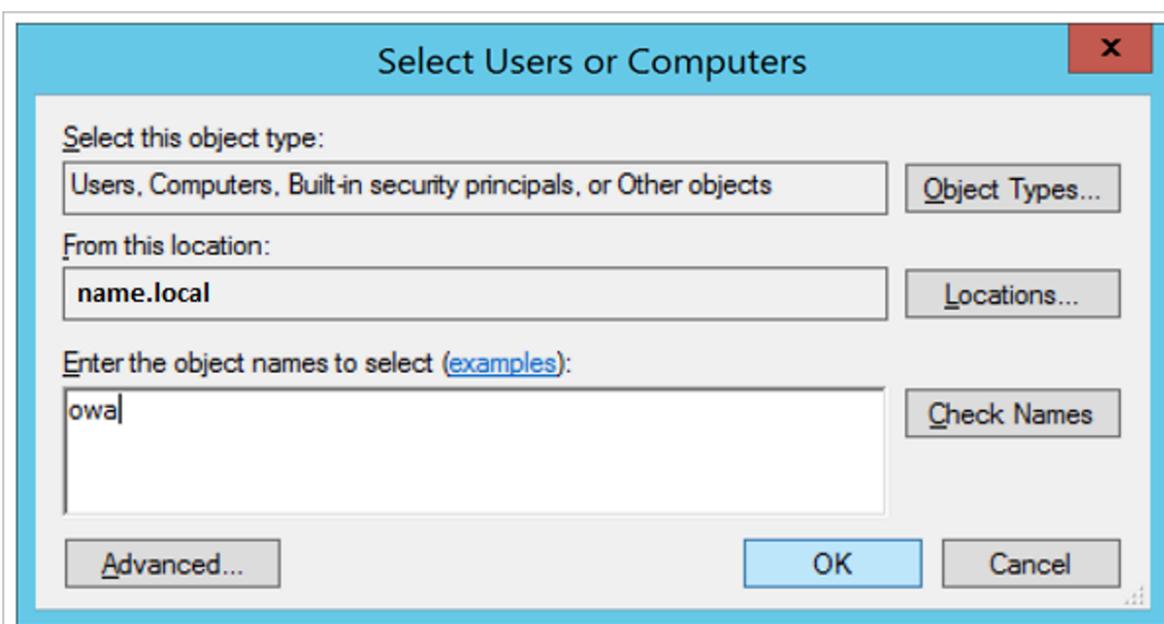
Help

11. On Add Services, select Users or Computers.



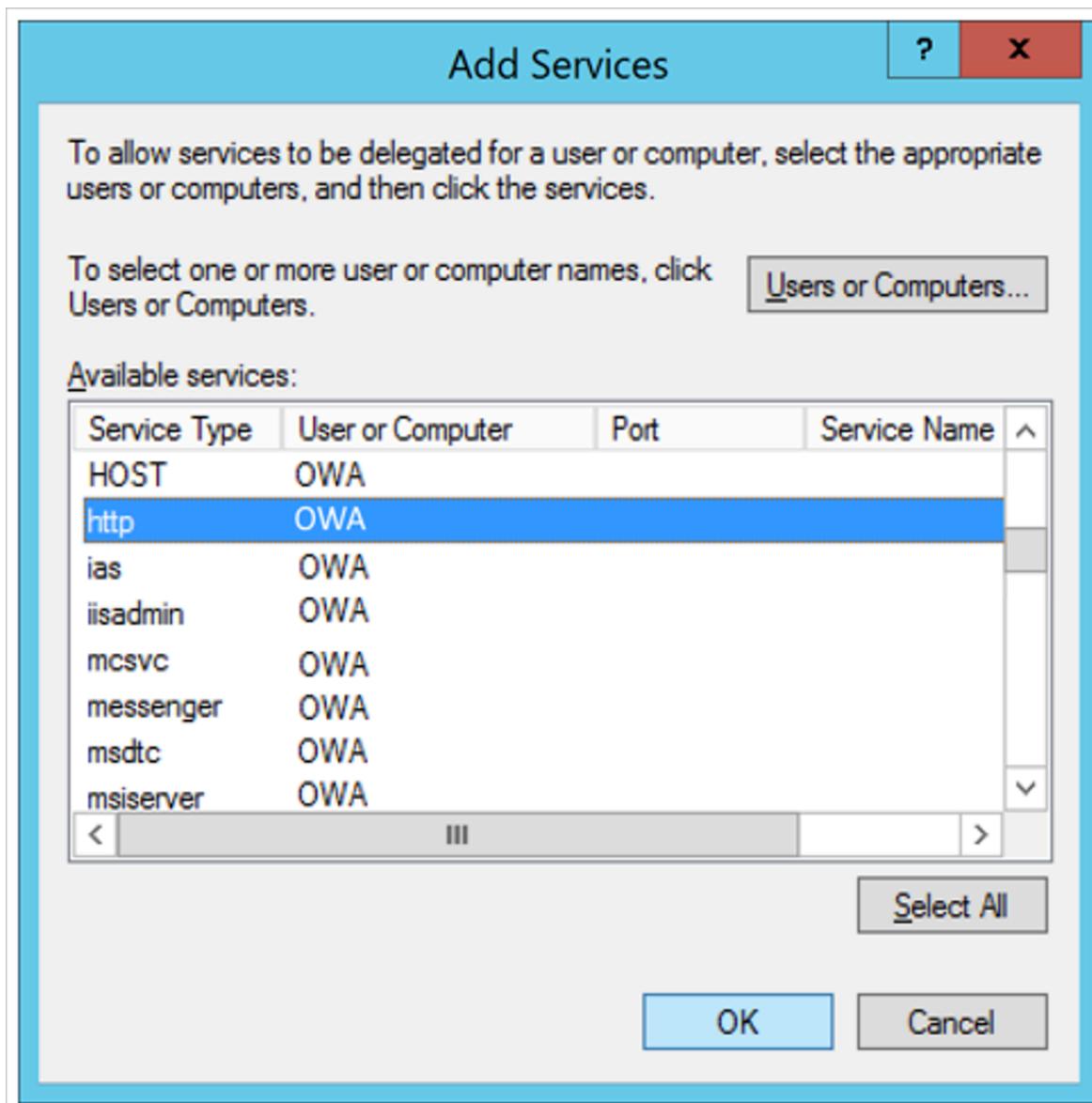
12. In **Enter the object names to select**, type in the machine name.

13. Select OK



14. On Add Services, in Available services, under Service Type, select **http**.

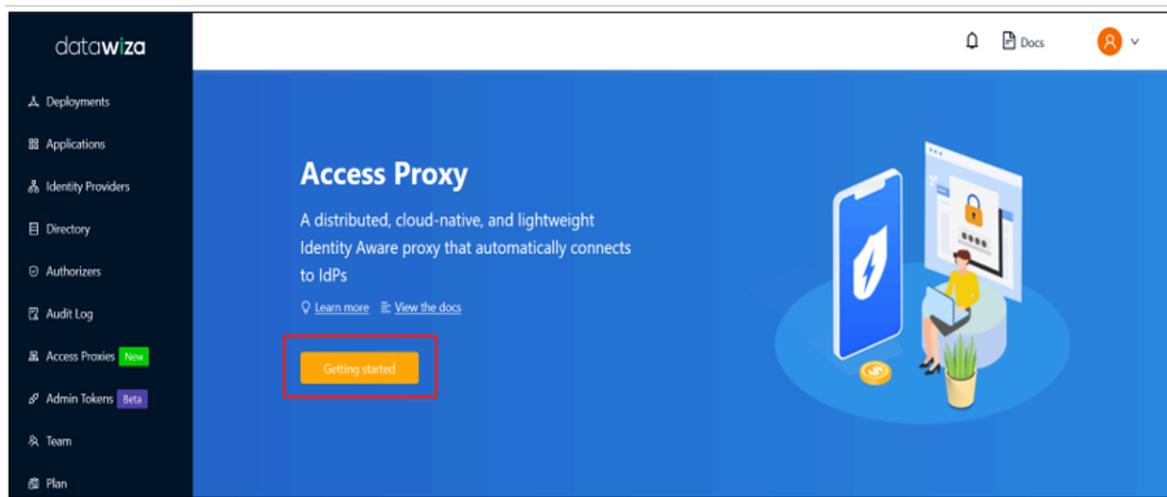
15. Select OK



Integrate OWA with Microsoft Entra ID

Use the following instructions to integrate OWA with Microsoft Entra ID.

1. Sign in to the Datawiza Cloud Management Console (DCMC) [↗](#).
2. The Welcome page appears.
3. Select the orange **Getting started** button.



Deployment Name

1. On Deployment Name, type a Name and a Description.
2. Select Next.

X Create a New Deployment (Step 1 of 4)

Deployment Name

Setting up Deployment is quite easy and simple, you just need to follow the instructions. After several steps, the setup will be finished in a few minutes.

Name *	<input type="text" value="OWA demo"/>
Description	<input type="text" value="Description"/> 0 / 1024

Add Application

1. On Add Application, for Platform, select Web.
2. For App name, enter the app name. We recommend a meaningful naming convention.
3. For Public Domain, enter the app's external-facing URL. For example, <https://external.example.com>. Use localhost domain name server (DNS) for

testing.

4. For **Listen Port**, enter the port DAP listens on. If DAP isn't deployed behind a load balancer, you can use port indicated in Public Domain.
5. For **Upstream Servers**, enter the OWA implementations' URL and port combination.
6. Select **Next**.

X Create a New Deployment (Step 2 of 4)

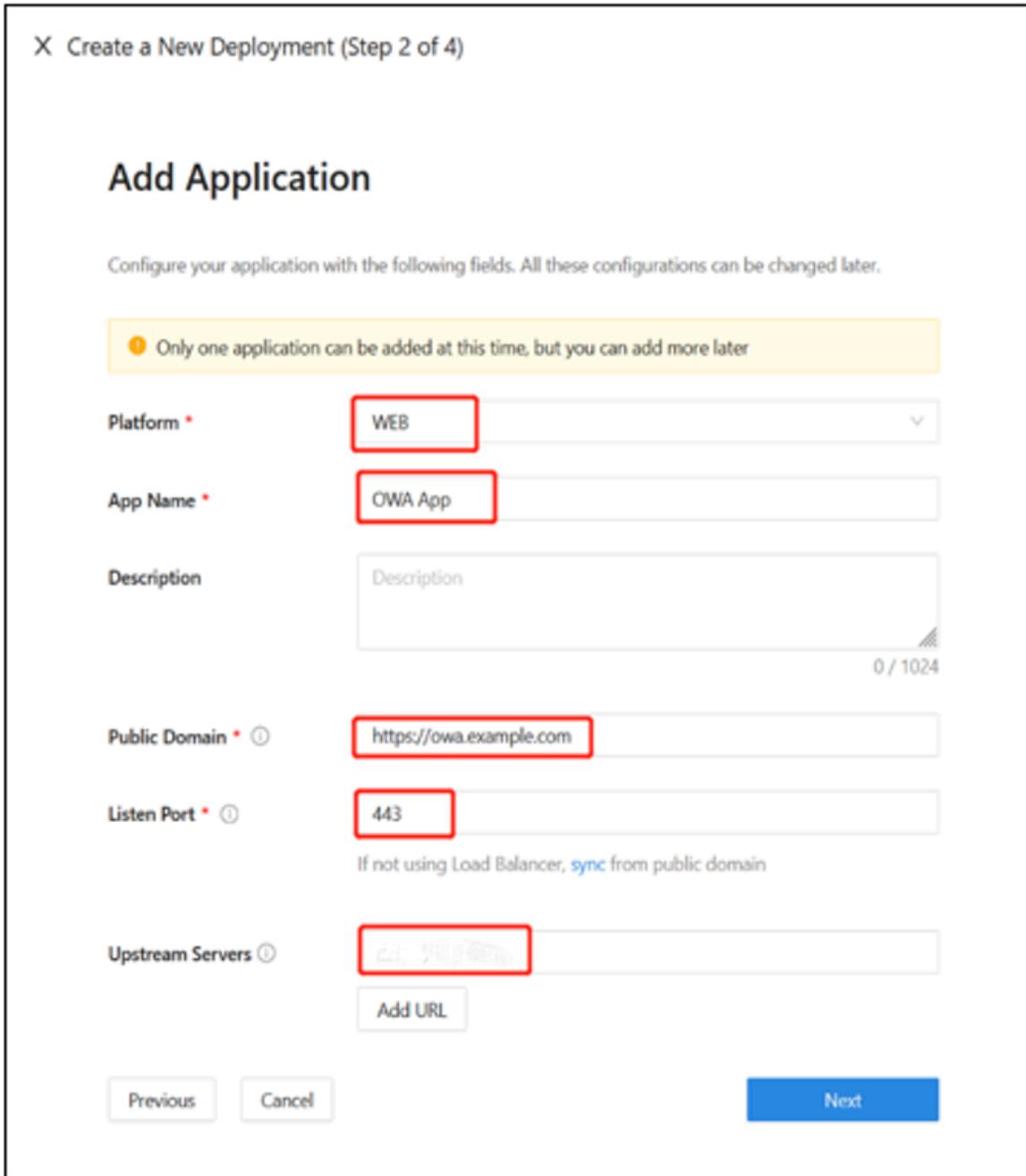
Add Application

Configure your application with the following fields. All these configurations can be changed later.

ⓘ Only one application can be added at this time, but you can add more later

Platform *	WEB
App Name *	OWA App
Description	Description 0 / 1024
Public Domain *	https://owa.example.com
Listen Port *	443 If not using Load Balancer, sync from public domain
Upstream Servers	<input type="text"/> Add URL

Previous Cancel Next



Configure IdP

DCMC integration features help complete Microsoft Entra configuration. Instead, DCMC calls Microsoft Graph API to perform the tasks. The feature reduces time, effort, and errors.

1. On **Configure IdP**, enter a **Name**.
2. For **Protocol**, select **OIDC**.
3. For **Identity Provider**, select **Microsoft Entra ID**.
4. Enable **Automatic Generator**.
5. For **Supported account types**, select **Account in this organizational directory only (Single tenant)**.
6. Select **Create**.
7. A page appears with deployment steps for DAP and the application.
8. See the deployment's Docker Compose file, which includes an image of the DAP, also **PROVISIONING_KEY** and **PROVISIONING_SECRET**. DAP uses the keys to pull the latest DCMC configuration and policies.

Configure Kerberos

1. On your application page, select **Application Detail**.
2. Select the **Advanced** tab.
3. On the **Kerberos** sub tab, enable **Kerberos**.
4. For **Kerberos Realm**, enter the location where the Kerberos database is stored, or the domain.
5. For **SPN**, enter the OWA application's service principal name. It's not the same SPN you created.
6. For **Delegated Login Identity**, enter the applications external-facing URL. Use localhost DNS for testing.
7. For **KDC**, enter a domain controller IP. If DNS is configured, enter a fully qualified domain name (FQDN).
8. For **Service Account**, enter the service account you created.
9. For **Auth Type**, select **Password**.
10. Enter a service account **Password**.
11. Select **Save**.

Access Proxy provides single sign-on (SSO) to applications that use Integrated Windows Authentication (IWA), or claims-aware applications. Choose Integrated Windows Authentication single sign-on mode to provide single sign-on to an on-premises app that authenticates with IWA.

Kerberos

Kerberos Realm *

A realm where kerberos database is stored or the active directory domain

SPN *

The Service Principal Name of the internal network application. This SPN will be used by Access Proxy to provide SSO to your private network application.

Delegated Login Identity *

This enables you to define the delegated identity to be sent for authentication in your on-premises Active Directory when there is a disparity between user login identities.

KDC *

IP of a Domain Controller (Or FQDN if DNS is configured & efficient)

Service Account *

an AD service account with Kerberos constrained delegation

Auth Type *

Password * (S)

The password for the delegation account

Save

SSL configuration

1. On your application page, select the **Advanced** tab.
2. Select the **SSL** subtab.
3. Select **Edit**.

The screenshot shows the Datawiza Access Proxy configuration interface. On the left is a sidebar with various options like Deployments, Applications, Identity Providers, and Audit Log. The main area has tabs for General, IdP Configuration, Profile, Mappings, Attribute Pass, Rules, and Advanced. The Advanced tab is selected and highlighted with a red box. Below it is the 'Advanced Settings' section, which includes tabs for SSL, Advanced Options, JWT, Conditional Access, and Kerberos. The SSL tab is also highlighted with a red box. It shows a toggle switch labeled 'Enable SSL' which is currently set to 'Off'. To the right of the toggle is an 'Edit' button. Below the settings is a diagram showing a 'Browser' connected to a central circle labeled 'datawiza Access Proxy' via a blue line with a lock icon. This circle is then connected to an 'Application' via another blue line with a lock icon.

4. Select the option to **Enable SSL**.
5. From **Cert Type**, select a certificate type. You can use the provided self-signed localhost certificate for testing.

This screenshot shows a detailed view of the SSL configuration. It includes a 'SSL' section with a 'Enable SSL' toggle switch (which is turned on and highlighted with a red box). Below it is a 'Cert Type' dropdown menu (also highlighted with a red box) containing three options: 'Self Signed' (which is selected), 'Self Signed' again, and 'Upload'. There is also a 'Custom Logout URL' input field containing '/ab-logout'. The 'Advanced Options' section is collapsed.

6. Select Save.

Optional: Enable Microsoft Entra multifactor authentication



Steps in this article might vary slightly based on the portal you start from.

To provide more sign-in security, you can enforce Microsoft Entra multifactor authentication. The process starts in the Microsoft Entra admin center.

1. Sign in to the [Microsoft Entra admin center](#) as a [Application Administrator](#).
2. Browse to **Identity > Overview > Properties** tab.
3. Under **Security defaults**, select **Manage security defaults**.

4. On the **Security defaults** pane, toggle the dropdown menu to select **Enabled**.

5. Select **Save**.

Next steps

- Enable SSO and MFA for Oracle JD Edwards with Microsoft Entra ID via Datawiza [↗](#)
 - Configure Secure Hybrid Access with Microsoft Entra ID and Datawiza
 - Go to docs.datawiza.com for [Datawiza user guides](#) [↗](#)
-

Feedback

Was this page helpful?



[Provide product feedback](#) [↗](#)

Deploy F5 BIG-IP Virtual Edition VM in Azure

Article • 11/07/2024

In this tutorial, learn to deploy BIG-IP Vitural Edition (VE) in Azure infrastructure as a service (IaaS). At the end of the tutorial you'll have:

- A prepared BIG-IP virtual machine (VM) to model a secure hybrid access (SHA) proof-of-concept
- A staging instance to test new BIG-IP system updates and hotfixes

Learn more: [SHA: Secure legacy apps with Microsoft Entra ID](#)

Prerequisites

Prior F5 BIG-IP experience or knowledge isn't necessary. However, we recommend you review industry standard terminology in the F5 [Glossary](#).

Deploying a BIG-IP in Azure for SHA requires:

- A paid Azure subscription
 - If you don't have one, you can get an [Azure free trial](#)
- Any of the following F5 BIG-IP license SKUs:
 - F5 BIG-IP® Best bundle
 - F5 BIG-IP Access Policy Manager™ (APM) standalone license
 - F5 BIG-IP Access Policy Manager™ (APM) add-on license on a BIG-IP F5 BIG-IP® Local Traffic Manager™ (LTM)
 - 90-day BIG-IP full feature [trial license](#)
- A wildcard or Subject Alternative Name (SAN) certificate, to publish web applications over Secure Socket Layer (SSL)
 - Go to [letsencrypt.org](#) to see offers. Select [Get Started](#).
- An SSL certificate to secure the BIG-IP management interface. You can use a certificate to publish web apps, if its subject corresponds to the BIG-IP fully qualified domain name (FQDN). For example, you can use a wildcard certificate with a subject `*.contoso.com` for `https://big-ip-vm.contoso.com:8443`.

VM deployment and base system configurations take approximately 30 minutes, then BIG-IP is to implement SHA scenarios in [Integrate F5 BIG-IP with Microsoft Entra ID](#).

Testing scenarios

When you test the scenarios, this tutorial assumes:

- The BIG-IP is deployed into an Azure resource group with an Active Directory (AD) environment
- The environment consists of a Domain Controller (DC) and Internet Information Services (IIS) web host VMs
- Servers not in the same locations as the BIG-IP VM is acceptable, if the BIG-IP sees roles required to support a scenario
- BIG-IP VM connected to another environment, over a VPN connection, is supported

If you don't have the previous items for testing, you can deploy an AD domain environment into Azure, using a script on [Cloud Identity Lab](#). You can programmatically deploy sample test applications to an IIS web host using a scripted automation on [Demo Suite](#).

Note

Some steps in this tutorial might differ from the layout in the Microsoft Entra admin center.

Azure deployment

Tip

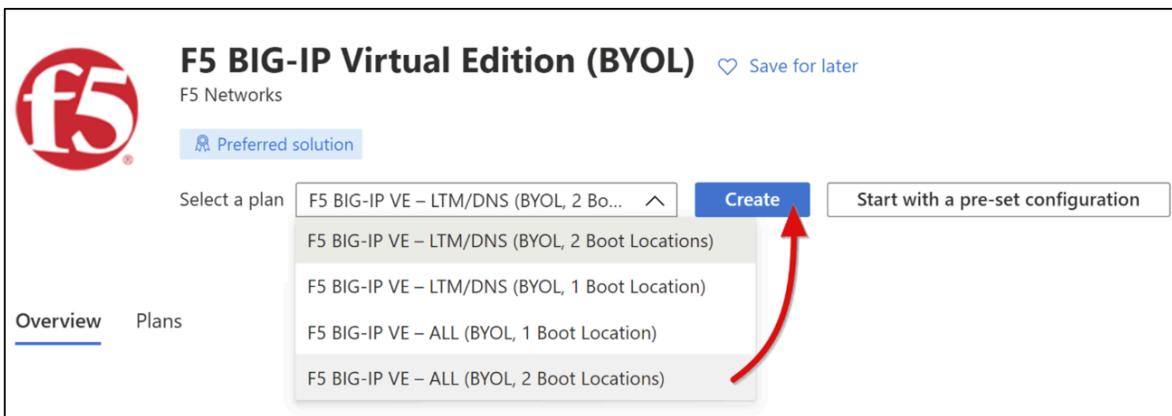
Steps in this article might vary slightly based on the portal you start from.

You can deploy a BIG-IP in different topologies. This guide focuses on a network interface card (NIC) deployment. However, if your BIG-IP deployment requires multiple network interfaces for high availability, network segregation, or more than 1-GB throughput, consider using F5 pre-compiled [Azure Resource Manager \(ARM\) templates](#).

To deploy BIG-IP VE from the [Azure Marketplace](#).

1. Sign in to the [Microsoft Entra admin center](#) with an account with permissions to create VMs, such as an Application Administrator.
2. In the top ribbon search box, type **marketplace**
3. Select **Enter**.

4. Type F5 into the Marketplace filter.
5. Select Enter.
6. From the top ribbon, select + Add.
7. For the marketplace filter, enter F5.
8. Select Enter.
9. Select F5 BIG-IP Virtual Edition (BYOL) > Select a software plan > F5 BIG-IP VE - ALL (BYOL, 2 Boot Locations).
10. Select Create.



11. For Basics:

 - **Subscription:** Target subscription for the BIG-IP VM deployment
 - **Resource group:** The Azure RG the BIG-IP VM will be deployed into, or create one. It's your DC and IIS VMs resource group

12. For Instance details:

- **VM Name** Example BIG-IP-VM
- **Region:** Target Azure geo for BIG-IP-VM
- **Availability options** Enable if using VM in production
- **Image:** F5 BIG-IP VE - ALL (BYOL, 2 Boot Locations)
- **Azure Spot instance:** No, but enable it, if needed
- **Size:** Minimum specifications are 2 vCPUs and 8-GB memory

13. For Administrator account:

- **Authentication type:** Select a password for now, and switch to a key pair later
- **Username:** The identity to be created as a BIG-IP local account to access its management interfaces. Username is CASE sensitive.
- **Password:** Secure admin access with a strong password

14. **Inbound port rules:** Public inbound ports, None.
15. Select **Next: Disks**. Leave the defaults.
16. Select **Next: Networking**.
17. For **Networking**:
 - **Virtual network:** The Azure VNet used by your DC and IIS VMs, or create one
 - **Subnet:** The same Azure internal subnet as your DC and IIS VMs, or create one
 - **Public IP:** None
 - **NIC Network Security Group:** Select None, if the Azure subnet you selected is associated with a network security group (NSG); otherwise select Basic
 - **Accelerate Networking:** Off

18. For **Load balancing**: Load balance VM, No.

19. Select **Next: Management** and complete the settings:

- **Detailed monitoring:** Off
- **Boot diagnostics** Enable with custom storage account. This feature allows connection to the BIG-IP secure shell (SSH) interface via the Serial Console option in the Microsoft Entra admin center. Select an available Azure storage account.

20. For **Identity**:

- **System assigned managed identity:** Off
- **Microsoft Entra ID:** BIG-IP doesn't support this option

21. For **Autoshutdown**: Enable, or if testing, you can set the BIG-IP-VM to shut down daily

22. Select **Next: Advanced**; leave the defaults.

23. Select **Next: Tags**.

24. To review your BIG-IP-VM configuration, select **Next: Review + create**.

25. Select **Create**. Time to deploy a BIG-IP VM typically is 5 minutes.

26. When complete, expand the Microsoft Entra admin center left-hand menu.

27. Select **Resource groups** and navigate to the BIG-IP-VM.

Note

If the VM creation fails, select **Back and Next**.

Network configuration

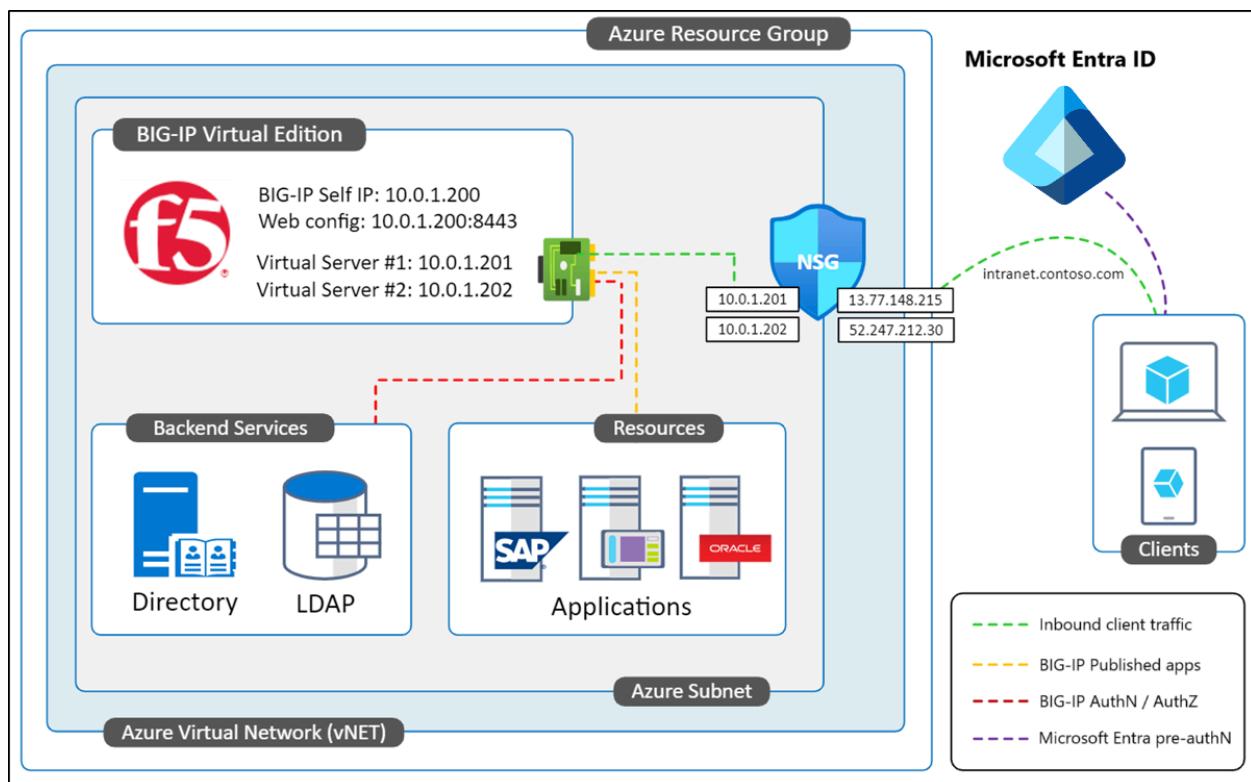
When the BIG-IP VM starts, its NIC is provisioned with a **Primary** private IP issued by the Dynamic Host Configuration Protocol (DHCP) service of the Azure subnet it's connected

to. BIG-IP Traffic Management Operating System (TMOS) uses the IP to communicate with:

- Hosts and services
- Outbound access to the public internet
- Inbound access to the BIG-IP web config and SSH management interfaces

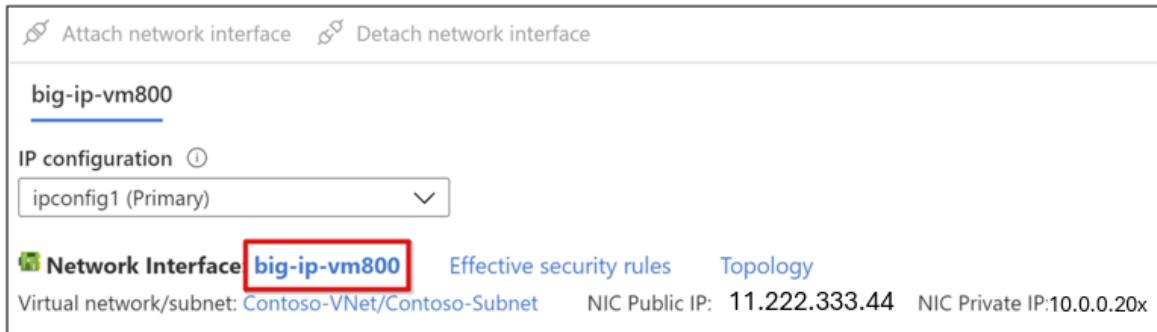
Exposing the management interfaces to the internet increases BIG-IP attack surface. This risk is why the BIG-IP primary IP wasn't provisioned with a public IP during deployment. Instead, a secondary internal IP, and associated public IP, is provisioned for publishing. This one-to-one mapping between a VM public IP, and private IP, enables external traffic to reach a VM. However, an Azure NSG rule is required to allow the traffic, similar to a firewall.

The following diagram shows a NIC deployment of a BIG-IP VE in Azure, configured with a primary IP for general operations and management. There's a separate virtual server IP for publishing services. An NSG rule allows remote traffic destined for `intranet.contoso.com` to route to the public IP for the published service, before being forwarded to the BIG-IP virtual server.



By default, private and public IPs issued to Azure VMs are dynamic, so can change when a VM restarts. Avoid connectivity issues by changing the BIG-IP management IP to static. Do the same action on secondary IPs for publishing services.

1. From your BIG-IP VM menu, go to **Settings > Networking**.
2. In the networking view, select the link to the right of **Network Interface**.



! Note

VM names are generated randomly during deployment.

3. In the left-hand pane, select **IP configurations**.
4. Select the **ipconfig1** row.
5. Set the **IP Assignment** option to **Static**. If necessary, change the BIG-IP VM primary IP address.
6. Select **Save**.
7. Close the **ipconfig1** menu.

! Note

Use the primary IP to connect and manage the BIG-IP-VM.

8. On the top ribbon, select **+ Add**.
9. Provide a secondary private IP name, for example, **ipconfig2**.
10. For the private IP address setting, set the **Allocation** option to **Static**. Providing the next-higher or -lower IP helps preserve orderliness.
11. Set the Public IP address to **Associate**.
12. Select **Create**.
13. For the new public IP address, provide a name, for example, **BIG-IP-VM_ipconfig2_Public**.
14. If prompted, set the **SKU** to **Standard**.
15. If prompted, set the **Tier** to **Global**.
16. Set the **Assignment** option to **Static**.
17. Select **OK** twice.

Your BIG-IP-VM is ready for:

- **Primary private IP:** Dedicated to managing the BIG-IP-VM via its Web config utility and SSH. It's used by the BIG-IP system, as its Self-IP, to connect to published back-end services. It connects to external services:

- Network Time Protocol (NTP)
- Active Directory (AD)
- Lightweight Directory Access Protocol (LDAP)
- **Secondary private IP:** Use to create a BIG-IP APM virtual server to listen for inbound request to a published service(s)
- **Public IP:** It is associated with the secondary private IP; it enables client traffic from the public internet to reach the BIG-IP virtual server for the published service(s)

The example illustrates the one-to-one relationship between a VM public and private IPs. An Azure VM NIC has one primary IP, and other IPs are secondary.

Note

You need the secondary IP mappings for publishing BIG-IP services.

IP configurations					
Subnet *		Contoso-Subnet (10.0.0.0/00)			
<input type="text"/> Search IP configurations					
Name	IP Version	Type	Private IP address	Public IP address	...
ipconfig1	IPv4	Primary	10.0.0.20x	(Static) -	...
ipconfig2	IPv4	Secondary	10.0.0.20y	(Static) 11.22.333.444 (BIG-IP-VM_ipconfig2_Public)	...
ipconfig3	IPv4	Secondary	10.0.0.20z	(Static) 11.222.333.44 (BIG-IP-VM_ipconfig3_Public)	...

To implement SHA using the BIG-IP Access Guided Configuration, repeat steps to create more private and public IP pairs for services you publish via the BIG-IP APM. Use the same approach for publishing services using BIG-IP Advanced Configuration. However, avoid public IP overhead by using a [Server Name Indicator \(SNI\)](#) configuration: a BIG-IP virtual server accepts client traffic it receives, and sends it to its destination.

DNS configuration

To resolve your published SHA services to your BIG-IP-VM public IP(s), configure DNS for clients. The following steps assume the public domain DNS zone for your SHA services is managed in Azure. Apply DNS principles of creating a locator, no matter where your DNS zone is managed.

1. Expand the portal left-hand menu.
2. With the **Resource Groups** option, navigate to your BIG-IP-VM.

3. From the BIG-IP VM menu, go to **Settings** > **Networking**.
4. In the BIG-IP-VMs networking view, from the drop-down IP configuration list, select the first secondary IP.
5. Select the **NIC Public IP** link.

BIG-IP-VM | Networking
Virtual machine

Attach network interface Detach network interface

big-ip-vm800

IP configuration ①

ipconfig2

Filter the ip configurations

	Effective security rules	Topology
ipconfig1 (Primary)	o-Subnet	NIC Public IP: 11.22.333.444 NIC Private IP: 10.0.0.20y
ipconfig2	es	Application security groups Load balancing
ipconfig3		

6. In the left-hand pane, below the **Settings** section, select **Configuration**.
7. The public IP and DNS properties menu appears.
8. Select and **Create** alias record.
9. From the drop-down menu, select your **DNS zone**. If there's no DNS zone, then it can be managed outside Azure, or create one for the domain suffix to verify in Microsoft Entra ID.
10. To create the first DNS alias record:
 - **Subscription:** Same subscription as the BIG-IP-VM
 - **DNS zone:** DNS zone authoritative for the verified domain suffix your published websites use, for example, www.contoso.com
 - **Name:** The hostname you specify resolves to the public IP associated with the selected secondary IP. Define DNS-to IP-mappings. For example, intranet.contoso.com to 11.22.333.444
 - **TTL:** 1
 - **TTL units:** Hours
11. Select **Create**.
12. Leave the **DNS name label (optional)**.

13. Select **Save**.

14. Close the Public IP menu.

ⓘ Note

To create additional DNS records for the services you will publish using BIG-IP Guided Configuration, repeat steps 1 through 6.

With DNS records in place, you can use tools such as [DNS checker](#) to verify a created record propagated across global public DNS servers. If you manage your DNS domain namespace using an external provider like [GoDaddy](#), create records using their DNS management facility.

ⓘ Note

If testing and frequently switching DNS records, you can use a PC local hosts file: Select **Win + R**. In the **Run** box, enter **drivers**. A local host record provides DNS resolution for the local PC, not other clients.

Client traffic

By default, Azure virtual networks (VNets) and associated subnets are private networks unable to receive Internet traffic. Attach your BIG-IP-VM NIC to the NSG specified during deployment. For external web traffic to reach the BIG-IP-VM, define an inbound NSG rule to permit ports 443 (HTTPS) and 80 (HTTP) from the public internet.

1. From the BIG-IP VM main **Overview** menu, select **Networking**.

2. Select **Add** inbound rule.

3. Enter NSG rule properties:

- **Source:** Any
- **Source port ranges:** *|
- **Destination IP addresses:** Comma-separated list of BIG-IP-VM secondary private IPs
- **Destination ports:** 80, 443
- **Protocol:** TCP
- **Action:** Allow
- **Priority:** Lowest available value between 100 and 4096
- **Name:** A descriptive name, for example: `BIG-IP-VM_Web_Services_80_443`

4. Select Add.

5. Close the **Networking** menu.

HTTP and HTTPS traffic can reach your BIG-IP-VMs secondary interfaces. Permitting port 80 allows the BIG-IP APM to auto-redirect users from HTTP to HTTPS. Edit this rule to add or remove destination IPs.

Manage BIG-IP

A BIG-IP system is administered with its web config UI. Access the UI from:

- A machine in the BIG-IP internal network
- A VPN client connected to the BIG-IP-VM internal network
- Published via [Microsoft Entra application proxy](#)

ⓘ Note

Select one of the three previous methods before you proceed with the remaining configurations. If necessary, connect directly to the web config from the internet by configuring the BIG-IP primary IP with a public IP. Then add an NSG rule to allow the 8443 traffic to that primary IP. Restrict the source to your own trusted IP, otherwise anyone can connect.

Confirm connection

Confirm you can connect to the BIG-IP VM web config and sign in with the credentials specified during VM deployment:

- If connecting from a VM on its internal network or via VPN, connect to the BIG-IP primary IP and web config port. For example, `https://<BIG-IP-VM_Primary_IP:8443`. Your browser prompt might state the connection is insecure. Ignore the prompt until the BIG-IP is configured. If the browser blocks access, clear its cache, and try again.
- If you published the web config via Application Proxy, use the URL defined to access the web config externally. Don't append the port, for example, `https://big-ip-vm.contoso.com`. Define the internal URL by using the web config port, for example, `https://big-ip-vm.contoso.com:8443`.

ⓘ Note

You can manage a BIG-IP system with its SSH environment, typically used for command-line (CLI) tasks and root-level access.

To connect to the CLI:

- [Azure Bastion service](#): Connect to VMs in a VNet, from any location
- SSH client, such as PowerShell with the just-in-time (JIT) approach
- Serial Console: In the portal, in the VM menu, Support and troubleshooting section. It doesn't support file transfers.
- From the internet: Configure the BIG-IP primary IP with a public IP. Add an NSG rule to allow SSH traffic. Restrict your trusted IP source.

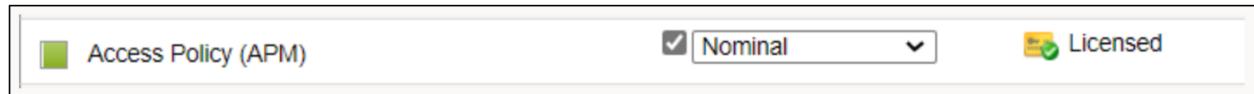
BIG-IP license

Before it can be configured for publishing services and SHA, activate and provision a BIG-IP system with the APM module.

1. Sign in to the web config.
2. On the **General properties** page, select **Activate**.
3. In the **Base Registration key** field, enter the case-sensitive key provided by F5.
4. Leave the **Activation Method** set to **Automatic**.
5. Select **Next**.
6. BIG-IP validates the license and shows the end-user license agreement (EULA).
7. Select **Accept** and wait for activation to complete.
8. Select **Continue**.
9. At the bottom of the License summary page, sign in.
10. Select **Next**.
11. A list of modules, required for SHA, appears.

ⓘ Note

If the list does not appear, in the main tab, go to **System > Resource Provisioning**. Check the provisioning column for Access Policy (APM)



12. Select **Submit**.
13. Accept the warning.
14. Wait for initialization to complete.
15. Select **Continue**.

16. On the **About** tab, select **Run the setup utility**.

ⓘ Important

An F5 license is for one BIG-IP VE instance. To migrate a license from one instance to another, see the AskF5 article, [K41458656: Reusing a BIG-IP VE license on a different BIG-IP VE system](#). Revoke your trial license on the active instance before you decommission it, otherwise the license will be permanently lost.

Provision BIG-IP

It's important to secure management traffic to and from BIG-IP web config. To help protect the web config channel from compromise, configure a device management certificate.

1. From the left-navigation bar, go to **System > Certificate Management > Traffic Certificate Management > SSL Certificate List > Import**.
2. From the **Import Type** drop-down list, select **PKCS 12(IIS)** and **Choose File**.
3. Locate an SSL web certificate with a Subject name or SAN that covers the FQDN, which you assign the BIG-IP-VM later.
4. Provide the certificate password.
5. Select **Import**.
6. From the left-navigation bar, go to **System > Platform**.
7. Under General Properties, enter a qualified **Host Name** and environment **Time Zone**.

General Properties	
Management Config IPV4	<input checked="" type="radio"/> Automatic (DHCP) <input type="radio"/> Manual
Management Config IPV6	<input checked="" type="radio"/> Automatic (DHCP) <input type="radio"/> Manual
Host Name	<input type="text" value="big-ip-vm.contoso.com"/>
Host IP Address	<input type="button" value="Use Management Port IP Address ▾"/>
Time Zone	<input type="button" value="Europe/London ▾"/>

8. Select **Update**.

9. From the left-navigation bar, go to **System > Configuration > Device > NTP**.

10. Specify an NTP source.

11. Select **Add**.

12. Select **Update**. For example, `time.windows.com`

You need a DNS record to resolve the BIG-IPs FQDN to its primary private IP, which you specified in previous steps. Add a record to your environment internal DNS, or to a PC localhost file to connect to the BIG-IP web config. When you connect to the web config, the browser warning no longer appears, not with Application Proxy or any other reverse proxy.

SSL profile

As a reverse proxy, a BIG-IP system is a forwarding service, otherwise known as a Transparent proxy, or a Full proxy that participates in exchanges between clients and servers. A full proxy creates two connections: a front-end TCP client connection and a back-end TCP server connection, with a soft gap in the middle. Clients connect to the proxy listener on one end, a virtual server, and the proxy establishes a separate, independent connection to the back-end server. This configuration is bi-directional on both sides. In this full proxy mode, the F5 BIG-IP system can inspect traffic, and interact with requests and responses. Functions such as load balancing and web performance optimization, and advanced traffic management services (application layer security, web acceleration, page routing, and secure remote access) rely on this functionality. When you publish SSL-based services, BIG-IP SSL profiles handle decrypting and encrypting traffic between clients and back-end services.

There are two profile types:

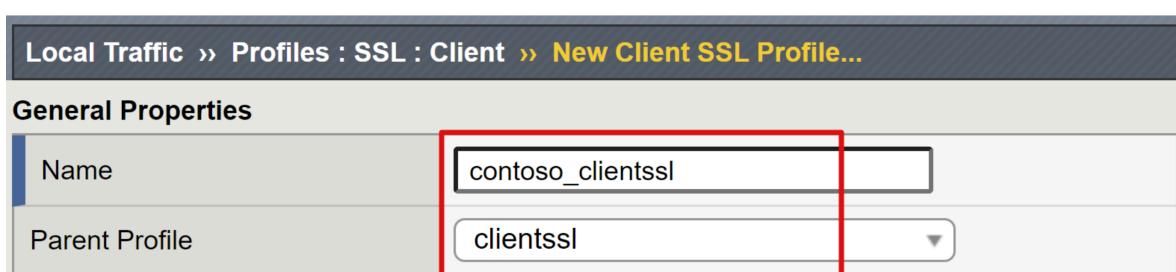
- **Client SSL:** Creating this profile is the most common way to set up a BIG-IP system to publish internal services with SSL. With a Client SSL profile, a BIG-IP system decrypts inbound client requests, before sending them to a down-stream service. It encrypts outbound back-end responses, then sends them to clients.
- **Server SSL:** For back-end services configured for HTTPS, you can configure BIG-IP to use a Server SSL profile. With this profile, the BIG-IP re-encrypts the client request, then sends it to the destination back-end service. When the server returns an encrypted response, the BIG-IP system decrypts and re-encrypts the response, then sends it to the client, through the configured Client SSL profile.

For BIG-IP to be pre-configured and ready for SHA scenarios, provision Client and Server SSL profiles.

1. From the left-navigation, go to **System > Certificate Management > Traffic Certificate Management > SSL Certificate List > Import**.
2. From the **Import Type** drop-down list, select **PKCS 12 (IIS)**.
3. For the imported certificate, enter a name such as **ContosoWildcardCert**.
4. Select **Choose File**.
5. Browse to the SSL web certificate with a Subject name that corresponds to the domain suffix for published services.
6. For the imported certificate, provide the **password**.
7. Select **Import**.
8. From the left-navigation, go to **Local Traffic > Profiles > SSL > Client**.
9. Select **Create**.

10. In the **New Client SSL Profile** page, enter a unique, friendly **Name**.

11. Ensure the Parent profile is set to **clientssl**.



12. In the **Certificate Key Chain** row, select the far-right check box.

13. Select **Add**.

14. From the **Certificate, Key, and Chain** drop-down lists, select the wildcard certificate you imported without a passphrase.

15. Select **Add**.

16. Select **Finished**.

Add SSL Certificate Key Chain

Certificate	ContosoWildcard
Key	ContosoWildcard
Chain	ContosoWildcard
Passphrase	

Add Cancel

17. Repeat steps to create an **SSL server certificate profile**.
18. From the top ribbon, select **SSL > Server > Create**.
19. In the **New Server SSL Profile** page, enter a unique, friendly **Name**.
20. Ensure the Parent profile is set to **serverssl**.
21. Select the far-right check box for the **Certificate** and **Key** rows
22. From the **Certificate** and **Key** drop-down lists, select your imported certificate.
23. Select **Finished**.

Local Traffic » Profiles : SSL : Server » New Server SSL Profile...

General Properties		
Name	contoso_serverssl	
Parent Profile	serverssl	
Configuration:	Basic	Custom <input type="checkbox"/>
Certificate	ContosoWildCard	<input checked="" type="checkbox"/>
Key	ContosoWildCard	<input checked="" type="checkbox"/>

Note

If you're unable to procure an SSL certificate, use the integrated self-signed BIG-IP server and client SSL certificates. A certificate error appears in the browser.

Locate the resource

To prepare a BIG-IP for SHA, locate the resources its publishing, and the directory service it relies on for SSO. A BIG-IP has two sources of name resolution, starting with its local/.../hosts file. If a record isn't found, the BIG-IP system uses the DNS service it was configured with. The hosts file method doesn't apply to APM nodes and pools that use an FQDN.

1. In the web config, go to **System > Configuration > Device > DNS**.
2. In **DNS Lookup Server List**, enter the IP address of your environment DNS server.
3. Select **Add**.
4. Select **Update**.

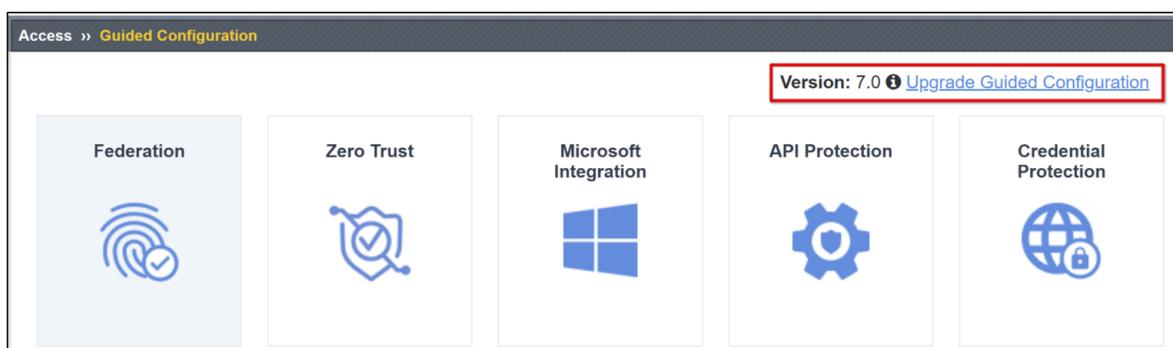
An optional step is an [LDAP configuration](#) to authenticate BIG-IP sysadmins against Active Directory, instead of managing local BIG-IP accounts.

Update BIG-IP

See the following list for update-related guidance. Update instructions follow.

- To check the traffic management operating system (TMOS) version:
 - On the top-left of the main page, hover your cursor over the BIG-IP hostname
- Run v15.x and above. See, [F5 download](#). Sign-in required.
- To update the main TMOS, see the F5 article, [K34745165: Managing software images on the BIG-IP system](#)
 - If you can't update the main TMOS, you can upgrade the Guided Configuration. Use the following steps.
- See also, [scenario-based guidance](#)

1. In the BIG-IP web config, on the main tab, go to **Access > Guided Configuration**.
2. On the **Guided Configuration** page, select **Upgrade Guided Configuration**.



3. On the **Upgrade Guided Configuration** dialog, select **Choose File**.
4. Select **Upload and Install**.
5. Wait for the upgrade to complete.

6. Select **Continue**.

Back up BIG-IP

When the BIG-IP system is provisioned, we recommend a full configuration backup.

1. Go to **System > Archives > Create**.
2. Provide a unique **File Name**.
3. Enable **Encryption** with a passphrase.
4. Set the **Private Keys** option to **Include** to back up device and SSL certificates.
5. Select **Finished**.
6. Wait for the process to complete.
7. A message appears with results.
8. Select **OK**.
9. Select the back-up link.
10. Save the user configuration set (UCS) archive locally.
11. Select **Download**.

You can create a backup of the entire system disk using [Azure snapshots](#). This tool provides contingency for testing between TMOS versions, or rolling back to a fresh system.

```
PowerShell

# Install modules
Install-module Az
Install-module AzureVMSnapshots

# Authenticate to Azure
Connect-azAccount

# Set subscription by Id
Set-AzContext -SubscriptionId '<Azure_Subscription_ID>'

#Create Snapshot
New-AzVmSnapshot -ResourceGroupName '<E.g.contoso-RG>' -VmName '<E.g.BIG-IP-VM>'

#List Snapshots
#Get-AzVmSnapshot -ResourceGroupName '<E.g.contoso-RG>'

#Get-AzVmSnapshot -ResourceGroupName '<E.g.contoso-RG>' -VmName '<E.g.BIG-IP-VM>' | Restore-AzVmSnapshot -RemoveOriginalDisk
```

Restore BIG-IP

Restoring a BIG-IP is similar to the back-up process and can be used to migrate configs between BIG-IP VMs. Before you import a backup, confirm supported upgrade paths.

1. Go to **System > Archives**.

- Select a backup link, or
- Select Upload and browse to a saved UCS archive not in the list

2. Provide the backup passphrase.

3. select **Restore**

PowerShell

```
# Authenticate to Azure
Connect-azAccount

# Set subscription by Id
Set-AzContext -SubscriptionId '<Azure_Subscription_ID>'

#Restore Snapshot
Get-AzVmSnapshot -ResourceGroupName '<E.g. contoso-RG>' -VmName '<E.g. BIG-IP-VM>' | Restore-AzVmSnapshot
```

ⓘ Note

Currently, the AzVmSnapshot cmdlet can restore the most recent snapshot, based on date. Snapshots are stored in the VM resource-group root. Restoring snapshots restarts an Azure VM, therefore ensure optimal timing for the task.

Resources

- [Reset BIG-IP VE password in Azure ↗](#)
- [Reset the password without using the portal ↗](#)
- [Change the NIC used for BIG-IP VE management ↗](#)
- [About routes in a single NIC configuration ↗](#)
- [Microsoft Azure: Waagent ↗](#)

Next steps

Select a [deployment scenario](#) and start your implementation.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Integrate F5 BIG-IP with Microsoft Entra ID

Article • 06/28/2024

With increases in the threat landscape and the use of multiple mobile devices, organizations are rethinking resource access and governance. Part of modernization programs include assessing your readiness across identities, devices, apps, infrastructure, network, and data. You can learn about the [Zero Trust framework to enable remote work ↗](#) and the Zero Trust Assessment tool.

Digital transformation is a long-term journey, and potentially critical resources are exposed until modernized. The goal of F5 BIG-IP and Microsoft Entra ID secure hybrid access (SHA) is to improve remote access to on-premises applications, and strengthen the security posture of vulnerable legacy services.

Research estimates that 60%-80% of on-premises applications are legacy, or incapable of being integrated with Microsoft Entra ID. The same study indicates a large proportion of similar systems run on previous versions of SAP, Oracle, SAGE, and other well-known workloads for critical services.

SHA enables organizations to continue using investments in F5 network and application delivery. With Microsoft Entra ID, SHA bridges the gap with the identity control plane.

Benefits

When Microsoft Entra ID preauthenticates access to BIG-IP published services, there are many benefits:

- Password-less authentication with:
 - [Windows Hello for Business](#)
 - [Microsoft Authenticator ↗](#)
 - [Fast Identity Online \(FIDO\) keys](#)
 - [Certificate-based authentication](#)

Other benefits include:

- One control plane to govern identity and access
 - The [Microsoft Entra admin center ↗](#)
- Preemptive [Conditional Access](#)
- [Microsoft Entra multifactor authentication](#)
- Adaptive protection through user and session risk profiling

- Identity Protection
- Leaked credential detection
- Self-service password reset (SSPR)
- Entitlement management for governed guest access
 - Partner collaboration
- App discovery and control
 - Defender for Cloud Apps
- Threat monitoring and analytics with Microsoft Sentinel ↗

Scenario description

As an Application Delivery Controller (ADC) and secure socket layer virtual private network (SSL-VPN), a BIG-IP system provides local and remote access to services, including:

- Modern and legacy web applications
- Non-web-based applications
- Representational State Transfer (REST) and Simple Object Access Protocol (SOAP)
Web application programming interface (API) services

BIG-IP Local Traffic Manager (LTM) is for secure service publishing, while an Access Policy Manager (APM) extends BIG-IP functions that enable identity federation and single sign-on (SSO).

With integration, you achieve the protocol transition to secure legacy, or other integrated services, with controls such as:

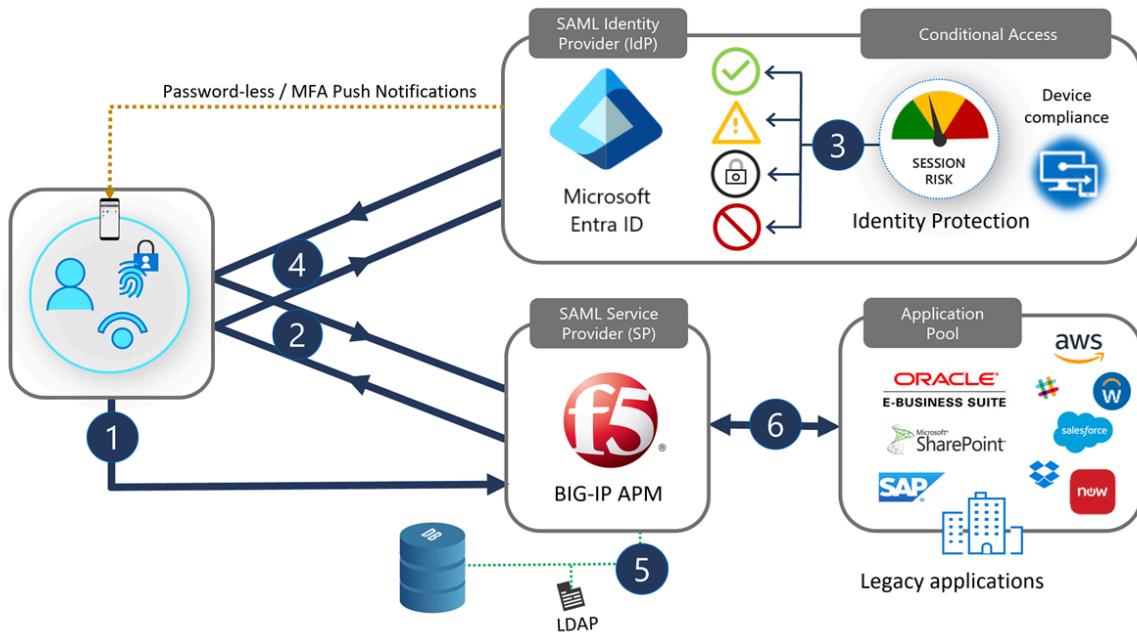
- Passwordless authentication ↗
- Conditional Access

In the scenario, a BIG-IP is a reverse proxy that hands off service preauthentication and authorization to Microsoft Entra ID. The integration is based on a standard federation trust between the APM and Microsoft Entra ID. This scenario is common with SHA. Learn more: [Configure F5 BIG-IP SSL-VPN for Microsoft Entra SSO](#). With SHA you can secure Security Assertion Markup Language (SAML), Open Authorization (OAuth), and OpenID Connect (OIDC) resources.

ⓘ Note

When used for local and remote access, a BIG-IP can be a choke point for Zero Trust access to services, including software as a service (SaaS) apps.

The following diagram illustrates the front-end preauthentication exchange between a user, a BIG-IP, and Microsoft Entra ID, in a service provider (SP) initiated flow. It then shows subsequent APM session enrichment, and SSO to individual back-end services.



1. Users select an application icon in the portal, resolving URL to the SAML SP (BIG-IP)
2. BIG-IP redirects the user to the SAML identity provider (IdP), Microsoft Entra ID, for preauthentication
3. Microsoft Entra ID processes Conditional Access policies and [session controls](#) for authorization
4. Users return to BIG-IP, and present the SAML claims issued by Microsoft Entra ID
5. BIG-IP requests session information for [SSO](#) and [role-based access control \(RBAC\)](#) to the published service
6. BIG-IP forwards the client request to the back-end service

User experience

Whether an employee, affiliate, or consumer, most users are acquainted with the Office 365 sign-in experience. Accessing BIG-IP services is similar.

Users can find their BIG-IP published services in the [My Apps portal](#) or [Microsoft 365 app launcher](#) with self-service capabilities, regardless of device or location. Users can continue accessing published services with the BIG-IP Webtop portal. When users sign out, SHA ensures session termination for BIG-IP and Microsoft Entra ID, helping services remain protected from unauthorized access.

Users access the My Apps portal to find BIG-IP published services and to manage their account properties. See the gallery and self-service page in the following graphics.

The screenshot shows the 'My Apps' section of the BIG-IP interface. At the top, there's a navigation bar with 'WOODGROVE' logo, 'My Apps' dropdown, a search bar ('Search apps...'), and user profile icons. Below the bar, tabs include 'DevOps', 'Workpad' (which is selected), 'Microsoft 365', and 'All apps'. The main area displays a grid of service icons:

- Row 1: Box, Google Cloud, GitHub.com, My Workday, Amazon Web Ser...
- Row 2: Atlassian Cloud, Concur, SAP SuccessFactors, Adobe Creative Cloud, Oracle PeopleSoft
- Row 3: My Expenses, My Travel, VPN, Edge Manage, My Forms
- Row 4: ServiceNow, My Access

The screenshot shows the 'My Account' section of the BIG-IP interface. At the top, there's a navigation bar with 'WOODGROVE' logo, 'My Account' dropdown, and user profile icons. The main area displays a grid of account management options:

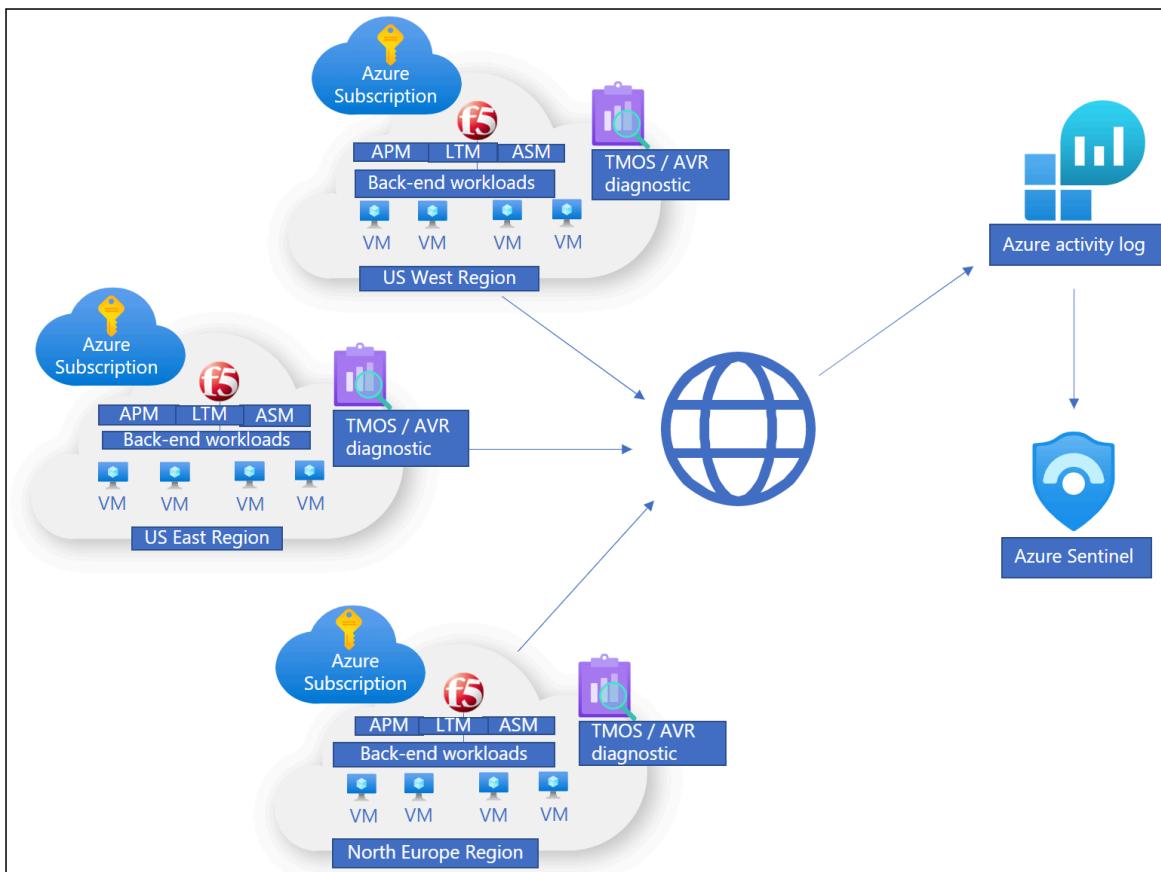
Ed Duck edward.duck@woodgrove.net Why can't I edit?	Security info UPDATE INFO >	Password CHANGE PASSWORD >	Settings & Privacy VIEW SETTINGS AND PRIVACY >
Devices MANAGE DEVICES >	Organizations MANAGE ORGANIZATIONS >	My sign-ins REVIEW RECENT ACTIVITY >	

Insights and analytics

You can monitor deployed BIG-IP instances to ensure published services are highly available, at an SHA level and operationally.

There are several options to log events locally, or remotely through a Security Information and Event Management (SIEM) solution, which enables storage and telemetry processing. To monitor Microsoft Entra ID and SHA activity, you can use [Azure Monitor](#) and [Microsoft Sentinel](#), together:

- Overview of your organization, potentially across multiple clouds, and on-premises locations, including BIG-IP infrastructure
- One control plane with view of signals, avoiding reliance on complex, and disparate tools



Integration prerequisites

No previous experience, or F5 BIG-IP knowledge, is necessary to implement SHA, but we recommend you learn some F5 BIG-IP terminology. See the F5 service [Glossary](#).

Integrating an F5 BIG-IP with Microsoft Entra ID for SHA has the following prerequisites:

- An F5 BIG-IP instance running on:
 - Physical appliance
 - Hypervisor Virtual Edition such as Microsoft Hyper-V, VMware ESXi, Linux kernel-based virtual machine (KVM), and Citrix Hypervisor
 - Cloud Virtual Edition such as Azure, VMware, KVM, Community Xen, MS Hyper-V, AWS, OpenStack, and Google Cloud

Note

The BIG-IP instance location can be on-premises or a supported cloud platform including Azure. The instance has internet connectivity, resources being published, and other services.

- An active F5 BIG-IP APM license:
 - F5 BIG-IP® Best bundle
 - F5 BIG-IP Access Policy Manager™ standalone license
 - F5 BIG-IP Access Policy Manager™ (APM) add-on license on an existing BIG-IP F5 BIG-IP® Local Traffic Manager™ (LTM)
 - A 90-day BIG-IP Access Policy Manager™ (APM) [trial license](#)
- Microsoft Entra ID licensing:
 - An [Azure free account](#) has minimum core requirements for SHA with password-less authentication
 - A [Premium subscription](#) has [Conditional Access](#), [multifactor authentication](#), and [Identity Protection](#)

Configuration scenarios

You can configure a BIG-IP for SHA with template-based options, or a manual configuration. The following tutorials have guidance on implementing BIG-IP and Microsoft Entra ID secure hybrid access.

Advanced configuration

The advanced approach is a flexible way to implement SHA. You manually create all BIG-IP configuration objects. Use this approach for scenarios not in guided configuration templates.

Advanced configuration tutorials:

- [F5 BIG-IP in Azure deployment walk-through](#)
- [F5 BIG-IP SSL-VPN with Microsoft Entra SHA](#)
- [Azure AD B2C protects applications using F5 BIG-IP](#)
- [F5 BIG-IP APM and Microsoft Entra SSO to Kerberos applications](#)
- [F5 BIG-IP APM and Microsoft Entra SSO to header-based applications](#)
- [F5 BIG-IP APM and Microsoft Entra SSO to forms-based applications](#)

Guided Configuration and Easy Button templates

The BIG-IP version 13.1 Guided Configuration wizard, minimizes time and effort to implement common BIG-IP publishing scenarios. Its workflow framework provides an intuitive deployment experience, for specific access topologies.

Guided Configuration version 16.x has the Easy Button feature. Administrators don't back and forth between Microsoft Entra ID and a BIG-IP to enable services for SHA. The APM Guided Configuration wizard and Microsoft Graph handle deployment and policy management. This integration between BIG-IP APM and Microsoft Entra ID ensures applications support identity federation, SSO, and Microsoft Entra Conditional Access, without the management overhead of doing so for each app.

Tutorials for using Easy Button templates, F5 BIG-IP Easy Button for SSO to:

- Kerberos applications
- Header-based applications
- Header-based and Lightweight Directory Access Protocol (LDAP) applications
- Oracle Enterprise Business Suite (EBS)
- Oracle JD Edwards
- Oracle PeopleSoft
- SAP Enterprise Resource Planning (ERP)

Microsoft Entra B2B guest access

Microsoft Entra B2B guest access to SHA-protected applications is possible, but might require steps not in the tutorials. One example is Kerberos SSO, when a BIG-IP performs kerberos constrained delegation (KCD) to obtain a service ticket from domain controllers. Without a local representation of a local guest user, a domain controller doesn't honor the request because there's no user. To support this scenario, ensure external identities are flowed down from your Microsoft Entra tenant to the directory used by the application.

Learn more: [Grant B2B users in Microsoft Entra ID access to your on-premises applications](#)

Next steps

You can conduct a proof-of-concept (POC) for SHA using your BIG-IP infrastructure, or by [Deploying a BIG-IP Virtual Edition virtual machine into Azure](#). It takes approximately 30 minutes to deploy a virtual machine (VM) in Azure. The result is:

- A secured platform to model a pilot for SHA
- A preproduction instance to test new BIG-IP system updates and hotfixes

Identify one or two applications to be published with BIG-IP and protected with SHA.

Our recommendation is to start with an application that isn't published via a BIG-IP. This action avoids potential disruption to production services. The guidelines in this article can help you learn about the procedure to create BIG-IP configuration objects and setting up SHA. You can then convert BIG-IP published services to SHA.

Resources

- [Passwordless strategies ↗](#)
 - [Microsoft Entra ID secure hybrid access ↗](#)
 - [Microsoft Zero Trust framework to enable remote work ↗](#)
 - [Microsoft Sentinel ↗](#)
-

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Tutorial: Configure F5 BIG-IP Easy Button for Kerberos single sign-on

Article • 07/02/2024

Learn to secure Kerberos-based applications with Microsoft Entra ID, through F5 BIG-IP Easy Button Guided Configuration 16.1.

Integrating a BIG-IP with Microsoft Entra ID provides many benefits, including:

- Improved governance: See, [Zero Trust framework to enable remote work](#), and learn more about Microsoft Entra preauthentication.
- Enforce organizational policies. See [What is Conditional Access?](#).
- Full SSO between Microsoft Entra ID and BIG-IP published services
- Manage identities and access from a single control plane, the [Microsoft Entra admin center](#).

To learn more about benefits, see the article on [F5 BIG-IP and Microsoft Entra integration](#).

Scenario description

This scenario is a legacy application using Kerberos authentication, also known as Integrated Windows Authentication (IWA), to gate access to protected content.

Because it's legacy, the application lacks modern protocols to support direct integration with Microsoft Entra ID. You can modernize the application, but it's costly, requires planning, and introduces risk of potential downtime. Instead, an F5 BIG-IP Application Delivery Controller (ADC) bridges the gap between the legacy application and the modern ID control plane, through protocol transitioning.

A BIG-IP in front of the application enables overlay of the service with Microsoft Entra preauthentication and headers-based SSO, improving the security posture of the application.

Note

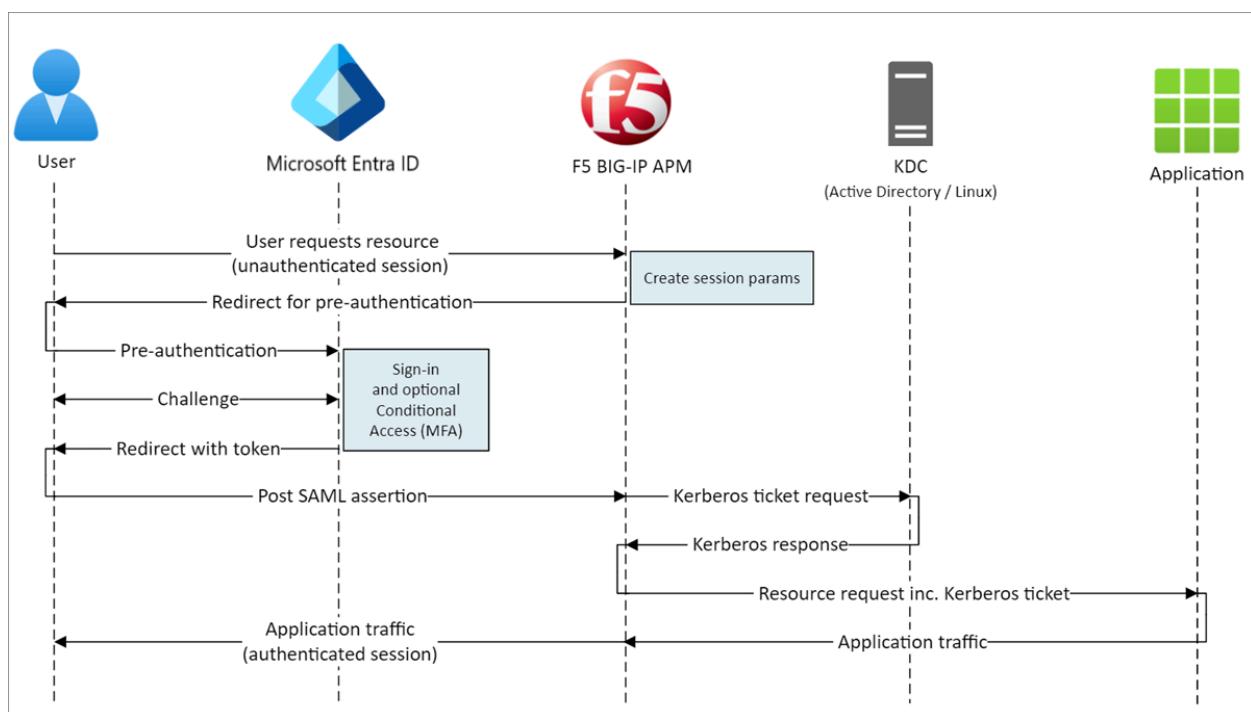
Organizations gain remote access to this type of application with [Microsoft Entra application proxy](#).

Scenario architecture

The secure hybrid access (SHA) solution for this scenario has the following components:

- **Application:** BIG-IP published service to be protected by Microsoft Entra SHA. The application host is domain-joined.
- **Microsoft Entra ID:** Security Assertion Markup Language (SAML) identity provider (IdP) that verifies user credentials, Conditional Access, and SAML-based SSO to the BIG-IP. Through SSO, Microsoft Entra ID provides BIG-IP with required session attributes.
- **KDC:** Key Distribution Center (KDC) role on a Domain Controller (DC), issuing Kerberos tickets
- **BIG-IP:** Reverse proxy and SAML service provider (SP) to the application, delegating authentication to the SAML IdP before performing Kerberos-based SSO to the back-end application.

SHA for this scenario supports SP- and IdP-initiated flows. The following image illustrates the SP flow.



1. User connects to application endpoint (BIG-IP)
2. BIG-IP Access Policy Manager (APM) access policy redirects user to Microsoft Entra ID (SAML IdP)
3. Microsoft Entra ID preauthenticates user and applies any enforced Conditional Access policies
4. User is redirected to BIG-IP (SAML SP) and SSO is performed using issued SAML token
5. BIG-IP requests Kerberos ticket from KDC

6. BIG-IP sends request to backend application, along with Kerberos ticket for SSO

7. Application authorizes request and returns payload

Prerequisites

Prior BIG-IP experience isn't necessary, but you need:

- An [Azure free account ↗](#), or higher
- A BIG-IP or [deploy a BIG-IP Virtual Edition \(VE\) in Azure](#)
- Any of the following F5 BIG-IP licenses:
 - F5 BIG-IP® Best bundle
 - F5 BIG-IP APM standalone
 - F5 BIG-IP APM add-on license on a BIG-IP F5 BIG-IP® Local Traffic Manager™ (LTM)
 - 90-day BIG-IP [Free Trial ↗](#) license
- User identities [synchronized](#) from an on-premises directory to Microsoft Entra ID, or created in Microsoft Entra ID and flowed back to your on-premises directory
- One of the following roles: Cloud Application Administrator, or Application Administrator.
- An [SSL Web certificate](#) for publishing services over HTTPS, or use the default BIG-IP certificates while testing
- A Kerberos application, or learn to configure [SSO with Internet Information Services \(IIS\) on Windows ↗](#).

BIG-IP configuration methods

This tutorial covers Guided Configuration 16.1 with an Easy Button template. With the Easy Button, admins don't go back and forth between Microsoft Entra ID and a BIG-IP to enable services for SHA. APM Guided Configuration wizard and Microsoft Graph handle the deployment and policy management. The integration between BIG-IP APM and Microsoft Entra ID ensures applications support identity federation, SSO, and Microsoft Entra Conditional Access, reducing administrative overhead.

 **Note**

Replace example strings or values in this article with those for your environment.

Register Easy Button

💡 Tip

Steps in this article might vary slightly based on the portal you start from.

Microsoft identity platform trusts a service or client, and then either can access Microsoft Graph. This action creates a tenant app registration to authorize Easy Button access to Graph. Through these permissions, the BIG-IP pushes the configurations to establish a trust between a SAML SP instance for published application, and Microsoft Entra ID as the SAML IdP.

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Identity > Applications > App registrations > New registration**.
3. Enter a display name for your application. For example, F5 BIG-IP Easy Button.
4. Specify who can use the application > **Accounts in this organizational directory only**.
5. Select **Register**.
6. Navigate to **API permissions** and authorize the following Microsoft Graph **Application permissions**:
 - Application.Read.All
 - Application.ReadWrite.All
 - Application.ReadWrite.OwnedBy
 - Directory.Read.All
 - Group.Read.All
 - IdentityRiskyUser.Read.All
 - Policy.Read.All
 - Policy.ReadWrite.ApplicationConfiguration
 - Policy.ReadWrite.ConditionalAccess
 - User.Read.All
7. Grant admin consent for your organization.
8. On **Certificates & Secrets**, generate a new client secret. Make a note of this secret.
9. From **Overview**, note the Client ID and Tenant ID.

Configure Easy Button

Initiate the APM Guided Configuration to launch the Easy Button template.

1. Navigate to **Access > Guided Configuration > Microsoft Integration** and select **Microsoft Entra Application**.
2. Review the configuration steps and select **Next**
3. To publish your application, follow the next steps.



Configuration Properties

The **Configuration Properties** tab creates a BIG-IP application config and SSO object. The **Azure Service Account Details** section can represent the client you registered in your Microsoft Entra tenant earlier, as an application. These settings allow a BIG-IP OAuth client to register a SAML SP in your tenant, with the SSO properties you configure manually. Easy Button does this action for every BIG-IP service published and enabled for SHA.

Some settings are global, which can be reused for publishing more applications, reducing deployment time and effort.

1. Provide a unique **Configuration Name**.
2. Enable **Single Sign-On (SSO) & HTTP Headers**.
3. Enter the **Tenant ID**, **Client ID**, and **Client Secret** you noted when registering the Easy Button client in your tenant.

Configuration Properties

General Properties ▾

Configuration Name

My_Expenses

Type a name for this guided configuration.

Description ⓘ

On

Single Sign-On (SSO) & HTTP Headers ⓘ



Endpoint Checks ⓘ



Additional Checks ⓘ

Azure Service Account Details ▾



Copy Account Info from Existing Configuration ⓘ

Tenant ID ⓘ

5a2bf825-c9cc-488f-9541-22f5602a5a3a

Client ID ⓘ

8b52b0b7-cf0e-4b9f-8f18-0a9ea3300e6e

Client Secret ⓘ

.....

Test Connection



Connection is valid

4. Confirm the BIG-IP connects to your tenant.

5. Select **Next**.

Service Provider

The Service Provider settings are the properties for the SAML SP instance of the application protected through SHA.

1. For **Host**, enter the public fully qualified domain name (FQDN) of the application being secured.
2. For **Entity ID**, enter the identifier Microsoft Entra ID uses to identify the SAML SP requesting a token.

Service Provider

Advanced Settings

Service Provider Properties ▾

Host ⓘ

myexpenses.contoso.com

Entity ID ⓘ

https://myexpenses.contoso.com/My_Expenses

Description ⓘ

(empty)

Rely State ⓘ

(empty)

The optional **Security Settings** specify whether Microsoft Entra ID encrypts issued SAML assertions. Encrypting assertions between Microsoft Entra ID and the BIG-IP APM provides more assurance the content tokens can't be intercepted, and personal or corporate data can't be compromised.

3. From the Assertion Decryption Private Key list, select Create New.

Security Settings ▾

Enable Encrypted Assertion ⓘ

Assertion Decryption Private Key ⓘ

Key Name	Action
--Select--	<input type="button" value="Delete"/>
F5Demo	<input type="button" value="Edit"/>
F5DemoCert	<input type="button" value="Edit"/>

Create New

4. Select OK. The Import SSL Certificate and Keys dialog appears.
5. Select PKCS 12 (IIS) to import your certificate and private key.
6. After provisioning, close the browser tab to return to the main tab.

SSL Certificate/Key Source

Import Type	PKCS 12 (IIS) <input style="width: 20px; height: 15px;" type="button" value="..."/>
Certificate and Key Name	<input checked="" type="radio"/> New <input type="radio"/> Overwrite Existing Contoso_SAML_Cert
Certificate and Key Source	<input type="button" value="Choose File"/> No file chosen
Password
Key Security	Normal <input style="width: 20px; height: 15px;" type="button" value="..."/>
Free Space on Disk	8022 MB

7. Check **Enable Encrypted Assertion**.
8. If you enabled encryption, select your certificate from the **Assertion Decryption Private Key** list. This private key is for the certificate that BIG-IP APM uses to decrypt Microsoft Entra assertions.
9. If you enabled encryption, select your certificate from the **Assertion Decryption Certificate** list. BIG-IP uploads this certificate to Microsoft Entra ID to encrypt the issued SAML assertions.

Security Settings ▾

Enable Encrypted Assertion i

Assertion Decryption Private Key i

Assertion Decryption Certificate i

Microsoft Entra ID

This section defines properties to manually configure a new BIG-IP SAML application in your Microsoft Entra tenant. Easy Button has application templates for Oracle PeopleSoft, Oracle E-business Suite, Oracle JD Edwards, SAP Enterprise Resource Planning (ERP), and an SHA template for other apps.

For this scenario, select **F5 BIG-IP APM Microsoft Entra ID Integration > Add**.

Azure Configuration

1. Enter a **Display Name** for the app that BIG-IP creates in your Microsoft Entra tenant, and the icon in [MyApps portal](#).
2. Leave the **Sign On URL** (optional) blank to enable IdP initiated sign-on.
3. Select the refresh icon next to the **Signing Key** and **Signing Certificate** to locate the certificate you imported.
4. In **Signing Key Passphrase**, enter the certificate password.
5. Enable **Signing Option** (optional) to ensure BIG-IP accepts tokens and claims signed by Microsoft Entra ID.

SAML Signing Certificate

Signing Key	Contoso_Wildcard_Cert	<input type="button" value=""/>
Signing Certificate	Contoso_Wildcard_Cert	<input type="button" value=""/>
Signing Key Passphrase	
Signing Option	Sign SAML assertion	
Signing Algorithm	RSA-SHA256	

6. **User and User Groups** are dynamically queried from your Microsoft Entra tenant and authorize access to the application. Add a user or group for testing, otherwise all access is denied.

User And User Groups

<input type="button" value="+ Add"/>	<input type="button" value="Remove"/>	
<input type="checkbox"/> Name	Type	Description
<input type="checkbox"/> Contoso_Personnel	User Group	Contoso full time employees

User Attributes & Claims

When a user authenticates to Microsoft Entra ID, it issues a SAML token with a default set of claims and attributes identifying the user. The **User Attributes & Claims** tab shows the default claims to issue for the new application. Use it to configure more claims.

The infrastructure is based on a .com domain suffix used internally and externally. More attributes aren't required to achieve a functional Kerberos Constrained Delegation single sign-on (KCD SSO) implementation. See the [advanced tutorial](#) for multiple domains or user sign-in using an alternate suffix.

The screenshot shows the 'User Attributes & Claims' tab selected in the Azure Configuration interface. It displays two sections: 'Required Claims' and 'Additional Claims'.

Required Claims:

Claim Name	Value
Unique User Identifier (Name ID)	user.userprincipalname
Identity	user.onpremisesamaccountname

Additional Claims:

Claim Name	Value	Add
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	user.mail	edit delete
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	user.given...	edit delete
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	user.user...	edit delete
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	user.surn...	edit delete

Additional User Attributes

The **Additional User Attributes** tab supports various distributed systems requiring attributes stored in other directories, for session augmentation. Attributes fetched from a Lightweight Directory Access Protocol (LDAP) source can be injected as SSO headers to help control access based on roles, Partner IDs, and so on.

Note

This feature has no correlation to Microsoft Entra ID but is another source of attributes.

Conditional Access Policy

Conditional Access policies are enforced after Microsoft Entra preauthentication to control access based on device, application, location, and risk signals.

The **Available Policies** view shows Conditional Access policies without user-based actions.

The **Selected Policies** view shows policies targeting cloud apps. You can't deselect policies enforced at the tenant level, nor move them to the Available Policies list.

To select a policy to apply to the application being published:

1. From the **Available Policies** list, select a policy.
2. Select the **right arrow** and move it to the **Selected Policies** list.

Selected policies need an **Include** or **Exclude** option checked. If both options are checked, the selected policy isn't enforced.

Name	Include	Exclude	Apps
Block legacy authentication	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
Term of Use	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
Session Controls	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
MFA for all	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All

ⓘ Note

The policy list appears once, after switching to this tab. You can use the **refresh** button to manually force the wizard to query your tenant, but this button appears after the application is deployed.

Virtual Server Properties

A virtual server is a BIG-IP data plane object represented by a virtual IP address listening for client requests to the application. Received traffic is processed and evaluated against the APM profile associated with the virtual server. Traffic is directed according to policy.

1. Enter a **Destination Address**, an available IPv4/IPv6 address the BIG-IP can use to receive client traffic. There's a corresponding record in domain name server (DNS), enabling clients to resolve the external URL of your BIG-IP published application to this IP, instead of the application. Using a test PC localhost DNS is acceptable for testing.
2. For **Service Port** enter 443 for HTTPS.
3. Check **Enable Redirect Port** and then enter **Redirect Port**, which redirects incoming HTTP client traffic to HTTPS.
4. The Client SSL Profile enables the virtual server for HTTPS, so client connections are encrypted over Transport Layer Security (TLS). Select the **Client SSL Profile** you created for prerequisites, or leave the default if you're testing.

Virtual Server Properties

Advanced Settings

General Properties ▾

Virtual Server
 Create New Use Existing

Destination Address ⓘ
172.16.76.27

Service Port ⓘ
443

Enable Redirect Port ⓘ

Redirect Port ⓘ
80

Client SSL Profile ⓘ
 Create new Use Existing

Available
Filter
Common
clientssl
clientssl-insecure-compatible

Selected
Common
Contoso_clientssl

Create Profile in BIG-IP UI

Pool Properties

The **Application Pool** tab shows the services behind a BIG-IP, represented as a pool with application servers.

1. For **Select a Pool**, create a new pool or select one.
2. Choose a **Load Balancing Method**, such as Round Robin.
3. For **Pool Servers** select a server node, or specify an IP and port for the back-end node hosting the header-based application.

Pool Properties

Advanced Settings

Application Pool ▾

Select a Pool

Create New

Select an existing pool or select Create New.

Resources Properties

Load Balancing Method ⓘ

Round Robin

Pool Servers ⓘ

IP Address/Node name	Port	Priority Group	Action
172.16.74.12	443	HTTPS	<input type="button" value="+"/> <input type="button" value="x"/>

The back-end application runs on HTTP port 80. You can switch the port to 443, if your application runs on HTTPS.

Single sign-on and HTTP Headers

Enabling SSO allows users to access BIG-IP published services without having to enter credentials. The Easy Button wizard supports Kerberos, OAuth Bearer, and HTTP authorization headers for SSO. For these instructions, use the Kerberos delegation account you created.

Enable **Kerberos** and **Show Advanced Setting** to enter the following:

- **Username Source:** The preferred username to cache for SSO. You can provide a session variable as the source of the user ID, but `session.saml.last.identity` works better because it holds the Microsoft Entra claim containing the logged in user ID.
- **User Realm Source:** Required if the user domain differs from the BIG-IP Kerberos realm. In that case, the APM session variable contains the logged-in user domain. For example, `session.saml.last.attr.name.domain`

Single Sign-On & HTTP Headers

Advanced Settings On

Single Sign-On ▾

Selected Single Sign-On Type i

Kerberos

▼

SSO Configuration Object i

Create New



Credentials Source ▾

Username Source i

session.saml.last.identity

User Realm Source i

- **KDC:** Domain controller IP, or FQDN if DNS is configured and efficient
- **UPN Support:** Enable this option for the APM to use the universal principal name (UPN) for Kerberos ticketing
- **SPN Pattern:** Use HTTP/%h to inform the APM to use the host header of the client request, and build the service principal name (SPN) for which it's requesting a Kerberos token
- **Send Authorization:** Disable for applications that negotiate authentication instead of receiving the kerberos token in the first request. For example, Tomcat.

SSO Method Configuration ▾

Kerberos Realm ⓘ
CONTOSO.COM

Account Name ⓘ
host/f5-big-ip.contoso.com@CONTOSO.COM

Account Password
.....
The password for the delegation account specified in the previous field.

Confirm Account Password
.....
Re-type the password for the delegation account specified in the previous field.

KDC ⓘ
172.16.76.4

UPN Support
Enable this to allow the User Principal Name to be used for SSO.

SPN Pattern ⓘ
HTTP/%h

Ticket Lifetime ⓘ
600

Send Authorization ⓘ
Always

Session Management

The BIG-IPs session management settings define the conditions under which user sessions terminate or continue, limits for users and IP addresses, and corresponding user info. Refer to the AskF5 article [K18390492: Security | BIG-IP APM operations guide](#) for settings details.

What isn't covered is Single Log Out (SLO) functionality, which ensures sessions between the IdP, the BIG-IP, and the user agent terminate when a user signs out. When the Easy Button instantiates a SAML application in your Microsoft Entra tenant, it populates the sign-out URL with the APM SLO endpoint. An IdP-initiated sign out from the Microsoft Entra My Apps portal terminates the session between the BIG-IP and a client.

The SAML federation metadata for the published application is imported from your tenant, providing the APM with the SAML sign-out endpoint for Microsoft Entra ID. This action ensures an SP-initiated sign out terminates the session between a client and Microsoft Entra ID. The APM needs to know when a user signs out of the application.

If the BIG-IP webtop portal accesses published applications, then APM processes a sign-out to call the Microsoft Entra sign-out endpoint. But consider a scenario when the BIG-IP webtop portal isn't used, then the user can't instruct the APM to sign out. Even if the user signs out of the application, the BIG-IP is oblivious. Therefore, consider SP-initiated sign out to ensure sessions terminate securely. You can add an SLO function to your application Sign-out button, so it redirects your client to the Microsoft Entra SAML, or the BIG-IP sign out endpoint.

The URL for SAML sign-out endpoint for your tenant is found in [App Registrations > Endpoints](#).

If you can't change the app, then consider having the BIG-IP listen for the application sign-out call, and upon detecting the request, it triggers SLO. To learn about BIG-IP iRules, refer to [Oracle PeopleSoft SLO guidance](#). For more information about using BIG-IP iRules, see:

- [K42052145: Configuring automatic session termination \(log out\) based on a URI-referenced file name ↗](#)
- [K12056: Overview of the Log-out URI Include option ↗](#).

Summary

This section is a breakdown of your configurations.

Select **Deploy** to commit settings and verify the application is in the tenant list of Enterprise applications.

KCD configurations

For the BIG-IP APM to perform SSO to the back-end application on behalf of users, configure key distribution center (KCD) in the target domain. Delegating authentication requires you to provision the BIG-IP APM with a domain service account.

Skip this section if your APM service account and delegation are set up. Otherwise, sign in to a domain controller with an administrator account.

For this scenario, the application is hosted on server APP-VM-01 and runs in the context of a service account named `web_svc_account`, not the computer identity. The delegating service account assigned to the APM is F5-BIG-IP.

Create a BIG-IP APM delegation account

The BIG-IP doesn't support group Managed Service Accounts (gMSA), therefore create a standard user account for the APM service account.

1. Enter the following PowerShell command. Replace the **UserPrincipalName** and **SamAccountName** values with your environment values. For better security, use a dedicated SPN that matches the host header of the application.

```
New-ADUser -Name "F5 BIG-IP Delegation Account" UserPrincipalName $HOST_SPN  
SamAccountName "f5-big-ip" -PasswordNeverExpires $true Enabled $true -  
AccountPassword (Read-Host -AsSecureString "Account Password")
```

HOST_SPN = host/f5-big-ip.contoso.com@contoso.com

 **Note**

When the Host is used, any application running on the host will delegate the account whereas when HTTPS is used, it will allow only HTTP protocol-related operations.

2. Create a **Service Principal Name (SPN)** for the APM service account to use during delegation to the web application service account:

```
Set-AdUser -Identity f5-big-ip -ServicePrincipalNames @{ Add="host/f5-big-  
ip.contoso.com" }
```

 **Note**

It's mandatory to include the host/ part in the format of UserPrincipleName (host/name.domain@domain) or ServicePrincipleName (host/name.domain).

3. Before you specify the target SPN, view its SPN configuration. Ensure the SPN shows against the APM service account. The APM service account delegates for the web application:

- Confirm your web application is running in the computer context or a dedicated service account.
- For the Computer context, use the following command to query the account object to see its defined SPNs. Replace <name_of_account> with the account for your environment.

```
Get-ADComputer -identity <name_of_account> -properties  
ServicePrincipalNames | Select-Object -ExpandProperty  
ServicePrincipalNames
```

For example: Get-User -identity f5-big-ip -properties ServicePrincipalNames |
Select-Object -ExpandProperty ServicePrincipalNames

- For the dedicated service account, use the following command to query the account object to see its defined SPNs. Replace <name_of_account> with the account for your environment.

```
Get-User -identity <name_of_account> -properties ServicePrincipalNames |  
Select-Object -ExpandProperty ServicePrincipalNames
```

For example:

```
Get-Computer -identity f5-big-ip -properties ServicePrincipalNames |  
Select-Object -ExpandProperty ServicePrincipalNames
```

4. If the application ran in the machine context, add the SPN to the object of the computer account:

```
Set-Computer -Identity APP-VM-01 -ServicePrincipalNames @{  
Add="http/myexpenses.contoso.com" }
```

With SPNs defined, establish trust for the APM service account delegate to that service. The configuration varies depending on the topology of your BIG-IP instance and application server.

Configure BIG-IP and target application in the same domain

1. Set trust for the APM service account to delegate authentication:

```
Get-User -Identity f5-big-ip | Set-AccountControl -TrustedToAuthForDelegation  
$true
```

2. The APM service account needs to know the target SPN to delegate to. Set the target SPN to the service account running your web application:

```
Set-User -Identity f5-big-ip -Add @{ 'msDS-  
AllowedToDelegateTo'=@('HTTP/myexpenses.contoso.com') }
```

Note

You can complete these tasks with the Users and Computers, Microsoft Management Console (MMC) snap-in, on a domain controller.

BIG-IP and application in different domains

In the Windows Server 2012 version, and higher, cross-domain KCD uses Resource-Based Constrained Delegation (RBCD). The constraints for a service are transferred from the domain administrator to the service administrator. This delegation allows the back-end service administrator to allow or deny SSO. This situation creates a different approach at configuration delegation, which is possible with PowerShell.

You can use the `PrincipalsAllowedToDelegateToAccount` property of the application service account (computer or dedicated service account) to grant delegation from BIG-IP. For this scenario, use the following PowerShell command on a domain controller (Windows Server 2012 R2, or later) in the same domain as the application.

Use an SPN defined against a web application service account. For better security, use a dedicated SPN that matches the host header of the application. For example, because the web application host header in this example is `myexpenses.contoso.com`, add `HTTP/myexpenses.contoso.com` to the application service account object:

```
Set-User -Identity web_svc_account -ServicePrincipalNames @{
Add="http/myexpenses.contoso.com" }
```

For the following commands, note the context.

If the `web_svc_account` service runs in the context of a user account, use these commands:

```
$big-ip= Get-Computer -Identity f5-big-ip -server dc.contoso.com
```

```
`Set-User -Identity web_svc_account -PrincipalsAllowedToDelegateToAccount`
```

```
$big-ip Get-User web_svc_account -Properties PrincipalsAllowedToDelegateToAccount
```

If the `web_svc_account` service runs in the context of a computer account, use these commands:

```
$big-ip= Get-Computer -Identity f5-big-ip -server dc.contoso.com
```

```
Set-Computer -Identity web_svc_account -PrincipalsAllowedToDelegateToAccount
```

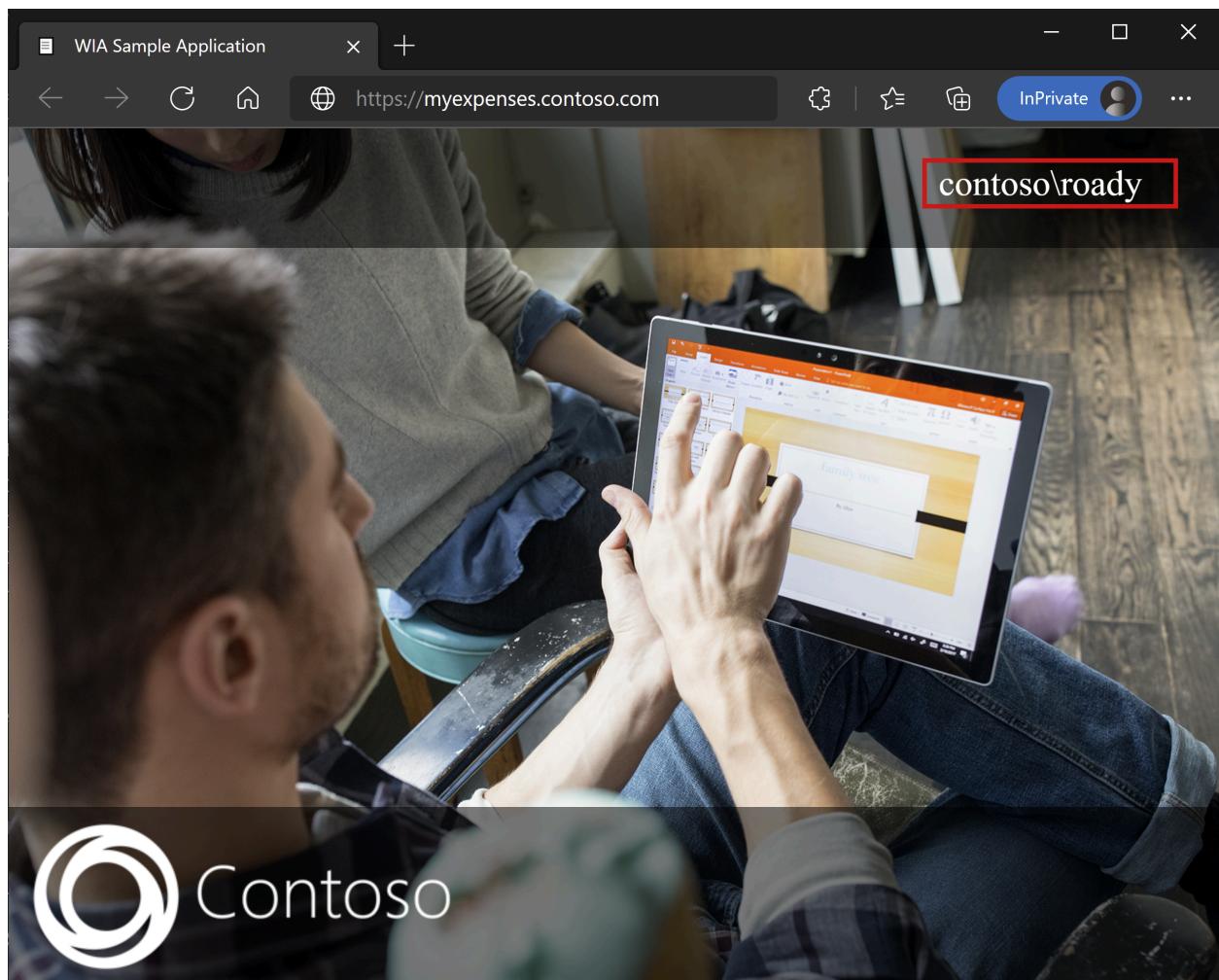
```
$big-ip Get-Computer web_svc_account -Properties
```

```
PrincipalsAllowedToDelegateToAccount
```

For more information, see [Kerberos Constrained Delegation across domains](#).

App view

From a browser, connect to the application external URL or select the **application** icon in the [Microsoft MyApps portal](#). After you authenticate to Microsoft Entra ID, redirection takes you to the BIG-IP virtual server for the application and signed in with SSO.



For increased security, organizations using this pattern can block direct access to the application, which forces a strict path through the BIG-IP.

Microsoft Entra B2B guest access

[Microsoft Entra B2B guest access](#) is supported for this scenario, with guest identities flowing down from your Microsoft Entra tenant to the directory the application uses for authorization. Without a local representation of a guest object in AD, the BIG-IP fails to receive a kerberos ticket for KCD SSO to the back-end application.

Advanced deployment

The Guided Configuration templates can lack the flexibility to achieve some requirements. For those scenarios, see [Advanced Configuration for kerberos-based SSO](#).

Alternatively, in BIG-IP you can disable the Guided Configuration strict management mode. You can manually change your configurations, although the bulk of your configurations are automated through the wizard-based templates.

You can navigate to **Access > Guided Configuration** and select the small **padlock** icon on the far-right of the row for your applications configs.



At this point, changes with the wizard UI aren't possible, but all BIG-IP objects associated with the published instance of the application are unlocked for management.

ⓘ Note

Re-enabling strict mode and deploying a configuration overwrites settings performed outside the Guided Configuration UI. Therefore we recommend the advanced configuration method for production services.

Troubleshooting

If troubleshooting kerberos SSO issues, be aware of the following concepts.

- Kerberos is time sensitive, so it requires servers and clients set to the correct time, and when possible, synchronized to a reliable time source
- Ensure the hostname for the domain controller and web application are resolvable in DNS
- Ensure there are no duplicate SPNs in your AD environment: execute the following query at the command line on a domain PC: setspn -q HTTP/my_target_SPN

You can refer to our [application proxy guidance](#) to validate an IIS application is configured for KCD. See also the AskF5 article, [Kerberos single sign on method ↗](#).

Log analysis: increase verbosity

Use BIG-IP logging to isolate issues with connectivity, SSO, policy violations, or misconfigured variable mappings. Start troubleshooting by increasing the log verbosity level.

1. Navigate to **Access Policy > Overview > Event Logs > Settings**.
2. Select the row for your published application, then **Edit > Access System Logs**.
3. Select **Debug** from the SSO list, and then select **OK**.

Reproduce your issue and inspect the logs. When complete, revert the feature because verbose mode generates much data.

BIG-IP error page

If a BIG-IP error appears after Microsoft Entra preauthentication, the issue might relate to SSO from Microsoft Entra ID to the BIG-IP.

1. Navigate to **Access > Overview > Access reports**.
2. To see logs for clues, run the report for the last hour.
3. Use the **View session variables** link to help understand if the APM receives the expected claims from Microsoft Entra ID.

Back-end request

If no error page appears, the issue is probably related to the back-end request, or SSO from the BIG-IP to the application.

1. Navigate to **Access Policy > Overview > Active Sessions**.
2. Select the link for your active session. The **View Variables** link in this location can help determine root cause KCD issues, particularly if the BIG-IP APM fails to obtain the right user and domain identifiers from session variables.

For more information, see:

- dev/central: [APM variable assign examples](#)
- MyF5: [Session Variables](#)

Feedback

Was this page helpful?



Yes



No

[Provide product feedback](#)

Tutorial: Configure F5 BIG-IP Easy Button for header-based SSO

Article • 04/18/2024

Learn to secure header-based applications with Microsoft Entra ID, with F5 BIG-IP Easy Button Guided Configuration v16.1.

Integrating a BIG-IP with Microsoft Entra ID provides many benefits, including:

- Improved Zero Trust governance through Microsoft Entra preauthentication and Conditional Access
 - See, [What is Conditional Access?](#)
 - See, [Zero Trust security](#)
- Full SSO between Microsoft Entra ID and BIG-IP published services
- Managed identities and access from one control plane
 - See, the [Microsoft Entra admin center](#)

Learn more:

- [Integrate F5 BIG-IP with Microsoft Entra ID](#)
- [Enable SSO for an enterprise application](#)

Scenario description

This scenario covers the legacy application using HTTP authorization headers to manage access to protected content. Legacy lacks modern protocols to support direct integration with Microsoft Entra ID. Modernization is costly, time consuming, and introduces downtime risk. Instead, use an F5 BIG-IP Application Delivery Controller (ADC) to bridge the gap between the legacy application and the modern ID control plane, with protocol transitioning.

A BIG-IP in front of the application enables overlay of the service with Microsoft Entra preauthentication and headers-based SSO. This configuration improves overall application security posture.

Note

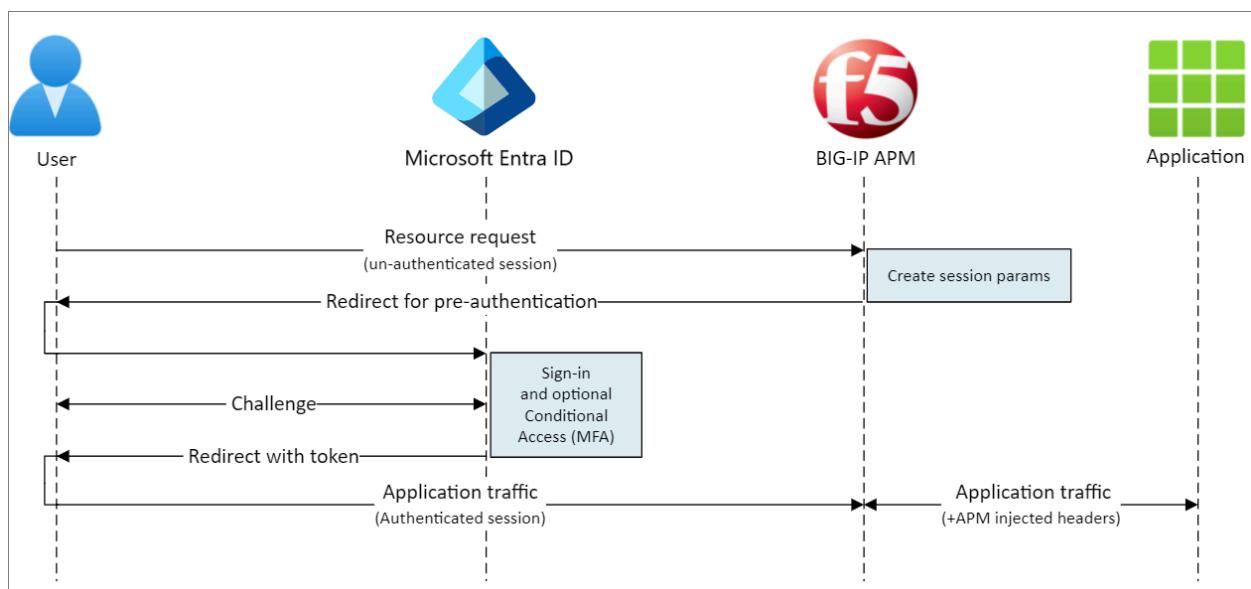
Organizations can have remote access to this application type with Microsoft Entra application proxy. Learn more: [Remote access to on-premises applications through Microsoft Entra application proxy](#)

Scenario architecture

The SHA solution contains:

- **Application** - BIG-IP published service protected by Microsoft Entra SHA
- **Microsoft Entra ID** - Security Assertion Markup Language (SAML) identity provider (IdP) that verifies user credentials, Conditional Access, and SAML-based SSO to the BIG-IP. With SSO, Microsoft Entra ID provides the BIG-IP with session attributes.
- **BIG-IP** - reverse-proxy and SAML service provider (SP) to the application, delegating authentication to the SAML IdP before performing header-based SSO to the backend application.

For this scenario, SHA supports SP- and IdP-initiated flows. The following diagram illustrates the SP-initiated flow.



1. User connects to application endpoint (BIG-IP).
2. BIG-IP APM access policy redirects user to Microsoft Entra ID (SAML IdP).
3. Microsoft Entra preauthenticates user and applies Conditional Access policies.
4. User is redirected to BIG-IP (SAML SP) and SSO occurs using issued SAML token.
5. BIG-IP injects Microsoft Entra attributes as headers in application request.
6. Application authorizes request and returns payload.

Prerequisites

For the scenario you need:

- An Azure subscription
 - If you don't have one, get an [Azure free account](#) ↗

- One of the following roles: Global Administrator, Cloud Application Administrator, or Application Administrator
- A BIG-IP or deploy a BIG-IP Virtual Edition (VE) in Azure
 - See, [Deploy F5 BIG-IP Virtual Edition Virtual Machine in Azure](#)
- Any of the following F5 BIG-IP licenses:
 - F5 BIG-IP® Best bundle
 - F5 BIG-IP Access Policy Manager™ (APM) standalone license
 - F5 BIG-IP Access Policy Manager™ (APM) add-on license on a BIG-IP F5 BIG-IP® Local Traffic Manager™ (LTM)
 - 90-day BIG-IP full feature trial. See, [Free Trials ↗](#)
- User identities synchronized from an on-premises directory to Microsoft Entra ID
 - See, [Microsoft Entra Connect Sync: Understand and customize synchronization](#)
- An SSL web certificate to publish services over HTTPS, or use default BIG-IP certs for testing
 - See, [SSL profile](#)
- A header-based application or set up an IIS header app for testing
 - See, [Set up an IIS header app](#)

BIG-IP configuration

This tutorial uses Guided Configuration v16.1 with an Easy button template. With the Easy Button, admins no longer go back and forth to enable SHA services. The Guided Configuration wizard and Microsoft Graph handle deployment and policy management. The BIG-IP APM and Microsoft Entra integration ensures applications support identity federation, SSO, and Conditional Access.

ⓘ Note

Replace example strings or values with those in your environment.

Register Easy Button

💡 Tip

Steps in this article might vary slightly based on the portal you start from.

Before a client or service accesses Microsoft Graph, the Microsoft identity platform must trust it.

Learn more: [Quickstart: Register an application with the Microsoft identity platform.](#)

Create a tenant app registration to authorize the Easy Button access to Graph. With these permissions, the BIG-IP pushes the configurations to establish a trust between a SAML SP instance for published application, and Microsoft Entra ID as the SAML IdP.

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Identity > Applications > App registrations > New registration**.
3. Under **Manage**, select **App registrations > New registration**.
4. Enter an application **Name**.
5. Specify who uses the application.
6. Select **Accounts in this organizational directory only**.
7. Select **Register**.
8. Navigate to **API permissions**.
9. Authorize the following Microsoft Graph **Application permissions**:
 - Application.Read.All
 - Application.ReadWrite.All
 - Application.ReadWrite.OwnedBy
 - Directory.Read.All
 - Group.Read.All
 - IdentityRiskyUser.Read.All
 - Policy.Read.All
 - Policy.ReadWrite.ApplicationConfiguration
 - Policy.ReadWrite.ConditionalAccess
 - User.Read.All
10. Grant admin consent for your organization.
11. On **Certificates & Secrets**, generate a new **Client Secret**. Make a note of the Client Secret.
12. On **Overview**, note the Client ID and Tenant ID.

Configure Easy Button

1. Start the APM Guided Configuration.
2. Start the **Easy Button** template.
3. Navigate to **Access > Guided Configuration**.
4. Select **Microsoft Integration**
5. Select **Microsoft Entra Application**.
6. Review the configuration steps.
7. Select **Next**.
8. Use the illustrated steps sequence to publish your application.



Configuration Properties

Use the **Configuration Properties** tab to create a BIG-IP application config and SSO object. Azure Service Account Details represent the client you registered in the Microsoft Entra tenant. Use the settings for BIG-IP OAuth client to register a SAML SP in your tenant, with SSO properties. Easy Button performs this action for BIG-IP services published and enabled for SHA.

You can reuse settings to publish more applications.

1. Enter a **Configuration Name**.
2. For **Single Sign-On (SSO) & HTTP Headers**, select **On**.
3. For **Tenant ID**, **Client ID**, and **Client Secret**, enter what you noted.
4. Confirm the BIG-IP connects to your tenant.
5. Select **Next**

Configuration Properties

General Properties ▾

Configuration Name

Type a name for this guided configuration.

Description ⓘ

 On

Single Sign-On (SSO) & HTTP Headers ⓘ



Endpoint Checks ⓘ



Additional Checks ⓘ

Azure Service Account Details ▾



Copy Account Info from Existing Configuration ⓘ

Tenant ID ⓘ

Client ID ⓘ

Client Secret ⓘ

 Connection is valid

Application Settings ▾



Use an existing Azure application ⓘ

Service Provider

In Service Provider settings, define SAML SP instance settings for the SHA-protected application.

1. Enter a **Host**, the application public FQDN.

2. Enter an **Entity ID**, the identifier Microsoft Entra ID uses to identify the SAML SP requesting a token.

Service Provider

Advanced Settings

Service Provider Properties ▾

Host ⓘ	myevents.wacketywack.com
Entity ID ⓘ	https://myevents.wacketywack.com/My_Events
Description ⓘ	
Relay State ⓘ	

3. (Optional) In Security Settings, select **Enable Encryption Assertion** to enable Microsoft Entra ID to encrypt issued SAML assertions. Microsoft Entra ID and BIG-IP APM encryption assertions help assure content tokens aren't intercepted, nor personal or corporate data compromised.

4. In **Security Settings**, from the Assertion Decryption Private Key list, select **Create New**.

Security Settings ▾

Enable Encrypted Assertion ⓘ

Assertion Decryption Private Key ⓘ

Create New	<input type="button" value=""/>
--Select--	<input type="button" value=""/>
F5Demo	<input type="button" value=""/>
Key Type: rsa-private	
Last Update Time: Thu, May 6, 2021	
Security Type: normal	
F5DemoCert	<input type="button" value=""/>
Key Type: rsa-private	
Last Update Time: Fri, May 7, 2021	
Security Type: normal	

5. Select **OK**.

6. The **Import SSL Certificate and Keys** dialog appears.

7. For **Import Type**, select **PKCS 12 (IIS)**. This action imports the certificate and private key.

8. For **Certificate and Key Name**, select **New** and enter the input.

9. Enter the **Password**.

10. Select **Import**.

11. Close the browser tab to return to the main tab.

System » Certificate Management : Traffic Certificate Management : SSL Certificate List » Import SSL Certificates and Keys

SSL Certificate/Key Source	
Import Type	PKCS 12 (IIS) ▾
Certificate and Key Name	<input checked="" type="radio"/> New <input type="radio"/> Overwrite Existing Contoso_SAML_Cert
Certificate and Key Source	Choose File No file chosen
Password
Key Security	Normal ▾
Free Space on Disk	8022 MB
<input type="button" value="Cancel"/> <input type="button" value="Import"/>	

12. Check the box for **Enable Encrypted Assertion**.

13. If you enabled encryption, from the **Assertion Decryption Private Key** list, select the certificate. BIG-IP APM uses this certificate private key to decrypt Microsoft Entra assertions.

14. If you enabled encryption, from the **Assertion Decryption Certificate** list, select the certificate. BIG-IP uploads this certificate to Microsoft Entra ID to encrypt the issued SAML assertions.

Security Settings ▾

<input checked="" type="checkbox"/> Enable Encrypted Assertion ⓘ
Assertion Decryption Private Key ⓘ
Contoso_SAML_cert ▾ <input type="button" value=""/>
Assertion Decryption Certificate ⓘ
Contoso_SAML_cert ▾ <input type="button" value=""/>

Microsoft Entra ID

Use the following instructions to configure a new BIG-IP SAML application in your Microsoft Entra tenant. Easy Button has application templates for Oracle PeopleSoft, Oracle E-Business Suite, Oracle JD Edwards, SAP ERP, and a generic SHA template.

1. In **Azure Configuration**, under **Configuration Properties**, select **F5 BIG-IP APM Microsoft Entra ID Integration**.
2. Select **Add**.

Azure Configuration

1. Enter an app **Display Name** BIG-IP creates in the Microsoft Entra tenant. Users see the name, with an icon, on Microsoft [My Apps](#).
2. Skip **Sign On URL (optional)**.
3. Next to **Signing Key** and **Signing Certificate**, select **refresh** to locate the certificate you imported.
4. In **Signing Key Passphrase**, enter the certificate password.
5. (Optional) Enable **Signing Option** to ensure BIG-IP accepts tokens and claims signed by Microsoft Entra ID.

SAML Signing Certificate ▾

Signing Key ⓘ
Contoso_Wildcard_Cert ⟳

Signing Certificate ⓘ
Contoso_Wildcard_Cert ⟳

Signing Key Passphrase ⓘ
.....

Signing Option ⓘ
Sign SAML assertion

Signing Algorithm ⓘ
RSA-SHA256

6. Input for **User And User Groups** is dynamically queried.

(i) **Important**

Add a user or group for testing, otherwise all access is denied. On **User And User Groups**, select + Add.

User And User Groups ▾

Add	Remove	
Name	Type	Description
Contoso_Personnel	User Group	Contoso full time employees

User Attributes & Claims

When a user authenticates, Microsoft Entra ID issues a SAML token with claims and attributes that identify the user. The **User Attributes & Claims** tab has default claims for the application. Use the tab to configure more claims.

Include one more attribute:

1. For **Header Name**, enter **employeeid**.
2. For **Source Attribute**, enter **user.employeeid**.

Azure Configuration **User Attributes & Claims** Additional User Attributes
Conditional Access Policy

Required Claims ▾

Claim Name	Value
Unique User Identifier (Name ID)	user.userprincipalname
Identity	user.onpremisesamaccountname

Additional Claims ▾

Claim Name	Value	Add
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	user.mail	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	user.givenname	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	user.userprincipalname	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	user.surname	
employeeid	user.employeeid	

Additional User Attributes

In the **Additional User Attributes** tab, enable session augmentation. Use this feature for distributed systems such as Oracle, SAP, and other JAVA implementations that require attributes to be stored in other directories. Attributes fetched from a Lightweight Directory Access Protocol (LDAP) source are injected as more SSO headers. This action helps control access based on roles, Partner IDs, etc.

 **Note**

This feature has no correlation to Microsoft Entra ID. It's an attribute source.

Conditional Access Policy

Conditional Access policies control access based on device, application, location, and risk signals.

- In **Available Policies**, find Conditional Access policies with no user actions
- In **Selected Policies**, find cloud app policy
 - You can't deselect these policies or move them to Available Policies because they're enforced at a tenant level

To select a policy to be applied to the application being published:

1. On the **Conditional Access Policy** tab, in the **Available Policies** list, select a policy.
2. Select the right arrow and move it to the **Selected Policies** list.

 **Note**

You can select the **Include** or **Exclude** option for a policy. If both options are selected, the policy is unenforced.

Azure Configuration User Attributes & Claims Additional User Attributes **Conditional Access Policy**

Conditional Access Policy ▾

[View Conditional Access policies in the Azure Portal](#)

Available Policies ⓘ		Selected Policies ⓘ		
Name	Apps	Name	Include	Exclude
F5 Intranet - MCAS	Selected	Block legacy authentication	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Intranet - Custom Control	Selected	Term of Use	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Intranet - Block downloads	Selected	Session Controls	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SPO - Managed endpoints only	Selected	MFA for all	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SharePoint Restricted Content	Selected			
Breakglass	None			
Managed or Compliant endpoints	None			
Restrict office 365 to Registered domains	Selected			
Block all	Selected			
Block Legacy clients (Office, Internet)	Selected			

ⓘ Note

The policy list appears when you select the **Conditional Access Policy** tab. Select **refresh**, and the wizard queries the tenant. Refresh appears after an application is deployed.

Virtual Server Properties

A virtual server is a BIG-IP data plane object, represented by a virtual IP address. The server listens for client requests to the application. Received traffic is processed and evaluated against the APM profile associated with the virtual server. Traffic is directed according to policy.

1. For **Destination Address**, enter an IPv4 or IPv6 address that BIG-IP uses to receive client traffic. Ensure a corresponding record in domain name server (DNS) that enables clients to resolve the external URL, of the BIG-IP published application, to this IP. You can use computer's localhost DNS for testing.
2. For **Service Port**, enter **443**, and select **HTTPS**.
3. Check the box for **Enable Redirect Port**.

4. Enter a value for **Redirect Port**. This option redirects incoming HTTP client traffic to HTTPS.
5. Select the **Client SSL Profile** you created, or leave the default for testing. The Client SSL Profile enables the virtual server for HTTPS, so client connections are encrypted over TLS.

Virtual Server Properties

Advanced Settings

General Properties ▾

Virtual Server

Create New Use Existing

Destination Address i 172.16.76.27

Service Port i 443 HTTPS ▼

Enable Redirect Port i

Redirect Port i 80 HTTP ▼

Client SSL Profile i

Create new Use Existing

Available	Selected
<small>Filter</small>	
Common	Common
clientssl	Contoso_clientssl
clientssl-insecure-compatible	

Create Profile in BIG-IP UI ↻

Pool Properties

The **Application Pool** tab has services behind a BIG-IP, represented as a pool, with one or more application servers.

1. For **Select a Pool**, select **Create New**, or select another.
2. For **Load Balancing Method**, select **Round Robin**.

3. For **Pool Servers**, select a node, or select an IP address and port for the server hosting the header-based application.

The screenshot shows the 'Pool Properties' configuration page. At the top, there is an 'Advanced Settings' toggle switch. Below it, a dropdown menu is set to 'Application Pool'. The main section is titled 'Select a Pool' with a dropdown menu showing 'Create New'. A note below says 'Select an existing pool or select Create New.' The next section is 'Resources Properties' with a 'Load Balancing Method' dropdown set to 'Round Robin'. The final section is 'Pool Servers' with a table:

IP Address/Node name	Port	Priority Group	Action
172.16.74.12	443	HTTPS	0 + x

! Note

The Microsoft back-end application is on HTTP Port 80. If you select HTTPS, use 443.

Single Sign-On & HTTP Headers

With SSO, users access BIG-IP published services without entering credentials. The Easy Button wizard supports Kerberos, OAuth Bearer, and HTTP authorization headers for SSO.

1. On **Single Sign-On & HTTP Headers**, in **SSO Headers**, for **Header Operation**, select **insert**
2. For **Header Name**, use **upn**.
3. For **Header Value**, use **%{session.saml.last.identity}**.
4. For **Header Operation**, select **insert**.
5. For **Header Name**, use **employeeid**.
6. For **Header Value**, use **%{session.saml.last.attr.name.employeeid}**.

Single Sign-On & HTTP Headers

Header Operation	Header Name	Header Value	Delimiter	Action
insert	upn	{session.sso.token.last.username}		+ X
insert	employeeid	{session.saml.last.attr.name.employeeid}		+ X

ⓘ Note

APM session variables in curly brackets are case-sensitive. Inconsistencies cause attribute mapping failures.

Session Management

Use BIG-IP session management settings to define conditions for user sessions termination or continuation.

To learn more, go to support.f5.com for [K18390492: Security | BIG-IP APM operations guide ↗](#)

Single log-out (SLO) ensures IdP, BIG-IP, and user agent sessions terminate when users sign out. When the Easy Button instantiates a SAML application in your Microsoft Entra tenant, it populates the sign out URL, with the APM SLO endpoint. IdP-initiated sign out from My Apps terminates BIG-IP and client sessions.

Learn more: see, [My Apps ↗](#)

The SAML federation metadata for the published application is imported from your tenant. The import provides the APM with the SAML sign out endpoint for Microsoft Entra ID. This action ensures SP-initiated sign out terminates client and Microsoft Entra sessions. Ensure the APM knows when user sign out occurs.

If the BIG-IP webtop portal accesses published applications, then the eAPM processes the sign out to call the Microsoft Entra sign-out endpoint. If the BIG-IP webtop portal isn't used, users can't instruct the APM to sign out. If users sign out of the application, the BIG-IP is oblivious. Thus, ensure SP-initiated sign out securely terminates sessions. You can add an SLO function to an application **Sign out** button, Then, clients are redirected

to the Microsoft Entra SAML or BIG-IP sign out endpoint. To locate the SAML sign out endpoint URL for your tenant, go to **App Registrations > Endpoints**.

If you can't change the app, enable the BIG-IP to listen for the application sign out call and trigger SLO.

Learn more:

- [PeopleSoft Single Logout](#)
- Go to support.f5.com for:
 - [K42052145: Configuring automatic session termination \(logout\) based on a URI-referenced file name ↗](#)
 - [K12056: Overview of the Logout URI Include option ↗](#).

Deploy

Deployment provides a breakdown of your configurations.

1. To commit settings, select **Deploy**.
2. Verify the application in your tenant list of Enterprise applications.
3. The application is published and accessible via SHA, with its URL, or on Microsoft application portals.

Test

1. In a browser, connect to the application external URL or select the application icon on [My Apps ↗](#).
2. Authenticate to Microsoft Entra ID.
3. A redirection occurs to the BIG-IP virtual server for the application and signed in with SSO.

The following screenshot is injected headers output from the header-based application.

The screenshot shows a browser window titled "My Events" with the URL "https://myevents.contoso.com". The page displays "Request Details" and "Server Variables".

Request Details

Session Id:	kiekszqbh3dydnpiogyjbvna	Request Type:	GET
Time of Request:	8/23/2021 1:48:29 PM	Status Code:	200
Request Encoding:	Unicode (UTF-8)	Response Encoding:	Unicode (UTF-8)

Server Variables

Name	Value
REMOTE_ADDR	172.16.76.16
REMOTE_PORT	32656
REQUEST_METHOD	GET
SCRIPT_NAME	/default.aspx
SERVER_PORT	80
SERVER_PROTOCOL	HTTP/1.1
SERVER_SOFTWARE	Microsoft-IIS/8.5
URL	/default.aspx
HTTP_CONNECTION	keep-alive
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,app
HTTP_ACCEPT_ENCODING	gzip, deflate, br
HTTP_ACCEPT_LANGUAGE	en-GB,en;q=0.9
HTTP_COOKIE	TIN=297000; LastMRH_Session=a16276ee; F5_ST=1z1z1z1629046109z604800
HTTP_HOST	myevents.contoso.com
HTTP_REFERER	https://login.microsoftonline.com/
HTTP_USER_AGENT	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/9
HTTP_UPN	ruby.a@contoso.com
HTTP_EMPLOYEEID	564738
HTTP_EVENTROLES	Approver

Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.8.4330.0

ⓘ Note

You can block direct access to the application, thereby enforcing a path through the BIG-IP.

Advanced deployment

For some scenarios, Guided Configuration templates lack flexibility.

Learn more: [Tutorial: Configure F5 BIG-IP Access Policy Manager for header-based SSO](#).

In BIG-IP, you can disable the Guided Configuration strict management mode. Then, manually change configurations, however most configurations are automated with wizard templates.

1. To disable strict mode, navigate to **Access > Guided Configuration**.
2. On the row for the application configuration, select the **padlock** icon.
3. BIG-IP objects associated with the published instance of the application are unlocked for management. Changes with the wizard are no longer possible.



! Note

If you re-enable strict mode and deploy a configuration, the action overwrites settings not in the Guided Configuration. We recommend the advanced configuration for production services.

Troubleshooting

Use the following guidance when troubleshooting.

Log verbosity

BIG-IP logs help isolate issues with connectivity, SSO, policy, or misconfigured variable mappings. To troubleshoot, increase the log verbosity.

1. Navigate to **Access Policy > Overview**.
2. Select **Event Logs**.
3. Select **Settings**.
4. Select the row of your published application
5. Select **Edit**.
6. Select **Access System Logs**.
7. From the SSO list, select **Debug**.
8. Select **OK**.
9. Reproduce the issue.
10. Inspect the logs.

! Note

Revert this feature when finished. Verbose mode generates excessive data.

BIG-IP error message

If a BIG-IP error message appears after Microsoft Entra preauthentication, the issue might relate to Microsoft Entra ID-to-BIG-IP SSO.

1. Navigate to **Access Policy > Overview**.
2. Select **Access reports**.
3. Run the report for the last hour.
4. Review the logs for clues.

Use the **View session** variables link, for the session, to help understand if the APM receives expected Microsoft Entra claims.

No BIG-IP error message

If no BIG-IP error message appears, the issue might be related to the back-end request, or BIG-IP-to-application SSO.

1. Navigate to **Access Policy > Overview**.
2. Select **Active Sessions**.
3. Select the active session link.

Use the **View Variables** link to help determine SSO issues, particularly if the BIG-IP APM doesn't obtain correct attributes.

Learn more:

- [Configuring LDAP remote authentication for Active Directory ↗](#)
- Go to [techdocs.f5.com](#) for [Manual Chapter: LDAP Query ↗](#)

Tutorial: Configure F5 BIG-IP Easy Button for header-based and LDAP single sign-on

Article • 04/19/2024

In this article, you can learn to secure header and LDAP-based applications using Microsoft Entra ID, by using the F5 BIG-IP Easy Button Guided Configuration 16.1. Integrating a BIG-IP with Microsoft Entra ID provides many benefits:

- Improved governance: See, [Zero Trust framework to enable remote work ↗](#) and learn more about Microsoft Entra pre-authentication
 - See also, [What is Conditional Access?](#) to learn about how it helps enforce organizational policies
- Full single sign-on (SSO) between Microsoft Entra ID and BIG-IP published services
- Manage identities and access from one control plane, the [Microsoft Entra admin center ↗](#)

To learn about more benefits, see [F5 BIG-IP and Microsoft Entra integration](#).

Scenario description

This scenario focuses on the classic, legacy application using **HTTP authorization headers** sourced from LDAP directory attributes, to manage access to protected content.

Because it's legacy, the application lacks modern protocols to support a direct integration with Microsoft Entra ID. You can modernize the app, but it's costly, requires planning, and introduces risk of potential downtime. Instead, you can use an F5 BIG-IP Application Delivery Controller (ADC) to bridge the gap between the legacy application and the modern ID control plane, with protocol transitioning.

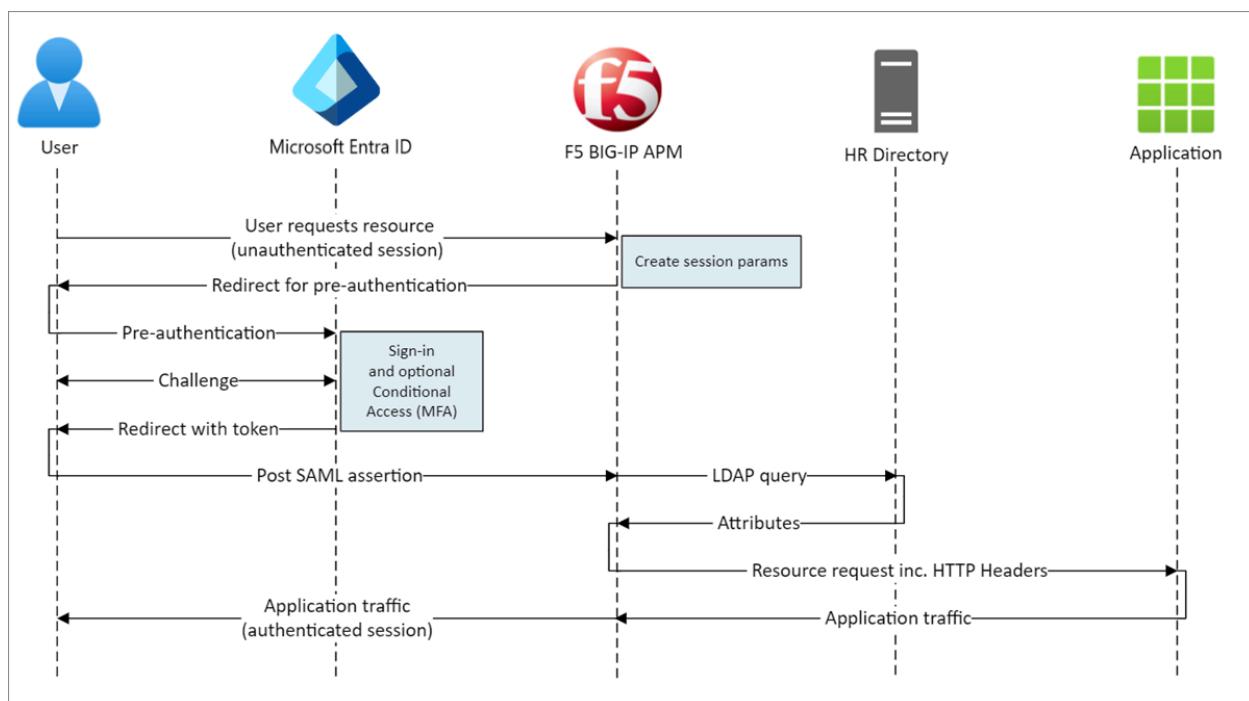
Having a BIG-IP in front of the app enables overlay of the service with Microsoft Entra pre-authentication and header-based SSO, improving the overall security posture of the application.

Scenario architecture

The secure hybrid access solution for this scenario has:

- **Application** - BIG-IP published service to be protected by Microsoft Entra ID secure hybrid access (SHA)
- **Microsoft Entra ID** - Security Assertion Markup Language (SAML) identity provider (IdP) that verifies user credentials, Conditional Access, and SAML-based SSO to the BIG-IP. With SSO, Microsoft Entra ID provides the BIG-IP with required session attributes.
- **HR system** - LDAP-based employee database as the source of truth for application permissions
- **BIG-IP** - Reverse proxy and SAML service provider (SP) to the application, delegating authentication to the SAML IdP, before performing header-based SSO to the back-end application

SHA for this scenario supports SP and IdP initiated flows. The following image illustrates the SP initiated flow.



1. User connects to application endpoint (BIG-IP)
2. BIG-IP APM access policy redirects user to Microsoft Entra ID (SAML IdP)
3. Microsoft Entra ID pre-authenticates user and applies enforced Conditional Access policies
4. User is redirected to BIG-IP (SAML SP) and SSO is performed using issued SAML token
5. BIG-IP requests more attributes from LDAP based HR system
6. BIG-IP injects Microsoft Entra ID and HR system attributes as headers in request to application
7. Application authorizes access with enriched session permissions

Prerequisites

Prior BIG-IP experience isn't necessary, but you need:

- An [Azure free account ↗](#), or a higher-tier subscription
- A BIG-IP or [deploy a BIG-IP Virtual Edition \(VE\) in Azure](#)
- Any of the following F5 BIG-IP licenses:
 - F5 BIG-IP® Best bundle
 - F5 BIG-IP Access Policy Manager™ (APM) standalone license
 - F5 BIG-IP Access Policy Manager™ (APM) add-on license on a BIG-IP F5 BIG-IP® Local Traffic Manager™ (LTM)
 - 90-day BIG-IP product [Free Trial ↗](#)
- User identities [synchronized](#) from an on-premises directory to Microsoft Entra ID
- One of the following roles: Global Administrator, Cloud Application Administrator, or Application Administrator.
- An [SSL Web certificate](#) for publishing services over HTTPS, or use default BIG-IP certificates while testing
- A header-based application or [set up a simple IIS header app](#) for testing
- A user directory that supports LDAP, such as Windows Active Directory Lightweight Directory Services (AD LDS), OpenLDAP etc.

BIG-IP configuration

This tutorial uses Guided Configuration 16.1 with an Easy Button template. With the Easy Button, admins don't go back and forth between Microsoft Entra ID and a BIG-IP to enable services for SHA. The deployment and policy management is handled between the APM Guided Configuration wizard and Microsoft Graph. This integration between BIG-IP APM and Microsoft Entra ID ensures applications support identity federation, SSO, and Microsoft Entra Conditional Access, reducing administrative overhead.

ⓘ Note

Replace example strings or values in this guide with those for your environment.

Register Easy Button

💡 Tip

Steps in this article might vary slightly based on the portal you start from.

Before a client or service can access Microsoft Graph, it's trusted by the [Microsoft identity platform](#).

This first step creates a tenant app registration to authorize the **Easy Button** access to Graph. With these permissions, the BIG-IP can push the configurations to establish a trust between a SAML SP instance for published application, and Microsoft Entra ID as the SAML IdP.

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > App registrations > New registration**.
3. Enter a display name for your application. For example, F5 BIG-IP Easy Button.
4. Specify who can use the application > **Accounts in this organizational directory only**.
5. Select **Register**.
6. Navigate to **API permissions** and authorize the following Microsoft Graph **Application permissions**:
 - Application.Read.All
 - Application.ReadWrite.All
 - Application.ReadWrite.OwnedBy
 - Directory.Read.All
 - Group.Read.All
 - IdentityRiskyUser.Read.All
 - Policy.Read.All
 - Policy.ReadWrite.ApplicationConfiguration
 - Policy.ReadWrite.ConditionalAccess
 - User.Read.All
7. Grant admin consent for your organization.
8. On **Certificates & Secrets**, generate a new **client secret**. Make a note of this secret.
9. On **Overview**, note the **Client ID** and **Tenant ID**.

Configure the Easy Button

Initiate the **APM Guided Configuration** to launch the **Easy Button** template.

1. Navigate to Access > Guided Configuration > Microsoft Integration and select Microsoft Entra Application.
2. Review the list of steps and select **Next**
3. To publish your application, follow the steps.



Configuration Properties

The **Configuration Properties** tab creates a BIG-IP application config and SSO object. The **Azure Service Account Details** section represents the client you registered in your Microsoft Entra tenant earlier, as an application. These settings allow a BIG-IP OAuth client to register a SAML SP in your tenant, with the SSO properties you would configure manually. Easy Button does this action for every BIG-IP service published and enabled for SHA.

Some of these settings are global, therefore can be reused to publish more applications, reducing deployment time and effort.

1. Enter a unique **Configuration Name** so admins can distinguish between Easy Button configurations.
2. Enable **Single Sign-On (SSO) & HTTP Headers**.
3. Enter the **Tenant ID**, **Client ID**, and **Client Secret** you noted when registering the Easy Button client in your tenant.
4. Confirm the BIG-IP can connect to your tenant.
5. Select **Next**.

Configuration Properties

General Properties ▾

Configuration Name

Type a name for this guided configuration.

Description ⓘ

 On

Single Sign-On (SSO) & HTTP Headers ⓘ



Endpoint Checks ⓘ



Additional Checks ⓘ

Azure Service Account Details ▾



Copy Account Info from Existing Configuration ⓘ

Tenant ID ⓘ

Client ID ⓘ

Client Secret ⓘ

 Connection is valid

Application Settings ▾



Use an existing Azure application ⓘ

Service Provider

The Service Provider settings define the properties for the SAML SP instance of the application protected through SHA.

1. Enter **Host**, the public fully qualified domain name (FQDN) of the application being secured.

2. Enter Entity ID, the identifier Microsoft Entra ID uses to identify the SAML SP requesting a token.

Service Provider

Advanced Settings

Service Provider Properties ▾

Host ⓘ	myevents.wacketywack.com
Entity ID ⓘ	https://myevents.wacketywack.com/My_Events
Description ⓘ	
Relay State ⓘ	

Use the optional **Security Settings** to specify whether Microsoft Entra ID encrypts issued SAML assertions. Encrypting assertions between Microsoft Entra ID and the BIG-IP APM provides assurance the content tokens can't be intercepted, and personal or corporate data can't be compromised.

3. From the Assertion Decryption Private Key list, select Create New

Security Settings ▾

Enable Encrypted Assertion ⓘ

Assertion Decryption Private Key ⓘ

Create New	<input type="button" value="Delete"/>
--Select--	<input type="button" value="Edit"/>
F5Demo	<input type="button" value="Edit"/>
Key Type: rsa-private	
Last Update Time: Thu, May 6, 2021	
Security Type: normal	
F5DemoCert	<input type="button" value="Edit"/>
Key Type: rsa-private	
Last Update Time: Fri, May 7, 2021	
Security Type: normal	
<input type="button" value="Create New"/>	

4. Select OK. The Import SSL Certificate and Keys dialog opens in a new tab.

5. Select PKCS 12 (IIS) to import your certificate and private key. After provisioning, close the browser tab to return to the main tab.

SSL Certificate/Key Source

Import Type	PKCS 12 (IIS) <input checked="" type="button"/>
Certificate and Key Name	<input checked="" type="radio"/> New <input type="radio"/> Overwrite Existing Contoso_SAML_Cert
Certificate and Key Source	Choose File <input type="button"/> No file chosen
Password	*****
Key Security	Normal <input type="button"/>
Free Space on Disk	8022 MB
<input type="button"/> Cancel <input type="button"/> Import	

6. Check Enable Encrypted Assertion.

7. If you enabled encryption, select your certificate from the **Assertion Decryption Private Key** list. BIG-IP APM uses this certificate private key to decrypt Microsoft Entra assertions.
8. If you enabled encryption, select your certificate from the **Assertion Decryption Certificate** list. BIG-IP uploads this certificate to Microsoft Entra ID to encrypt the issued SAML assertions.

Security Settings ▾

<input checked="" type="checkbox"/> Enable Encrypted Assertion <small>i</small>
Assertion Decryption Private Key <small>i</small>
Contoso_SAML_cert <input type="button"/> <input type="button"/>
Assertion Decryption Certificate <small>i</small>
Contoso_SAML_cert <input type="button"/> <input type="button"/>

Microsoft Entra ID

This section contains properties to manually configure a new BIG-IP SAML application in your Microsoft Entra tenant. Easy Button has application templates for Oracle PeopleSoft, Oracle E-business Suite, Oracle JD Edwards, SAP ERP, and an SHA template for other apps.

For this scenario, select **F5 BIG-IP APM Microsoft Entra ID Integration > Add**.

Azure Configuration

1. Enter **Display Name** of the app that the BIG-IP creates in your Microsoft Entra tenant, and the icon that users see on [MyApps portal](#).
2. Make no entry for **Sign On URL (optional)**.
3. To locate the certificate you imported, select the **Refresh** icon next to the **Signing Key** and **Signing Certificate**.
4. Enter the certificate password in **Signing Key Passphrase**.
5. Enable **Signing Option** (optional) to ensure BIG-IP accepts tokens and claims signed by Microsoft Entra ID.

SAML Signing Certificate

Signing Key i
Contoso_Wildcard_Cert ↻

Signing Certificate i
Contoso_Wildcard_Cert ↻

Signing Key Passphrase i
.....

Signing Option i
Sign SAML assertion

Signing Algorithm i
RSA-SHA256

6. **User and User Groups** are dynamically queried from your Microsoft Entra tenant and authorize access to the application. Add a user or group for testing, otherwise access is denied.

User And User Groups

<input type="checkbox"/> Name	Type	Description
<input type="checkbox"/> Contoso_Personnel	User Group	Contoso full time employees

User Attributes & Claims

When a user authenticates, Microsoft Entra ID issues a SAML token with a default set of claims and attributes uniquely identifying the user. The **User Attributes & Claims** tab shows the default claims to issue for the new application. It also lets you configure more claims.

For this example, include one more attribute:

1. For **Claim Name** enter **employeeid**.
2. For **Source Attribute** enter **user.employeeid**.

Azure Configuration	User Attributes & Claims	Additional User Attributes Conditional Access Policy																		
Required Claims ▾																				
<table border="1"><thead><tr><th>Claim Name</th><th>Value</th></tr></thead><tbody><tr><td>Unique User Identifier (Name ID)</td><td>user.userprincipalname</td></tr><tr><td>Identity</td><td>user.onpremisesamaccountname</td></tr></tbody></table>			Claim Name	Value	Unique User Identifier (Name ID)	user.userprincipalname	Identity	user.onpremisesamaccountname												
Claim Name	Value																			
Unique User Identifier (Name ID)	user.userprincipalname																			
Identity	user.onpremisesamaccountname																			
Additional Claims ▾																				
<table border="1"><thead><tr><th>Claim Name</th><th>Value</th><th>Add</th></tr></thead><tbody><tr><td>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress</td><td>user.mail</td><td> </td></tr><tr><td>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname</td><td>user.givenname</td><td> </td></tr><tr><td>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name</td><td>user.userprincipalname</td><td> </td></tr><tr><td>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname</td><td>user.surname</td><td> </td></tr><tr><td>employeeid</td><td>user.employeeid</td><td> </td></tr></tbody></table>			Claim Name	Value	Add	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	user.mail		http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	user.givenname		http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	user.userprincipalname		http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	user.surname		employeeid	user.employeeid	
Claim Name	Value	Add																		
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	user.mail																			
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	user.givenname																			
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	user.userprincipalname																			
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	user.surname																			
employeeid	user.employeeid																			

Additional User Attributes

On the **Additional User Attributes** tab, you can enable session augmentation for distributed systems such as Oracle, SAP, and other JAVA-based implementations requiring attributes stored in other directories. Attributes fetched from an LDAP source can be injected as more SSO headers to control access based on roles, Partner IDs, etc.

1. Enable the **Advanced Settings** option.
2. Check the **LDAP Attributes** check box.
3. In Choose Authentication Server, select **Create New**.

4. Depending on your setup, select either **Use pool** or **Direct Server Connection** mode to provide the **Server Address** of the target LDAP service. If using a single LDAP server, select **Direct**.
5. For **Service Port** enter 389, 636 (Secure), or another port your LDAP service uses.
6. For **Base Search DN** enter the distinguished name of the location containing the account the APM authenticates with, for LDAP service queries.

Azure Configuration	User Attributes & Claims	Additional User Attributes
Conditional Access Policy		
<p>Advanced Settings On</p> <p><input checked="" type="checkbox"/> LDAP Attributes ▼</p> <p>LDAP Server ▼</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Choose Authentication Server ⓘ</p> <p>Create New ▼</p> </div>		
<p>LDAP Server Properties ▼</p> <p>Server Connection ⓘ</p> <p><input type="radio"/> Use Pool <input checked="" type="radio"/> Direct</p> <p>Server Address 172.16.76.112</p> <p>Service Port 636</p> <p>Base Search DN cn=partners,contoso,dc=lds</p> <p>Admin DN cn=f5apm-partners,cn=partners,dc=contoso,dc=lds</p> <p>Type the distinguished name (DN) of the user with administrator rights.</p> <p>Admin Password *****</p> <p>Verify Admin Password *****</p> <p>Group Cache Lifetime 30 Days</p> <p>The Access device usually refreshes the group cache every 30 days. Type the number of days you would like to use.</p> <p>Timeout ⓘ 15 seconds</p>		

7. For **Search DN** enter the distinguished name of the location containing the user account objects that the APM queries via LDAP.
8. Set both membership options to **None** and add the name of the user object attribute to be returned from the LDAP directory. For this scenario: **eventroles**.

LDAP Query Properties ▾

Search Dn

Type the base node of the LDAP server search tree from which to search.

Search Filter ⓘ

Fetch groups to which the user or group belong ⓘ

Fetch users that belong to the group ⓘ

Required Attributes(optional) ⓘ

+ ×

Conditional Access Policy

Conditional Access policies are enforced after Microsoft Entra pre-authentication to control access based on device, application, location, and risk signals.

The **Available Policies** view lists Conditional Access policies that don't include user actions.

The **Selected Policies** view shows policies targeting all cloud apps. These policies can't be deselected or moved to the Available Policies list because they're enforced at a tenant level.

To select a policy to be applied to the application being published:

1. In the **Available Policies** list, select a policy.
2. Select the right arrow and move it to the **Selected Policies** list.

! Note

Selected policies have an **Include** or **Exclude** option checked. If both options are checked, the selected policy is not enforced.

Name	Include	Exclude	Apps
Block legacy authentication...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
Term of Use	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
Session Controls	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
MFA for all	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All

ⓘ Note

The policy list is enumerated once, when you initially select this tab. Use the **Refresh** button to manually force the wizard to query your tenant. This button appears when the application is deployed.

Virtual Server Properties

A virtual server is a BIG-IP data plane object represented by a virtual IP address listening for client requests to the application. Received traffic is processed and evaluated against the APM profile associated with the virtual server, before directed according to policy.

1. Enter the **Destination Address**, an available IPv4/IPv6 address the BIG-IP can use to receive client traffic. There should be a corresponding record in domain name server (DNS), which enables clients to resolve the external URL of your BIG-IP published application to this IP, instead of the application. Using a test PC localhost DNS is acceptable for testing.
2. For **Service Port** enter 443 and **HTTPS**.

- Check **Enable Redirect Port** and then enter **Redirect Port** to redirects incoming HTTP client traffic to HTTPS.
- The Client SSL Profile enables the virtual server for HTTPS, so client connections are encrypted over Transport Layer Security (TLS). Select the **Client SSL Profile** you created or leave the default while testing.

Virtual Server Properties

Advanced Settings

General Properties

Virtual Server
 Create New Use Existing

Destination Address

Service Port

Enable Redirect Port i

Redirect Port

Client SSL Profile i
 Create new Use Existing

Available	Selected
Common	Common
clientssl	Contoso_clientssl
clientssl-insecure-compatible	

[Create Profile in BIG-IP UI](#)

Pool Properties

The **Application Pool** tab has the services behind a BIG-IP represented as a pool, with one or more application servers.

- Choose from **Select a Pool**. Create a new pool or select one.
- Choose the **Load Balancing Method** such as Round Robin.

3. For Pool Servers select a node or specify an IP and port for the server hosting the header-based application.

The screenshot shows the 'Pool Properties' configuration screen. At the top, there is an 'Advanced Settings' toggle switch. Below it, a dropdown menu is set to 'Application Pool'. Under 'Select a Pool', there is a dropdown menu with 'Create New' selected. A note below says 'Select an existing pool or select Create New.' The 'Resources Properties' section includes 'Load Balancing Method' set to 'Round Robin' and a 'Pool Servers' table. The table has columns: IP Address/Node name, Port, Priority Group, and Action. It contains one row with values: 172.16.74.12, 443, 0, and a '+'/ '-' icon. The row for 172.16.74.12 and port 443 is highlighted with a red border.

ⓘ Note

Our back-end application sits on HTTP port 80. Switch to 443 if yours is HTTPS.

Single sign-on and HTTP Headers

Enabling SSO allows users to access BIG-IP published services without entering credentials. The **Easy Button** wizard supports Kerberos, OAuth Bearer, and HTTP authorization headers for SSO.

Use the following list to configure options.

- **Header Operation:** Insert
- **Header Name:** upn
- **Header Value:** %{session.saml.last.identity}
- **Header Operation:** Insert
- **Header Name:** employeeid
- **Header Value:** %{session.saml.last.attr.name.employeeid}

- **Header Operation:** Insert
- **Header Name:** eventroles
- **Header Value:** %{session.ldap.last.attr.eventroles}

Single Sign-On & HTTP Headers

Single Sign-On

HTTP Headers ▾

Header Operation	Header Name	Header Value	Delimiter	Action
insert	upn	%{session.saml.last.identity}	/	+ ×
insert	employeeid	%{session.saml.last.attr.name.employeeid}	/	+ ×
insert	eventroles	%{session.ldap.last.attr.eventroles}	/	+ ×

Cancel Save Draft Back **Save & Next**

ⓘ Note

APM session variables in curly brackets are case-sensitive. For example, if you enter OrclGUID and the Microsoft Entra attribute name is orclguid, an attribute mapping failure occurs.

Session management settings

The BIG-IPs session management settings define the conditions under which user sessions are terminated or allowed to continue, limits for users and IP addresses, and corresponding user info. Refer to the F5 article [K18390492: Security | BIG-IP APM operations guide](#) for details on these settings.

What isn't covered is Single Log Out (SLO) functionality, which ensures sessions between the IdP, the BIG-IP, and the user agent terminate as users sign out. When the Easy Button instantiates a SAML application in your Microsoft Entra tenant, it populates the sign-out URL with the APM SLO endpoint. An IdP-initiated sign-out from the Microsoft Entra My Apps portal terminates the session between the BIG-IP and a client.

The SAML federation metadata for the published application is imported from your tenant, which provides the APM with the SAML sign out endpoint for Microsoft Entra ID.

This action ensures an SP-initiated sign out terminates the session between a client and Microsoft Entra ID. The APM needs to know when a user signs out of the application.

If the BIG-IP webtop portal is used to access published applications, then APM processes sign-out to call the Microsoft Entra sign-out endpoint. But, consider a scenario wherein the BIG-IP webtop portal isn't used. The user can't instruct the APM to sign out. Even if the user signs out of the application, the BIG-IP is oblivious. Therefore, consider SP-initiated sign out to ensure sessions terminate securely. You can add an SLO function to an application Sign-out button, so it can redirect your client to the Microsoft Entra SAML or BIG-IP sign-out endpoint. The URL for SAML sign-out endpoint for your tenant is in [App Registrations > Endpoints](#).

If you can't make a change to the app, then consider having the BIG-IP listen for the application sign-out call, and upon detecting the request have it trigger SLO. Refer to the [Oracle PeopleSoft SLO guidance](#) to learn about BIG-IP iRules. For more information about using BIG-IP iRules, see:

- [K42052145: Configuring automatic session termination based on a URL-referenced file name ↗](#)
- [K12056: Overview of the Log-out URI Include option ↗](#)

Summary

This last step provides a breakdown of your configurations.

Select **Deploy** to commit settings and verify the application is in your tenant list of Enterprise applications.

Your application is published and accessible via SHA, either with its URL or through Microsoft application portals. For increased security, organizations using this pattern can block direct access to the application. This action forces a strict path through the BIG-IP.

Next steps

From a browser, in the [Microsoft MyApps portal ↗](#) connect to the application external URL or select the application icon. After authenticating against Microsoft Entra ID, you're redirected to the BIG-IP virtual server for the application and signed in through SSO.

See the following screenshot for output of the injected headers in our headers-based application.

My Events

https://myevents.contoso.com

Request Details

Session Id:	kiekszqbh3dydnpioygjvna	Request Type:	GET
Time of Request:	8/23/2021 1:48:29 PM	Status Code:	200
Request Encoding:	Unicode (UTF-8)	Response Encoding:	Unicode (UTF-8)

Server Variables

Name	Value
REMOTE_ADDR	172.16.76.16
REMOTE_PORT	32656
REQUEST_METHOD	GET
SCRIPT_NAME	/default.aspx
SERVER_PORT	80
SERVER_PROTOCOL	HTTP/1.1
SERVER_SOFTWARE	Microsoft-IIS/8.5
URL	/default.aspx
HTTP_CONNECTION	keep-alive
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,app
HTTP_ACCEPT_ENCODING	gzip, deflate, br
HTTP_ACCEPT_LANGUAGE	en-GB,en;q=0.9
HTTP_COOKIE	TIN=297000; LastMRH_Session=a16276ee; F5_ST=1z1z1z1629046109z604800
HTTP_HOST	myevents.contoso.com
HTTP_REFERER	https://login.microsoftonline.com/
HTTP_USER_AGENT	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/9
HTTP_UPN	ruby.a@contoso.com
HTTP_EMPLOYEEID	564738
HTTP_EVENTROLES	Approver

Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.8.4330.0

For increased security, organizations using this pattern can block direct access to the application. This action forces a strict path through the BIG-IP.

Advanced deployment

The Guided Configuration templates can lack flexibility to achieve specific requirements.

In BIG-IP, you can disable the Guided Configuration **strict management mode**. You can then manually change your configurations, although the bulk of your configurations are automated through the wizard-based templates.

For your applications configurations, you can navigate to **Access > Guided Configuration** and select the small **padlock** icon on the far-right of the row.



At this point, changes with the wizard UI are no longer possible, but all BIG-IP objects associated with the published instance of the application are unlocked for direct management.

Note

Re-enabling strict mode and deploying a configuration overwrites any settings performed outside the Guided Configuration UI. We recommend the advanced configuration method for production services.

Troubleshooting

BIG-IP logging

BIG-IP logging can help isolate issues with connectivity, SSO, policy violations, or misconfigured variable mappings.

To troubleshoot, you can increase the log verbosity level.

1. Navigate to **Access Policy > Overview > Event Logs > Settings**.
2. Select the row for your published application then **Edit > Access System Logs**.
3. From the SSO list, select **Debug**, then **OK**.

Reproduce your issue, then inspect the logs, but revert this setting when finished.

Verbose mode generates significant amounts of data.

BIG-IP error page

If a BIG-IP error appears after Microsoft Entra pre-authentication, it's possible the issue relates to SSO from Microsoft Entra ID to the BIG-IP.

1. Navigate to **Access > Overview > Access reports**.
2. Run the report for the last hour to see if the logs provide any clues.
3. Use the **View Variables** link for your session to understand if the APM is receiving the expected claims from Microsoft Entra ID.

Back-end request

If there's no error page, then the issue is probably related to the back-end request, or SSO from the BIG-IP to the application.

1. Navigate to **Access Policy > Overview > Active Sessions** and select the link for your active session.
2. To help root-cause the issue, use the **View Variables** link, particularly if the BIG-IP APM fails to obtain the right attributes from Microsoft Entra ID or another source.

Validate the APM service account

To validate the APM service account for LDAP queries, use the following command from the BIG-IP bash shell. Confirm authentication and query of a user object.

```
ldapsearch -xLLL -H 'ldap://192.168.0.58' -b "CN=partners,dc=contoso,dc=lds" -s sub  
-D "CN=f5-apm,CN=partners,DC=contoso,DC=lds" -w 'P@55w0rd!' "(cn=testuser)"
```

For more information, see the F5 article [K11072: Configuring LDAP remote authentication for Active Directory](#). You can use a BIG-IP reference table to help diagnose LDAP-related issues in AskF5 document, [LDAP Query](#).

Tutorial: Configure F5 BIG-IP Easy Button for SSO to Oracle EBS

Article • 10/23/2023

Learn to secure Oracle E-Business Suite (EBS) using Microsoft Entra ID, with F5 BIG-IP Easy Button Guided Configuration. Integrating a BIG-IP with Microsoft Entra ID has many benefits:

- Improved Zero Trust governance through Microsoft Entra preauthentication and Conditional Access
 - See, [What is Conditional Access?](#)
 - See, [Zero Trust security](#)
- Full SSO between Microsoft Entra ID and BIG-IP published services
- Managed identities and access from one control plane
 - See, the [Microsoft Entra admin center](#) ↗

Learn more:

- [Integrate F5 BIG-IP with Microsoft Entra ID](#)
- [Enable SSO for an enterprise application](#)

Scenario description

This scenario covers the classic Oracle EBS application that uses HTTP authorization headers to manage access to protected content.

Legacy applications lack modern protocols to support Microsoft Entra integration. Modernization is costly, time consuming, and introduces downtime risk. Instead, use an F5 BIG-IP Application Delivery Controller (ADC) to bridge the gap between legacy applications and the modern ID control plane, with protocol transitioning.

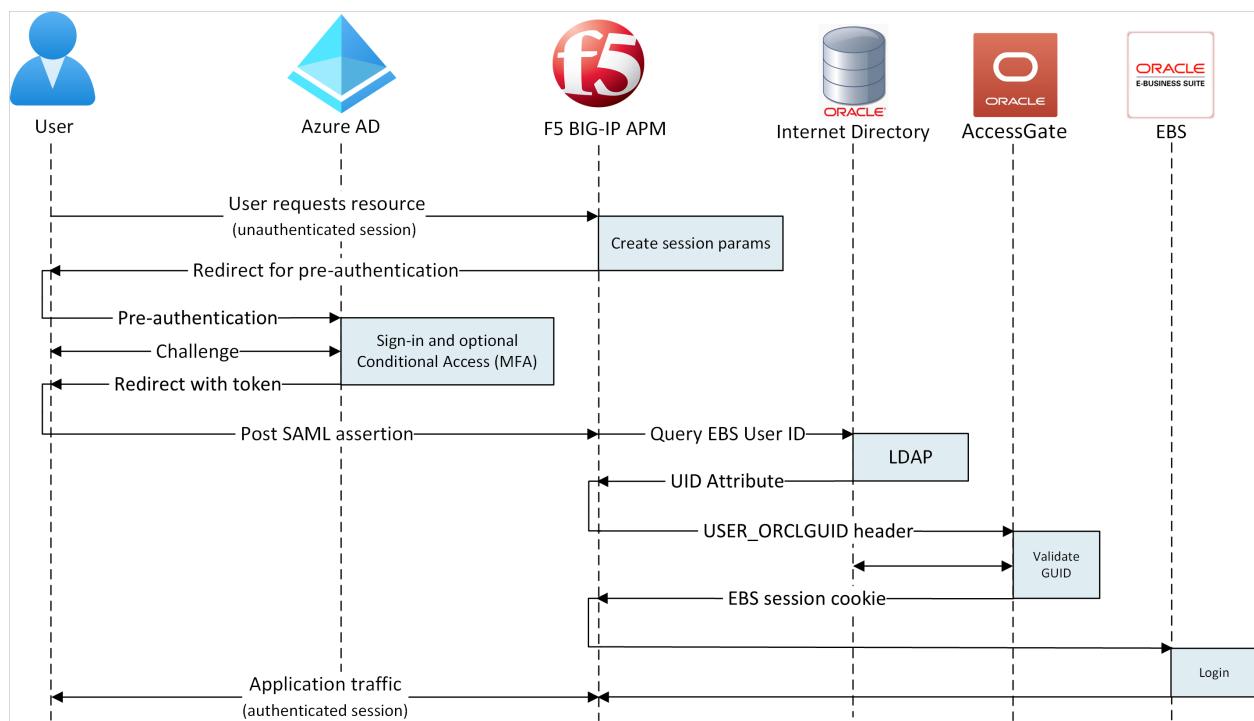
A BIG-IP in front of the app enables overlay of the service with Microsoft Entra preauthentication and header-based SSO. This configuration improves application security posture.

Scenario architecture

The secure hybrid access (SHA) solution has the following components:

- **Oracle EBS application** - BIG-IP published service to be protected by Microsoft Entra SHA
- **Microsoft Entra ID** - Security Assertion Markup Language (SAML) identity provider (IdP) that verifies user credentials, Conditional Access, and SAML-based SSO to the BIG-IP
 - With SSO, Microsoft Entra ID provides BIG-IP session attributes
- **Oracle Internet Directory (OID)** - hosts the user database
 - BIG-IP verifies authorization attributes with LDAP
- **Oracle E-Business Suite AccessGate** - validates authorization attributes with the OID service, then issues EBS access cookies
- **BIG-IP** - reverse-proxy and SAML service provider (SP) to the application
 - Authentication is delegated to the SAML IdP, then header-based SSO to the Oracle application occurs

SHA supports SP- and IdP-initiated flows. The following diagram illustrates the SP-initiated flow.



1. User connects to application endpoint (BIG-IP).
2. BIG-IP APM access policy redirects user to Microsoft Entra ID (SAML IdP).
3. Microsoft Entra preauthenticates user and applies Conditional Access policies.
4. User is redirected to BIG-IP (SAML SP) and SSO occurs using the issued SAML token.
5. BIG-IP performs an LDAP query for the user Unique ID (UID) attribute.
6. BIG-IP injects returned UID attribute as user_orclguid header in Oracle EBS session cookie request to Oracle AccessGate.

7. Oracle AccessGate validates UID against OID service and issues Oracle EBS access cookie.
8. Oracle EBS user headers and cookie sent to application and returns the payload to the user.

Prerequisites

You need the following components:

- An Azure subscription
 - If you don't have one, get an [Azure free account](#)
- Global Administrator, Cloud Application Administrator, or Application Administrator.
- A BIG-IP or deploy a BIG-IP Virtual Edition (VE) in Azure
 - See, [Deploy F5 BIG-IP Virtual Edition VM in Azure](#)
- Any of the following F5 BIG-IP license SKUs:
 - F5 BIG-IP® Best bundle
 - F5 BIG-IP Access Policy Manager™ (APM) standalone license
 - F5 BIG-IP Access Policy Manager™ (APM) add-on license on a BIG-IP F5 BIG-IP® Local Traffic Manager™ (LTM)
 - 90-day BIG-IP full feature trial. See, [Free Trials](#).
- User identities synchronized from an on-premises directory to Microsoft Entra ID
 - See, [Microsoft Entra Connect Sync: Understand and customize synchronization](#)
- An SSL certificate to publish services over HTTPS, or use default certificates while testing
 - See, [SSL profile](#)
- An Oracle EBS, Oracle AccessGate, and an LDAP-enabled Oracle Internet Database (OID)

BIG-IP configuration method

This tutorial uses the Guided Configuration v16.1 Easy Button template. With the Easy Button, admins no longer go back and forth to enable services for SHA. The APM Guided Configuration wizard and Microsoft Graph handle deployment and policy management. This integration ensures applications support identity federation, SSO, and Conditional Access, thus reducing administrative overhead.

Note

Replace example strings or values with those in your environment.

Register the Easy Button

💡 Tip

Steps in this article may vary slightly based on the portal you start from.

Before a client or service accesses Microsoft Graph, the Microsoft identity platform must trust it.

Learn more: [Quickstart: Register an application with the Microsoft identity platform](#)

Create a tenant app registration to authorize the Easy Button access to Graph. The BIG-IP pushes configurations to establish a trust between a SAML SP instance for published application, and Microsoft Entra ID as the SAML IdP.

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Identity > Applications > App registrations > New registration**.
3. Enter an application **Name**. For example, F5 BIG-IP Easy Button.
4. Specify who can use the application > **Accounts in this organizational directory only**.
5. Select **Register**.
6. Navigate to **API permissions**.
7. Authorize the following Microsoft Graph **Application permissions**:
 - Application.Read.All
 - Application.ReadWrite.All
 - Application.ReadWrite.OwnedBy
 - Directory.Read.All
 - Group.Read.All
 - IdentityRiskyUser.Read.All
 - Policy.Read.All
 - Policy.ReadWrite.ApplicationConfiguration
 - Policy.ReadWrite.ConditionalAccess
 - User.Read.All
8. Grant admin consent for your organization.

9. Go to Certificates & Secrets.

10. Generate a new Client Secret. Make a note of the Client Secret.

11. Go to Overview. Make a note of the Client ID and Tenant ID.

Configure the Easy Button

1. Initiate the APM Guided Configuration.

2. Start the Easy Button template.

3. Navigate to Access > Guided Configuration > Microsoft Integration.

4. Select Microsoft Entra Application.

The screenshot shows the 'Access > Guided Configuration' screen. At the top right, it says 'Version: 8.0' and has a link to 'Upgrade Guided Configuration'. Below are four cards: 'Federation' (fingerprint icon), 'Zero Trust' (shield with checkmark icon), 'Microsoft Integration' (Windows logo icon, highlighted with a blue border), and 'API Protection' (gear and shield icon). The 'Microsoft Integration' card is expanded, showing its details. It has a sub-section titled 'Microsoft Integration' which describes the integration with Microsoft Azure AD for secure access to mission-critical applications. Below this are two more sections: 'ADFS Proxy' and 'Azure AD Application'. 'Azure AD Application' is highlighted with a red border. Both sections have descriptions. At the bottom, there is a section for 'Exchange Proxy'.

Microsoft Integration

BIG-IP APM integration with Microsoft Azure AD provides secure and seamless access for all modern and classic mission-critical applications. It also provides additional security to Microsoft solutions, including Microsoft ADFS, Sharepoint, and Exchange with device posture context and adaptive MFA authentication.

ADFS Proxy

Consolidate and simplify deployments by load-balancing ADFS farm and performing ADFS proxy functionality.

Azure AD Application

Configure secure application access with Single sign-on across your hybrid identity environment by instantiating an Azure AD application template. It will setup BIG-IP APM as a SAML SP and Azure AD as an Identity Provider.

Exchange Proxy

Configure BIG-IP APM to provide proxy authentication and secure remote access to Exchange HTTP-based services.

5. Review the configuration options.

6. Select Next.

Configuring the solution using the below steps will create the required objects:

	Configuration Properties Configure the Azure service account and the application settings.
	Service Provider Uniquely identify the SAML Service Provider and specify security settings.
	Azure Active Directory Select Azure application template and update configuration properties and user attributes.
	Virtual Server Provide the IP address and port for the network traffic and select a client-side SSL profile.
	Pool Configure a pool and pool members for load balancing network traffic.
	Single Sign-On (SSO) and HTTP Headers (Optional) Configure Single Sign-On properties and HTTP Headers.
	Endpoint Checks (Optional) Select client types and the endpoint inspections to perform on them.
	Session Management Configure session timeouts and user settings.

[Cancel](#) [Next](#)

7. Use the graphic to help publish your application.



Configuration Properties

The **Configuration Properties** tab creates a BIG-IP application config and SSO object. The **Azure Service Account Details** section represents the client you registered in your Microsoft Entra tenant, as an application. With these settings, a BIG-IP OAuth client registers a SAML SP in your tenant, with SSO properties. Easy Button does this action for BIG-IP services published and enabled for SHA.

To reduce time and effort, reuse global settings to publish other applications.

1. Enter a **Configuration Name**.
2. For **Single sign-on (SSO) & HTTP Headers**, select **On**.
3. For **Tenant ID**, **Client ID**, and **Client Secret** enter what you noted during Easy Button client registration.
4. Confirm the BIG-IP connects to your tenant.

5. Select Next.

Configuration Properties

General Properties ▾

Configuration Name

OracleEBS

Type a name for this guided configuration.

Description ⓘ

On

Single Sign-On (SSO) & HTTP Headers ⓘ



Endpoint Checks ⓘ



Additional Checks ⓘ

Azure Service Account Details ▾



Copy Account Info from Existing Configuration ⓘ

Tenant ID ⓘ

81b650e9-9758-4f8c-9e6b-2a201fb390d

Client ID ⓘ

8b52b0b7-cf0e-4b9f-8f18-0a9ea3300e9e

Client Secret ⓘ

....

Test Connection

Connection is valid

Application Settings ▾



Use an existing Azure application ⓘ

Service Provider

Use Service Provider settings for the properties of the SAML SP instance of the protected application.

1. For **Host**, enter the public FQDN of the application.
2. For **Entity ID**, enter the identifier Microsoft Entra ID uses for the SAML SP requesting a token.

Service Provider

Advanced Settings

Service Provider Properties ▾

Host ⓘ

eportal.contoso.com

Entity ID ⓘ

https://eportal.contoso.com/

Description ⓘ

Relay State ⓘ

/OA_HTML/OA.jsp?OAFunc=OAHOME PAGE

3. (Optional) In **Security Settings**, select or clear the **Enable Encrypted Assertion** option. Encrypting assertions between Microsoft Entra ID and the BIG-IP APM means the content tokens can't be intercepted, nor personal or corporate data compromised.

4. From the **Assertion Decryption Private Key** list, select **Create New**

Security Settings ▾

Enable Encrypted Assertion ⓘ

Assertion Decryption Private Key ⓘ

--Select--

F5Demo Key Type: rsa-private Last Update Time: Thu, May 6, 2021 Security Type: normal
F5DemoCert Key Type: rsa-private Last Update Time: Fri, May 7, 2021 Security Type: normal

5. Select OK.

6. The **Import SSL Certificate and Keys** dialog appears in a new tab.

7. Select **PKCS 12 (IIS)**.

8. The certificate and private key are imported.

9. Close the browser tab to return to the main tab.

System » Certificate Management : Traffic Certificate Management : SSL Certificate List » Import SSL Certificates and Keys

SSL Certificate/Key Source

Import Type	PKCS 12 (IIS) <input type="button" value=""/>
Certificate and Key Name	<input checked="" type="radio"/> New <input type="radio"/> Overwrite Existing Contoso_SAML_Cert <input type="button" value=""/>
Certificate and Key Source	Choose File No file chosen
Password	***** <input type="button" value=""/>
Key Security	Normal <input type="button" value=""/>
Free Space on Disk	8022 MB

Cancel Import

10. Select **Enable Encrypted Assertion**.

11. For enabled encryption, from the **Assertion Decryption Private Key** list, select the certificate private key BIG-IP APM uses to decrypt Microsoft Entra assertions.

12. For enabled encryption, from the **Assertion Decryption Certificate** list, select the certificate BIG-IP uploads to Microsoft Entra ID to encrypt the issued SAML assertions.

Security Settings ▾

Enable Encrypted Assertion i

Assertion Decryption Private Key i
Contoso_SAML_cert

Assertion Decryption Certificate i
Contoso_SAML_cert

Microsoft Entra ID

Easy Button has application templates for Oracle PeopleSoft, Oracle E-Business Suite, Oracle JD Edwards, SAP ERP and a generic SHA template. The following screenshot is the Oracle E-Business Suite option under Azure Configuration.

1. Select **Oracle E-Business Suite**.

2. Select **Add**.

Azure Configuration !

User Attributes & Claims
Conditional Access Policy

Additional User Attributes

Configuration Properties ▾

Search Q



F5 BIG-IP APM
Azure AD...



Oracle
PeopleSoft ...



SAP ERP Central
Component...



Oracle E-
Business Suite...



JD Edwards -
Protected by F...

Add

Azure Configuration

1. Enter a **Display Name** for the app BIG-IP creates in your Microsoft Entra tenant, and the icon on MyApps.
2. In **Sign On URL (optional)**, enter the EBS application public FQDN.
3. Enter the default path for the Oracle EBS homepage.

Azure Configuration

User Attributes & Claims
Conditional Access Policy

Additional User Attributes

Advanced Settings []

Configuration Properties ▾



Oracle E-
Business Suite...

Change

Display Name ⓘ
OracleEBS

Sign On URL ⓘ
https://eportal.contoso.com/OA_HTML/OA.jsp?OAFunc=OAHOME PAGE

4. Next to the **Signing Key** and **Signing Certificate**, select the refresh icon.
5. Locate the certificate you imported.
6. In **Signing Key Passphrase**, enter the certificate password.
7. (Optional) Enable **Signing Option**. This option ensures BIG-IP accepts tokens and claims signed by Microsoft Entra ID.

SAML Signing Certificate

Signing Key ⓘ	Contoso_Wildcard_Cert	⟳
Signing Certificate ⓘ	Contoso_Wildcard_Cert	⟳
Signing Key Passphrase ⓘ	
Signing Option ⓘ	Sign SAML assertion	
Signing Algorithm ⓘ	RSA-SHA256	

8. For **User And User Groups**, add a user or group for testing, otherwise all access is denied. Users and user groups are dynamically queried from the Microsoft Entra tenant and authorize access to the application.

User And User Groups

+ Add		Remove
<input type="checkbox"/> Name	Type	Description
<input type="checkbox"/> Contoso_Personnel	User Group	Contoso full time employees

User Attributes & Claims

When a user authenticates, Microsoft Entra ID issues a SAML token with default claims and attributes identifying the user. The **User Attributes & Claims** tab has default claims to issue for the new application. Use this area to configure more claims. If needed, add Microsoft Entra attributes, however the Oracle EBS scenario requires the default attributes.

[Azure Configuration](#)[User Attributes & Claims](#)[Additional User Attributes](#)[Conditional Access Policy](#)**Required Claims ▾**

Claim Name	Value
Unique User Identifier (Name ID)	user.userprincipalname
Identity	user.onpremisessamaccountname

Additional Claims ▾

Add		
Claim Name	Value	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	user.mail	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	user.given...	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	user.user...	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	user.surn...	

Additional User Attributes

The **Additional User Attributes** tab supports distributed systems that require attributes stored in directories for session augmentation. Attributes fetched from an LDAP source are injected as more SSO headers to control access based on roles, partner ID, etc.

1. Enable the **Advanced Settings** option.
2. Check the **LDAP Attributes** check box.
3. In **Choose Authentication Server**, select **Create New**.
4. Depending on your setup, select **Use pool** or **Direct** server connection mode for the target LDAP service server address. For a single LDAP server, select **Direct**.
5. For **Service Port**, enter **3060** (Default), **3161** (Secure), or another port for the Oracle LDAP service.
6. Enter a **Base Search DN**. Use the distinguished name (DN) to search for groups in a directory.
7. For **Admin DN**, enter the account distinguished name APM uses to authenticate LDAP queries.

8. For Admin Password, enter the password.

Azure Configuration	User Attributes & Claims	Additional User Attributes
Conditional Access Policy		
<p>Advanced Settings <input type="button" value="On"/></p> <p><input checked="" type="checkbox"/> LDAP Attributes ▾</p> <p>LDAP Server ▾</p> <p>Choose Authentication Server ⓘ <input type="button" value="Create New"/></p>		
<p>LDAP Server Properties ▾</p> <p>Server Connection ⓘ <input type="radio"/> Use Pool <input checked="" type="radio"/> Direct</p> <p>Server Address <input type="text" value="172.16.76.114"/></p> <p>Type the IP address of your LDAP or LDAPS server.</p> <p>Service Port <input type="text" value="3060"/></p> <p>Accept the default value (389 for LDAP and 636 for LDAPS) or type the port number used on your authentication server.</p> <p>Base Search DN <input type="text" value="cn=users,dc=contoso,dc=dir"/></p> <p>Type the base distinguished name from which to search. This search DN is used to search groups across a whole directory.</p> <p>Admin DN <input type="text" value="cn=f5ldap"/></p> <p>Type the distinguished name (DN) of the user with administrator rights.</p> <p>Admin Password <input type="password" value="....."/></p> <p>Type the administrator password for your authentication server.</p> <p>Verify Admin Password <input type="password" value="....."/></p> <p>Type the password again.</p> <p>Group Cache Lifetime <input type="text" value="30"/> Days</p> <p>The Access device usually refreshes the group cache every 30 days. Type the number of days you would like to use.</p> <p>Timeout ⓘ <input type="text" value="15"/> seconds</p>		

9. Leave the default LDAP Schema Attributes.

LDAP Schema Attributes

User Object Class ⓘ
user

User Membership
memberOf

If the user object maintains a group membership, this specifies the value of the membership attribute and defaults to memberOf.

Group Object Class
group

Specifies the value of the objectClass attribute for a group object. Defaults to group.

Group Membership
memberOf

If the group object maintains membership in other groups, specifies the value of the membership attribute. Defaults to memberOf.

Group Member
member

If the group object maintains a list of users that belong to it, specifies the value of the attribute that indicates this. Defaults to member.

Group Member Value
dn

If the Group Member attribute is specified, this field specifies the attribute that is used to add users to a group. Defaults to dn.

10. Under **LDAP Query Properties**, for **Search Dn** enter the LDAP server base node for user object search.
11. For **Required Attributes**, enter the user object attribute name to be returned from the LDAP directory. For EBS, the default is **orclguid**.

LDAP Query Properties ▾

Search Dn
cn=users,dc=contoso,dc=dir

Type the base node of the LDAP server search tree from which to search.

Search Filter ⓘ
cn=%{subsession.logon.last.username}

Fetch groups to which the user or group belong ⓘ
Direct

Fetch users that belong to the group ⓘ
Direct

Required Attributes(optional) ⓘ
orclguid

Conditional Access Policy

Conditional Access policies control access based on device, application, location, and risk signals. Policies are enforced after Microsoft Entra preauthentication. The Available Policies view has Conditional Access policies with no user actions. The Selected Policies view has policies for cloud apps. You can't deselect these policies or move them to Available Policies because they're enforced at the tenant level.

To select a policy for the application to be published:

1. In **Available Policies**, select a policy.
2. Select the **right arrow**.
3. Move the policy to **Selected Policies**.

ⓘ Note

The **Include** or **Exclude** option is selected for some policies. If both options are checked, the policy is unenforced.

Azure Configuration User Attributes & Claims Additional User Attributes Conditional Access Policy

Conditional Access Policy ▾

[View Conditional Access policies in the Azure Portal](#)

Available Policies ⓘ		Selected Policies ⓘ			
Items: 11	Filter by Name...	Items: 4	Filter by Name...		
Name	Apps	Name	Include	Exclude	Apps
<input type="checkbox"/> F5 Intranet - MCAS	Selected	<input type="checkbox"/> Block legacy authenticat...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
<input type="checkbox"/> Intranet - Custom Control	Selected	<input type="checkbox"/> Term of Use	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
<input type="checkbox"/> Intranet - Block downloads	Selected	<input type="checkbox"/> Session Controls	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
<input type="checkbox"/> SPO - Managed endpoints only	Selected	<input type="checkbox"/> MFA for all	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
<input type="checkbox"/> SharePoint Restricted Content...	Selected				
<input type="checkbox"/> Breakglass	None				
<input type="checkbox"/> Managed or Compliant endpo...	None				
<input type="checkbox"/> Restrict office 365 to Register...	Selected				
<input type="checkbox"/> Block all	Selected				
<input type="checkbox"/> Block Legacy clients (Office, I...	Selected				

ⓘ Note

Select the **Conditional Access Policy** tab and the policy list appears. Select **Refresh** and the wizard queries your tenant. Refresh appears for deployed applications.

Virtual Server Properties

A virtual server is a BIG-IP data plane object represented by a virtual IP address listening for application client requests. Received traffic is processed and evaluated against the APM profile associated with the virtual server. Then, traffic is directed according to policy.

1. Enter a **Destination Address**, an IPv4 or IPv6 address BIG-IP uses to receive client traffic. Ensure a corresponding record in DNS that enables clients to resolve the external URL, of the BIG-IP published application, to the IP. Use a test computer localhost DNS for testing.
2. For **Service Port**, enter **443**, and select **HTTPS**.
3. Select **Enable Redirect Port**.
4. For **Redirect Port**, enter **80**, and select **HTTP**. This action redirects incoming HTTP client traffic to HTTPS.

5. Select the **Client SSL Profile** you created, or leave the default for testing. Client SSL Profile enables the virtual server for HTTPS. Client connections are encrypted over TLS.

Virtual Server Properties

Advanced Settings

General Properties ▾

Virtual Server
 Create New Use Existing

Destination Address ⓘ
172.16.76.27

Service Port ⓘ
443

Enable Redirect Port ⓘ

Redirect Port ⓘ
80

Client SSL Profile ⓘ
 Create new Use Existing

Available	Selected
<input style="border: none; background-color: #f0f0f0; padding: 2px 5px;" type="button" value="Filter"/>	
Common	Common
clientssl	Contoso_clientssl
clientssl-insecure-compatible	

[Create Profile in BIG-IP UI](#)

Pool Properties

The **Application Pool** tab has services behind a BIG-IP, a pool with one or more application servers.

1. From **Select a Pool**, select **Create New**, or select another option.
2. For **Load Balancing Method**, select **Round Robin**.
3. Under **Pool Servers**, select and enter an **IP Address/Node Name** and **Port** for the servers hosting Oracle EBS.
4. Select **HTTPS**.

Pool Properties

Advanced Settings

Application Pool ▾

Select a Pool

Create New

Select an existing pool or select Create New.

Resources Properties

Load Balancing Method ⓘ

Round Robin

Pool Servers ⓘ

IP Address/Node name	Port	Priority Group	Action
172.16.74.12	443	HTTPS	<input type="button" value="+"/> <input type="button" value="x"/>

5. Under Access Gate Pool confirm the Access Gate Subpath.

6. For Pool Servers select and enter an IP Address/Node Name and Port for the servers hosting Oracle EBS.

7. Select HTTPS.

Access Gate Pool ▾

Select a Pool
Create New

Select an existing pool or select Create New.

Access Gate Subpath ⓘ
/accessgate/

Health Monitors ⓘ

Available	Selected
/Common/gateway_icmp	
/Common/http	
/Common/http_head_f5	

Resources Properties

Load Balancing Method ⓘ
Round Robin

Pool Servers ⓘ

IP Address/Node name	Port	Priority Group	Action
172.16.76.16	443	HTTPS	0

Single Sign-On & HTTP Headers

The Easy Button wizard supports Kerberos, OAuth Bearer, and HTTP authorization headers for SSO to published applications. The Oracle EBS application expects headers, therefore enable HTTP headers.

1. On **Single Sign-On & HTTP Headers**, select **HTTP Headers**.
2. For **Header Operation**, select **replace**.
3. For **Header Name**, enter **USER_NAME**.
4. For **Header Value**, enter **%{session.sso.token.last.username}**.
5. For **Header Operation**, select **replace**.
6. For **Header Name**, enter **USER_ORCLGUID**.
7. For **Header Value**, enter **%{session.ldap.last.attr.orclguid}**.

Single Sign-On & HTTP Headers

Single Sign-On

HTTP Headers ▾

SSO Headers ⓘ

Header Operation	Header Name	Header Value	Delimiter	Action
replace	USER_NAME	{session.sso.token.last.username}		+ ×
replace	USER_ORCLGUID	{session.ldap.last.attrib.orclguid}		+ ×

ⓘ Note

APM session variables in curly brackets are case-sensitive.

Session Management

Use BIG-IP Session Management to define conditions for user session termination or continuation.

To learn more, go to support.f5.com for [K18390492: Security | BIG-IP APM operations guide ↗](#)

Single Log-Out (SLO) functionality ensures sessions between the IdP, BIG-IP, and the user agent, terminate when users sign out. When the Easy Button instantiates a SAML application in your Microsoft Entra tenant, it populates the Logout URL with the APM SLO endpoint. Thus, IdP-initiated sign out, from the My Apps portal, terminates the session between the BIG-IP and a client.

See, Microsoft [My Apps ↗](#)

The SAML federation metadata for the published application is imported from the tenant. This action provides the APM with the SAML sign out endpoint for Microsoft Entra ID. Then, SP-initiated sign out terminates the client and Microsoft Entra session. Ensure the APM knows when a user signs out.

If you use the BIG-IP webtop portal to access published applications, APM processes a sign out to call the Microsoft Entra sign-out endpoint. If you don't use the BIG-IP webtop portal, the user can't instruct the APM to sign out. If the user signs out of the application, the BIG-IP is oblivious to the action. Ensure SP-initiated sign out triggers

secure sessions termination. Add an SLO function to the applications **Sign out** button to redirect the client to the Microsoft Entra SAML or BIG-IP sign out endpoint. Find the SAML sign out endpoint URL for your tenant in **App Registrations > Endpoints**.

If you can't change the app, have the BIG-IP listen for the application sign out call and then trigger SLO.

Learn more:

- [PeopleSoft SLO Logout](#)
- Go to support.f5.com for:
 - [K42052145: Configuring automatic session termination \(logout\) based on a URI-referenced file name ↗](#)
 - [K12056: Overview of the Logout URI Include option ↗](#)

Deploy

1. Select **Deploy** to commit settings.
2. Verify the application appears in the tenant Enterprise applications list.

Test

1. From a browser, connect to the Oracle EBS application external URL, or select the application icon in the [My Apps ↗](#).
2. Authenticate to Microsoft Entra ID.
3. You're redirected to the BIG-IP virtual server for the application and signed in by SSO.

For increased security, block direct application access, thereby enforcing a path through the BIG-IP.

Advanced deployment

Sometimes, the Guided Configuration templates lack flexibility for requirements.

Learn more: [Tutorial: Configure F5 BIG-IP's Access Policy Manager for header-based SSO](#).

Manually change configurations

Alternatively, in BIG-IP disable the Guided Configuration strict management mode to manually change configurations. Wizard templates automate most configurations.

1. Navigate to Access > **Guided Configuration**.
2. On the right end of the row for your application configuration, select the **padlock** icon.



After you disable strict mode, you can't make changes with the wizard. However, BIG-IP objects associated with the published app instance are unlocked for management.

ⓘ Note

If you re-enable strict mode, new configurations overwrite settings performed without the Guided Configuration. We recommend the advanced configuration method for production services.

Troubleshooting

Use the following instructions to help troubleshoot issues.

Increase log verbosity

Use BIG-IP logging to isolate issues with connectivity, SSO, policy violations, or misconfigured variable mappings. Increase the log verbosity level.

1. Navigate to Access Policy > Overview > Event Logs.
2. Select **Settings**.
3. Select the row for your published application.
4. Select **Edit > Access System Logs**.
5. From the SSO list, select **Debug**.
6. Select **OK**.
7. Reproduce the issue.
8. Inspect the logs.

Revert the settings changes because verbose mode generates excessive data.

BIG-IP error message

If a BIG-IP error appears after Microsoft Entra preauthentication, the issue might relate to Microsoft Entra ID and BIG-IP SSO.

1. Navigate to **Access > Overview.
2. Select **Access reports**.
3. Run the report for the last hour.
4. Review the logs for clues.

Use the **View session** link for your session to confirm the APM receives expected Microsoft Entra claims.

No BIG-IP error message

If no BIG-IP error page appears, the issue might relate to the back-end request, or BIG-IP and application SSO.

1. Navigate to **Access Policy > Overview**.
2. Select **Active Sessions**.
3. Select the link for your active session.

Use the **View Variables** link to investigate SSO issues, particularly if the BIG-IP APM doesn't obtain correct attributes from Microsoft Entra ID, or another source.

Learn more:

- Go to devcentral.f5.com for [APM variable assign examples ↗](#)
- Go to techdocs.f5.com for [Manual Chapter: Session Variables ↗](#)

Validate the APM service account

Use the following bash shell command to validate the APM service account for LDAP queries. The command authenticates and queries user objects.

```
ldapsearch -xLLL -H 'ldap://192.168.0.58' -b "CN=oraclef5,dc=contoso,dc=lds" -s sub  
-D "CN=f5-apm,CN=partners,DC=contoso,DC=lds" -w 'P@55w0rd!' "(cn=testuser)"
```

Learn more:

- Go to support.f5.com for [K11072: Configuring LDAP remote authentication for AD ↗](#)
- Go to techdocs.f5.com for [Manual Chapter: LDAP Query ↗](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#) | [Get help at Microsoft Q&A](#)

Tutorial: Configure F5 BIG-IP Easy Button for SSO to Oracle JDE

Article • 10/23/2023

In this tutorial, learn to secure Oracle JD Edwards (JDE) using Microsoft Entra ID, with F5 BIG-IP Easy Button Guided Configuration.

Integrate BIG-IP with Microsoft Entra ID for many benefits:

- Improved Zero Trust governance through Microsoft Entra preauthentication and Conditional Access
 - See, [Zero Trust framework to enable remote work](#)
 - See, [What is Conditional Access?](#)
- Single sign-on (SSO) between Microsoft Entra ID and BIG-IP published services
- Manage identities and access from the [Microsoft Entra admin center](#)

Learn more:

- [Integrate F5 BIG-IP with Microsoft Entra ID](#)
- [Enable single sign-on for an enterprise application](#)

Scenario description

This tutorial uses Oracle JDE application using HTTP authorization headers to manage access to protected content.

Legacy applications lack modern protocols to support Microsoft Entra integration. Modernization is costly, requires planning, and introduces potential downtime risk. Instead, use an F5 BIG-IP Application Delivery Controller (ADC) to bridge the gap between legacy applications and modern ID control, with protocol transitioning.

With a BIG-IP in front of the app, you overlay the service with Microsoft Entra preauthentication and header-based SSO. This action improves the application's security posture.

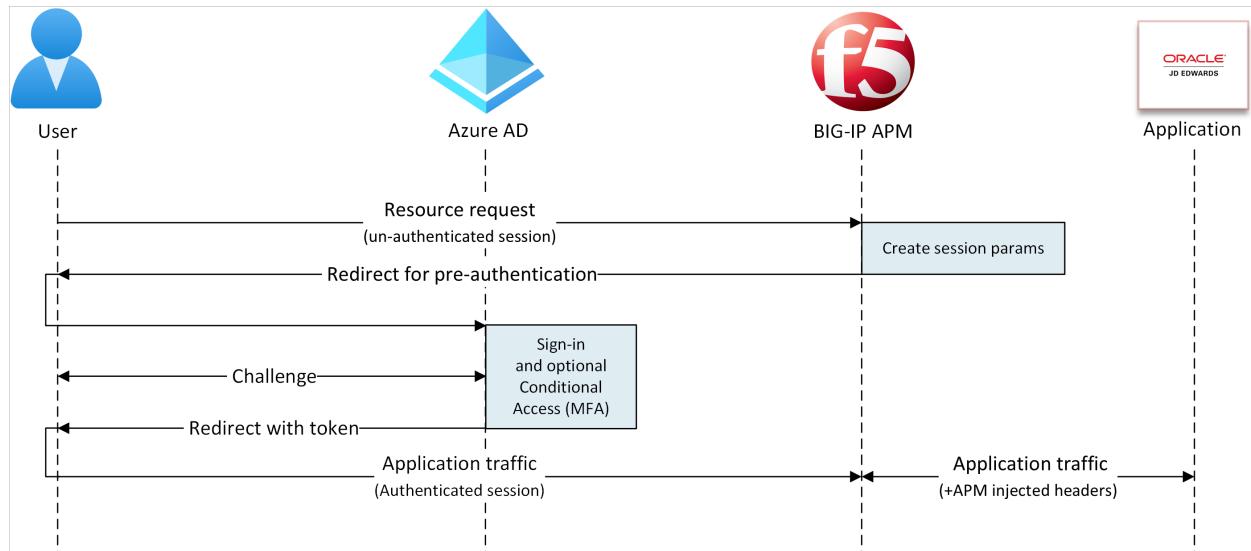
Scenario architecture

The SHA solution for this scenario is made up of several components:

- **Oracle JDE Application** - BIG-IP published service secured by Microsoft Entra SHA

- **Microsoft Entra ID** - Security Assertion Markup Language (SAML) identity provider (IdP) that verifies user credentials, Conditional Access, and SAML-based SSO to the BIG-IP
 - With SSO, Microsoft Entra ID provides session attributes to the BIG-IP
- **BIG-IP** - reverse-proxy and SAML service provider (SP) to the application
 - BIG-IP delegates authentication to the SAML IdP, then performs header-based SSO to the Oracle service

In this tutorial SHA supports SP- and IdP-initiated flows. The following diagram illustrates the SP-initiated flow.



1. User connects to application endpoint (BIG-IP).
2. BIG-IP APM access policy redirects user to Microsoft Entra ID (SAML IdP).
3. Microsoft Entra preauthenticates user and applies Conditional Access policies.
4. User is redirected to BIG-IP (SAML SP). SSO occurs using issued SAML token.
5. BIG-IP injects Microsoft Entra attributes as headers in the application request.
6. Application authorizes request and returns payload.

Prerequisites

- A Microsoft Entra ID Free account, or higher
 - If you don't have one, get an [Azure free account](#)
- A BIG-IP or a BIG-IP Virtual Edition (VE) in Azure
 - See, [Deploy F5 BIG-IP Virtual Edition VM in Azure](#)
- Any of the following F5 BIG-IP licenses:
 - F5 BIG-IP® Best bundle
 - F5 BIG-IP APM standalone license
 - F5 BIG-IP APM add-on license on an existing BIG-IP F5 BIG-IP® Local Traffic Manager™ (LTM)

- 90-day BIG-IP full feature [trial license ↗](#)
- User identities synchronized from an on-premises directory to Microsoft Entra ID, or created in Microsoft Entra ID and flowed back to the on-premises directory
 - See, [Microsoft Entra Connect Sync: Understand and customize synchronization](#)
- One of the following roles: Global Administrator, Cloud Application Administrator, or Application Administrator
- An SSL Web certificate to publish services over HTTPS, or use default BIG-IP certs for testing
 - See, [Deploy F5 BIG-IP Virtual Edition VM in Azure](#)
- An Oracle JDE environment

BIG-IP configuration

This tutorial uses Guided Configuration 16.1 with an Easy Button template. With the Easy Button, admins don't go between Microsoft Entra ID and a BIG-IP to enable services for SHA. The APM Guided Configuration wizard and Microsoft Graph handle deployment and policy management. The integration ensures applications support identity federation, SSO, and Conditional Access.

ⓘ Note

Replace example strings or values in this tutorial with those in your environment.

Register the Easy Button

💡 Tip

Steps in this article may vary slightly based on the portal you start from.

Before a client or service accesses Microsoft Graph, the Microsoft identity platform must trust it.

Learn more: [Quickstart: Register an application with the Microsoft identity platform](#)

The following instructions help you create a tenant app registration to authorize Easy Button access to Graph. With these permissions, the BIG-IP pushes the configurations to establish a trust between a SAML SP instance for published application, and Microsoft Entra ID as the SAML IdP.

1. Sign in to the Microsoft Entra admin center [↗](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > App registrations > New registration**.
3. Enter an application **Name**.
4. For **Accounts in this organizational directory only**, specify who uses the application.
5. Select **Register**.
6. Navigate to **API permissions**.
7. Authorize the following Microsoft Graph **Application permissions**:
 - Application.ReadWrite.All
 - Application.ReadWrite.OwnedBy
 - Directory.Read.All
 - Group.Read.All
 - IdentityRiskyUser.Read.All
 - Policy.Read.All
 - Policy.ReadWrite.ApplicationConfiguration
 - Policy.ReadWrite.ConditionalAccess
 - User.Read.All
8. Grant admin consent to your organization.
9. Go to **Certificates & Secrets**.
10. Generate a new **Client Secret** and note it.
11. Go to **Overview** and note the **Client ID** and **Tenant ID**

Configure the Easy Button

1. Initiate the APM Guided Configuration.
2. Launch the Easy Button template.
3. Navigate to **Access > Guided Configuration**.
4. Select **Microsoft Integration**.
5. Select **Microsoft Entra Application**.

Federation**Zero Trust****Microsoft Integration****API Protection****Microsoft Integration**

BIG-IP APM integration with Microsoft Azure AD provides secure and seamless access for all modern and classic mission-critical applications. It also provides additional security to Microsoft solutions, including Microsoft ADFS, Sharepoint, and Exchange with device posture context and adaptive MFA authentication.

ADFS Proxy

Consolidate and simplify deployments by load-balancing ADFS farm and performing ADFS proxy functionality.

Azure AD Application

Configure secure application access with Single sign-on across your hybrid identity environment by instantiating an Azure AD application template. It will setup BIG-IP APM as a SAML SP and Azure AD as an Identity Provider.

Exchange Proxy

Configure BIG-IP APM to provide proxy authentication and secure remote access to Exchange HTTP-based services.

6. Review the configuration sequence.

7. Select **Next**

Configuring the solution using the below steps will create the required objects:

	Configuration Properties Configure the Azure service account and the application settings.
	Service Provider Uniquely identify the SAML Service Provider and specify security settings.
	Azure Active Directory Select Azure application template and update configuration properties and user attributes.
	Virtual Server Provide the IP address and port for the network traffic and select a client-side SSL profile.
	Pool Configure a pool and pool members for load balancing network traffic.
	Single Sign-On (SSO) and HTTP Headers (Optional) Configure Single Sign-On properties and HTTP Headers.
	Endpoint Checks (Optional) Select client types and the endpoint inspections to perform on them.
	Session Management Configure session timeouts and user settings.

[Cancel](#) [Next](#)

8. Follow the configuration sequence.



Configuration Properties

Use the **Configuration Properties** tab to create new application configurations and SSO objects. The **Azure Service Account Details** section represents the client you registered in the Microsoft Entra tenant, as an application. Use the settings for BIG-IP OAuth client to register a SAML SP in the tenant, with SSO properties. Easy Button does this action for BIG-IP services published and enabled for SHA.

Note

Some of the following settings are global. You can reuse them to publish more applications.

1. For **Single Sign-On (SSO) & HTTP Headers**, select **On**.
2. Enter the **Tenant ID**, **Client ID**, and **Client Secret** you noted.

3. Confirm the BIG-IP connects to the tenant.

4. Select Next

Configuration Properties

General Properties ▾

Configuration Name
 OracleJDE

Type a name for this guided configuration.

Description i

On Single Sign-On (SSO) & HTTP Headers i

Endpoint Checks i

Additional Checks i

Azure Service Account Details ▾

Copy Account Info from Existing Configuration i

Tenant ID i
 4e95c47f-b591-4ced-a6b1-c8ay7b3e3946

Client ID i
 8b52b0b7-cf0e-4b9f-8f18-0a9ea3300e7e

Client Secret i

✓ Connection is valid

Application Settings ▾

Use an existing Azure application i

Service Provider

The Service Provider settings define the properties for the SAML SP instance of the application protected through SHA.

1. For **Host**, enter the public FQDN of the secured application.
2. For **Entity ID**, enter the identifier Microsoft Entra ID uses to identify the SAML SP requesting a token.

Service Provider

Advanced Settings

Service Provider Properties ▾

Host ⓘ

eportal.contoso.com

Entity ID ⓘ

https://eportal.contoso.com/

Description ⓘ

Relay State ⓘ

3. (Optional) For **Security Settings**, indicate Microsoft Entra ID encrypts issued SAML assertions. This option increases assurance that content tokens aren't intercepted, nor data compromised.
4. From the **Assertion Decryption Private Key** list, select **Create New**.

Security Settings ▾

Enable Encrypted Assertion ⓘ

Assertion Decryption Private Key ⓘ

Create New

--Select--

F5Demo	Key Type: rsa-private Last Update Time: Thu, May 6, 2021 Security Type: normal
F5DemoCert	Key Type: rsa-private Last Update Time: Fri, May 7, 2021 Security Type: normal

Create New

5. Select OK.
6. The **Import SSL Certificate and Keys** dialog appears in a new tab.
7. For **Import Type**, select **PKCS 12 (IIS)**. This option imports your certificate and private key.
8. Close the browser tab to return to the main tab.

SSL Certificate/Key Source

Import Type	PKCS 12 (IIS) <input checked="" type="button"/>
Certificate and Key Name	<input checked="" type="radio"/> New <input type="radio"/> Overwrite Existing Contoso_SAML_Cert
Certificate and Key Source	Choose File <input type="button"/> No file chosen
Password	*****
Key Security	Normal <input type="button"/>
Free Space on Disk	8022 MB

Cancel Import

9. For **Enable Encrypted Assertion**, check the box.

10. If you enabled encryption, from the **Assertion Decryption Private Key** list, select your certificate. This private key is for the certificate that BIG-IP APM uses to decrypt Microsoft Entra assertions.
11. If you enabled encryption, from the **Assertion Decryption Certificate** list, select your certificate. BIG-IP uploads this certificate to Microsoft Entra ID to encrypt issued SAML assertions.

Security Settings ▾

<input checked="" type="checkbox"/> Enable Encrypted Assertion <small>i</small>
Assertion Decryption Private Key <small>i</small>
<input type="button"/> Contoso_SAML_cert <input type="button"/>
Assertion Decryption Certificate <small>i</small>
<input type="button"/> Contoso_SAML_cert <input type="button"/>

Microsoft Entra ID

The Easy Button has templates for Oracle PeopleSoft, Oracle E-Business Suite, Oracle JD Edwards, SAP ERP and a generic SHA template.

1. Select JD Edwards Protected by F5 BIG-IP.
2. Select Add.

Azure Configuration !

User Attributes & Claims
Conditional Access Policy

Additional User Attributes

Configuration Properties ▾

Search Q



F5 BIG-IP APM
Azure AD...



Oracle
PeopleSoft -...



SAP ERP Central
Component...



Oracle E-
Business Suite...



JD Edwards -
Protected by F...

Add

Azure Configuration

1. Enter Display Name for the app BIG-IP creates in the tenant. The name appears on an icon in [My Apps](#).
2. (Optional) For Sign On URL enter the PeopleSoft application public FQDN.

Azure Configuration !

User Attributes & Claims
Conditional Access Policy

Advanced Settings

Configuration Properties ▾



JD Edwards -
Protected by F...

Change

Display Name i

Sign On URL i

3. Next to the **Signing Key** and **Signing Certificate**, select **refresh**. This action locates the certificate you imported.

4. For **Signing Key Passphrase**, enter the certificate password.

5. (Optional) For **Signing Option**, select an option. This selection ensures BIG-IP accepts tokens and claims signed by Microsoft Entra ID.

SAML Signing Certificate ▾

Signing Key ⓘ	Contoso_Wildcard_Cert	↻
Signing Certificate ⓘ	Contoso_Wildcard_Cert	↻
Signing Key Passphrase ⓘ	
Signing Option ⓘ	Sign SAML assertion	
Signing Algorithm ⓘ	RSA-SHA256	

6. **User And User Groups** are dynamically queried from the Microsoft Entra tenant.

7. Add a user or group for testing, otherwise access is denied.

User And User Groups ▾

+ Add	Remove	
<input type="checkbox"/> Name	Type ▲	Description
<input type="checkbox"/> Contoso_Personnel	User Group	Contoso full time employees

User Attributes & Claims

When a user authenticates, Microsoft Entra ID issues a SAML token with default claims and attributes identifying the user. The **User Attributes & Claims** tab has default claims to issue for the new application. Use it to configure more claims.

[Azure Configuration](#)[User Attributes & Claims](#)[Additional User Attributes](#)
[Conditional Access Policy](#)**Required Claims ▾**

Claim Name	Value
Unique User Identifier (Name ID)	user.userprincipalname
Identity	user.onpremisessamaccountname

Additional Claims ▾

Claim Name	Value	Add
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	user.mail	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	user.given...	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	user.user...	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	user.surn...	

If needed, include other Microsoft Entra attributes. The Oracle JDE scenario requires default attributes.

Additional User Attributes

The **Additional User Attributes** tab supports distributed systems that require attributes are stored in other directories for session augmentation. Attributes from an LDAP source are injected as more SSO headers to control access based on roles, Partner IDs, etc.

Azure Active Directory

[Azure Configuration](#)[User Attributes & Claims](#)[Additional User Attributes](#)[Conditional Access Policy](#)Advanced Settings **LDAP Attributes**[Cancel](#)[Save Draft](#)[Back](#)[Save & Next](#)

ⓘ Note

This feature has no correlation to Microsoft Entra ID; it's another attribute source.

Conditional Access Policy

Conditional Access policies are enforced after Microsoft Entra preauthentication to control access based on device, application, location, and risk signals. The **Available Policies** view has Conditional Access policies with no user actions. The **Selected Policies** view has policies targeting cloud apps. You can't deselect or move these policies to the Available Policies list because they're enforced at the tenant level.

Select a policy for the application.

1. In the **Available Policies** list, select a policy.
2. Select the **right arrow** and move the policy to **Selected Policies**.

Selected policies have the **Include** or **Exclude** option checked. If both options are checked, the policy isn't enforced.

The screenshot shows the Conditional Access Policy configuration screen. At the top, there are tabs: Azure Configuration, User Attributes & Claims, Additional User Attributes, and Conditional Access Policy (which is selected). Below the tabs, there is a dropdown menu for 'Conditional Access Policy'.

The main area contains two tables:

- Available Policies**: A table listing 11 policies. Most policies are labeled 'Selected' under the 'Apps' column, except for 'Breakglass' which is 'None'. The 'Name' column includes: F5 Intranet - MCAS, Intranet - Custom Control, Intranet - Block downloads, SPO - Managed endpoints only, SharePoint Restricted Content, Breakglass, Managed or Compliant endpoints, Restrict office 365 to Register..., Block all, and Block Legacy clients (Office, I...).
- Selected Policies**: A table showing 4 policies. Each row has columns for Name, Include (checkbox), Exclude (checkbox), and Apps (All). The 'Exclude' column for all four policies is checked and highlighted with a red border. The 'Name' column includes: Block legacy authentication, Term of Use, Session Controls, and MFA for all.

ⓘ Note

The policy list appears once, when you select the tab. Use **Refresh** for the wizard to query the tenant. This option appears after the application is deployed.

Virtual Server Properties

A virtual server is a BIG-IP data plane object represented by a virtual IP address. The server listens for client requests to the application. Received traffic is processed and evaluated against the virtual server APM profile. Then, traffic is directed according to policy.

1. For **Destination Address**, enter the IPv4 or IPv6 address BIG-IP uses to receive client traffic. A corresponding record appears in DNS, which enables clients to resolve the published application's external URL to the IP. Use a test computer localhost DNS for testing.
2. For **Service Port**, enter **443** and select **HTTPS**.
3. For **Enable Redirect Port**, check the box.
4. For **Redirect Port**, enter **80** and select **HTTP**. This option redirects incoming HTTP client traffic to HTTPS.
5. For **Client SSL Profile**, select **Use Existing**.
6. Under **Common** select the option you created. If testing, leave the default. Client SSL Profile enables the virtual server for HTTPS, so client connections are encrypted over TLS.

Virtual Server Properties

Advanced Settings

General Properties ▾

Virtual Server
 Create New Use Existing

Destination Address ⓘ
172.16.76.27

Service Port ⓘ
443 HTTPS ▾

Enable Redirect Port ⓘ

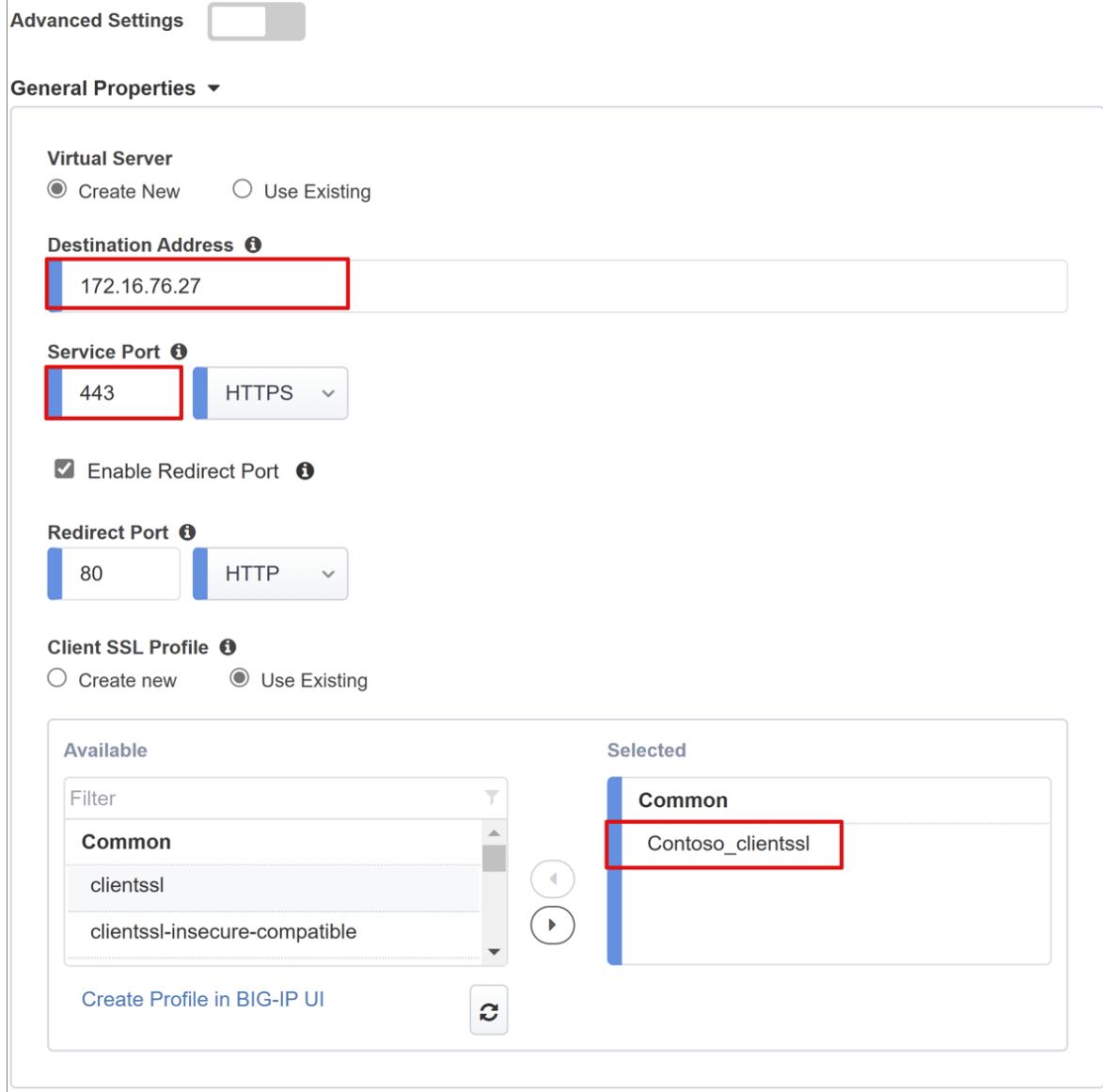
Redirect Port ⓘ
80 HTTP ▾

Client SSL Profile ⓘ
 Create new Use Existing

Available
Filter
Common
clientssl
clientssl-insecure-compatible

Selected
Common
Contoso_clientssl

Create Profile in BIG-IP UI



Pool Properties

The **Application Pool** tab has services behind a BIG-IP, represented as a pool with application servers.

1. For **Select a Pool**, select **Create New**, or select one.
2. For **Load Balancing Method**, select **Round Robin**.
3. For **Pool Servers**, in **IP Address/Node Name** select a node, or enter an IP and port for servers hosting the Oracle JDE application.

Application Pool ▾

Select a Pool
Create New

Select an existing pool or select Create New.

Resources Properties

Load Balancing Method ⓘ
Round Robin

Pool Servers ⓘ

IP Address/Node name	Port	Priority Group	Action
172.16.76.16	80	HTTP	0

+ **x**

Single Sign-On & HTTP Headers

The Easy Button wizard supports Kerberos, OAuth Bearer, and HTTP authorization headers for SSO to published applications. The PeopleSoft application expects headers.

1. For **HTTP Headers**, check the box.
2. For **Header Operation**, select **replace**.
3. For **Header Name**, enter **JDE_SSO_UID**.
4. For **Header Value**, enter **%{session.sso.token.last.username}**.

Single Sign-On & HTTP Headers

Single Sign-On

HTTP Headers ▾

SSO Headers ⓘ

Header Operation	Header Name	Header Value	Delimiter	Action
replace	JDE_SSO_UID	%{session.sso.token.last.username}		+ x

! Note

APM session variables in curly brackets are case-sensitive. For instance, if you enter **OrclGUID**, and the attribute name is **orclguid**, attribute mapping fails.

Session Management

Use BIG-IP Session Management settings to define conditions for user sessions termination or continuation. Set limits for users and IP addresses, and corresponding user info.

To learn more, go to support.f5.com for [K18390492: Security | BIG-IP APM operations guide](#)

Not covered in the operations guide is single log-out (SLO) functionality, which ensures IdP, BIG-IP, and user agent sessions terminate when users sign out. When the Easy Button instantiates a SAML application in the Microsoft Entra tenant, it populates the Logout URL with the APM SLO endpoint. IdP-initiated sign out from [My Apps](#) terminates BIG-IP and client sessions.

Published-application SAML federation data is imported from the tenant. This action provides the APM with the SAML sign out endpoint for Microsoft Entra ID, which ensures SP-initiated sign out terminates client and Microsoft Entra sessions. The APM needs to know when a user signs out.

When the BIG-IP webtop portal accesses published applications, the APM processes a sign out to call the Microsoft Entra sign-out endpoint. If the BIG-IP webtop portal isn't used, the user can't instruct the APM to sign out. If the user signs out of the application, the BIG-IP is oblivious. SP-initiated sign out needs secure session termination. Add an SLO function to your application **Sign out** button, to redirect your client to the Microsoft Entra SAML or BIG-IP sign out endpoint. The SAML sign out endpoint URL for your tenant in **App Registrations > Endpoints**.

If you can't change the app, consider having the BIG-IP listen for application sign out calls, and then trigger SLO.

Learn more: [Tutorial: Configure F5 BIG-IP Easy Button for SSO to Oracle PeopleSoft, PeopleSoft Single Logout](#)

To learn more, go to support.f5.com for:

- [K42052145: Configuring automatic session termination \(logout\) based on a URI-referenced file name](#)
- [K12056: Overview of the Logout URI Include option](#).

Deployment

1. Select Deploy.

2. Verify the application is in the tenant list of Enterprise applications.

Confirm configuration

1. Using a browser, connect to the Oracle JDE application's external URL or select the application icon in [My Apps](#).
2. Authenticate to Microsoft Entra ID.
3. You're redirected to the BIG-IP virtual server for the application and signed in with SSO.

 Note

You can block direct access to the application, thereby enforcing a path through the BIG-IP.

Advanced deployment

Sometimes, the Guided Configuration templates lack flexibility.

Learn more: [Tutorial: Configure F5 BIG-IP Access Policy Manager for header-based SSO](#)

Alternatively, in BIG-IP disable the Guided Configuration strict management mode. You can manually change configurations, although most configurations are automated with wizard templates.

1. Navigate to **Access > Guided Configuration**.
2. At the end of the row, select the **padlock**.



Changes with the wizard UI aren't possible, but BIG-IP objects associated with the application published instance are unlocked for management.

 Note

When you reenable strict mode and deploy a configuration, settings performed outside Guided Configuration are overwritten. We recommend advanced configuration for production services.

Troubleshooting

Use BIG-IP logging to isolate issues with connectivity, SSO, policy violations, or misconfigured variable mappings.

Log verbosity

1. Navigate to **Access Policy > Overview**.
2. Select **Event Logs**.
3. Select **Settings**.
4. Select the row of your published application.
5. Select **Edit**.
6. Select **Access System Logs**
7. From the SSO list, select **Debug**.
8. Select **OK**.
9. Reproduce your issue.
10. Inspect the logs.

When complete, revert this feature because verbose mode generates lots of data.

BIG-IP error message

If a BIG-IP error appears after Microsoft Entra preauthentication, it's possible the issue relates to Microsoft Entra ID to BIG-IP SSO.

1. Navigate to **Access > Overview**.
2. Select **Access reports**.
3. Run the report for the last hour.
4. Review the logs for clues.

Use the session's **View session** link to confirm the APM receives expected Microsoft Entra claims.

No BIG-IP error message

If no BIG-IP error message appears, the issue might be related to the back-end request, or BIG-IP to application SSO.

1. Navigate to **Access Policy > Overview**.
2. Select **Active Sessions**.
3. Select the active session link.

Use the **View Variables** link to determine SSO issues, particularly if BIG-IP APM obtains incorrect attributes from session variables.

Learn more:

- Go to devcentral.f5.com for [APM variable assign examples ↗](#)
 - Go to techdocs.f5.com for [Session Variables ↗](#)
-

Feedback

Was this page helpful?



[Provide product feedback ↗](#) | [Get help at Microsoft Q&A](#)

Tutorial: Configure F5 BIG-IP Easy Button for SSO to Oracle PeopleSoft

Article • 10/23/2023

In this article, learn to secure Oracle PeopleSoft (PeopleSoft) using Microsoft Entra ID, with F5 BIG-IP Easy Button Guided Configuration 16.1.

Integrate BIG-IP with Microsoft Entra ID for many benefits:

- Improved Zero Trust governance through Microsoft Entra preauthentication and Conditional Access
 - See, [Zero Trust framework to enable remote work](#)
 - See, [What is Conditional Access?](#)
- Single sign-on (SSO) between Microsoft Entra ID and BIG-IP published services
- Manage identities and access from the [Microsoft Entra admin center](#)

Learn more:

- [Integrate F5 BIG-IP with Microsoft Entra ID](#)
- [Enable single sign-on for an enterprise application](#)

Scenario description

For this tutorial, there's use of a PeopleSoft application using HTTP authorization headers to manage access to protected content.

Legacy applications lack modern protocols to support Microsoft Entra integration. Modernization is costly, requires planning, and introduces potential downtime risk. Instead, use an F5 BIG-IP Application Delivery Controller (ADC) to bridge the gap between legacy applications and modern ID control, with protocol transitioning.

With a BIG-IP in front of the app, you overlay the service with Microsoft Entra preauthentication and header-based SSO. This action improves the application's security posture.

Note

Gain remote access to this type of application with Microsoft Entra application proxy.

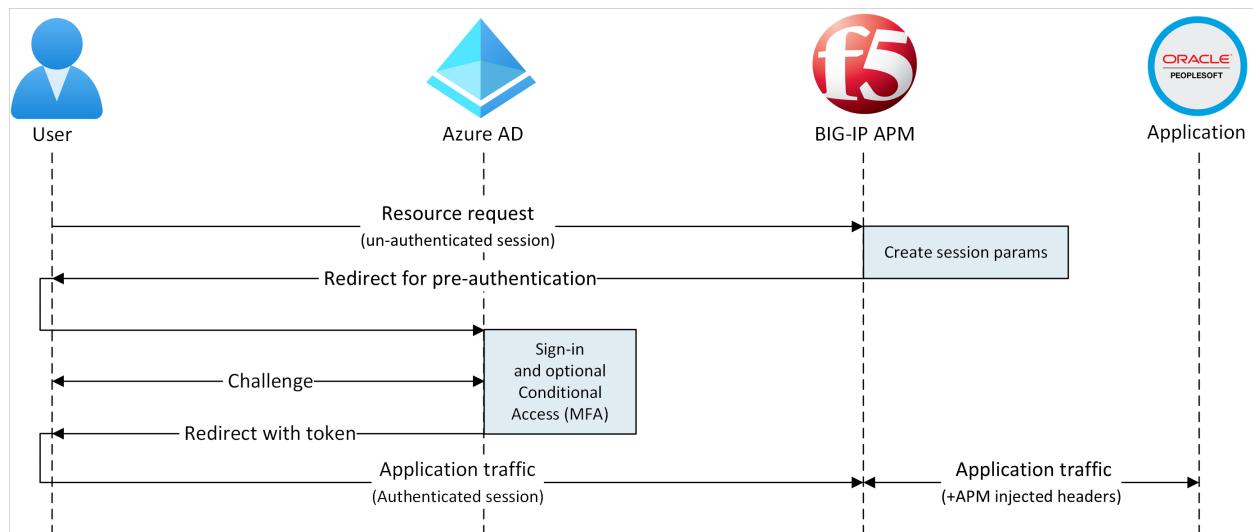
See, [Remote access to on-premises applications through Microsoft Entra application proxy](#).

Scenario architecture

The secure hybrid access (SHA) solution for this tutorial has the following components:

- **PeopleSoft Application** - BIG-IP published service secured by Microsoft Entra SHA
- **Microsoft Entra ID** - Security Assertion Markup Language (SAML) identity provider (IdP) that verifies user credentials, Conditional Access, and SAML-based SSO to the BIG-IP
 - Through SSO, Microsoft Entra ID provides session attributes to the BIG-IP
- **BIG-IP** - reverse-proxy and SAML service provider (SP) to the application. It delegates authentication to the SAML IdP, then performs header-based SSO to the PeopleSoft service.

For this scenario, SHA supports SP- and IdP-initiated flows. The following diagram illustrates the SP-initiated flow.



1. User connects to application endpoint (BIG-IP).
2. BIG-IP APM access policy redirects user to Microsoft Entra ID (SAML IdP).
3. Microsoft Entra preauthenticates user and applies Conditional Access policies.
4. User is redirected to BIG-IP (SAML SP) and SSO occurs with issued SAML token.
5. BIG-IP injects Microsoft Entra attributes as headers in the request to the application.
6. Application authorizes request and returns payload.

Prerequisites

- A Microsoft Entra ID Free account, or higher
 - If you don't have one, get an [Azure free account](#)
- A BIG-IP or a BIG-IP Virtual Edition (VE) in Azure
 - See, [Deploy F5 BIG-IP Virtual Edition VM in Azure](#)
- Any of the following F5 BIG-IP licenses:
 - F5 BIG-IP® Best bundle
 - F5 BIG-IP APM standalone license
 - F5 BIG-IP APM add-on license on an existing BIG-IP F5 BIG-IP® Local Traffic Manager™ (LTM)
 - 90-day BIG-IP full feature [trial license](#)
- User identities synchronized from an on-premises directory to Microsoft Entra ID, or created in Microsoft Entra ID and flowed back to the on-premises directory
 - See, [Microsoft Entra Connect Sync: Understand and customize synchronization](#)
- One of the following roles: Global Administrator, Cloud Application Administrator, or Application Administrator.
- An SSL Web certificate to publish services over HTTPS, or use default BIG-IP certs for testing
 - See, [Deploy F5 BIG-IP Virtual Edition VM in Azure](#)
- A PeopleSoft environment

BIG-IP configuration

This tutorial uses Guided Configuration 16.1 with an Easy button template.

With the Easy Button, admins don't go between Microsoft Entra ID and a BIG-IP to enable services for SHA. APM Guided Configuration wizard and Microsoft Graph handle deployment and policy management. The integration ensures applications support identity federation, SSO, and Conditional Access.

Note

Replace example strings or values in this tutorial with those in your environment.

Register the Easy Button

Tip

Steps in this article may vary slightly based on the portal you start from.

Before a client or service accesses Microsoft Graph, the Microsoft identity platform must trust it.

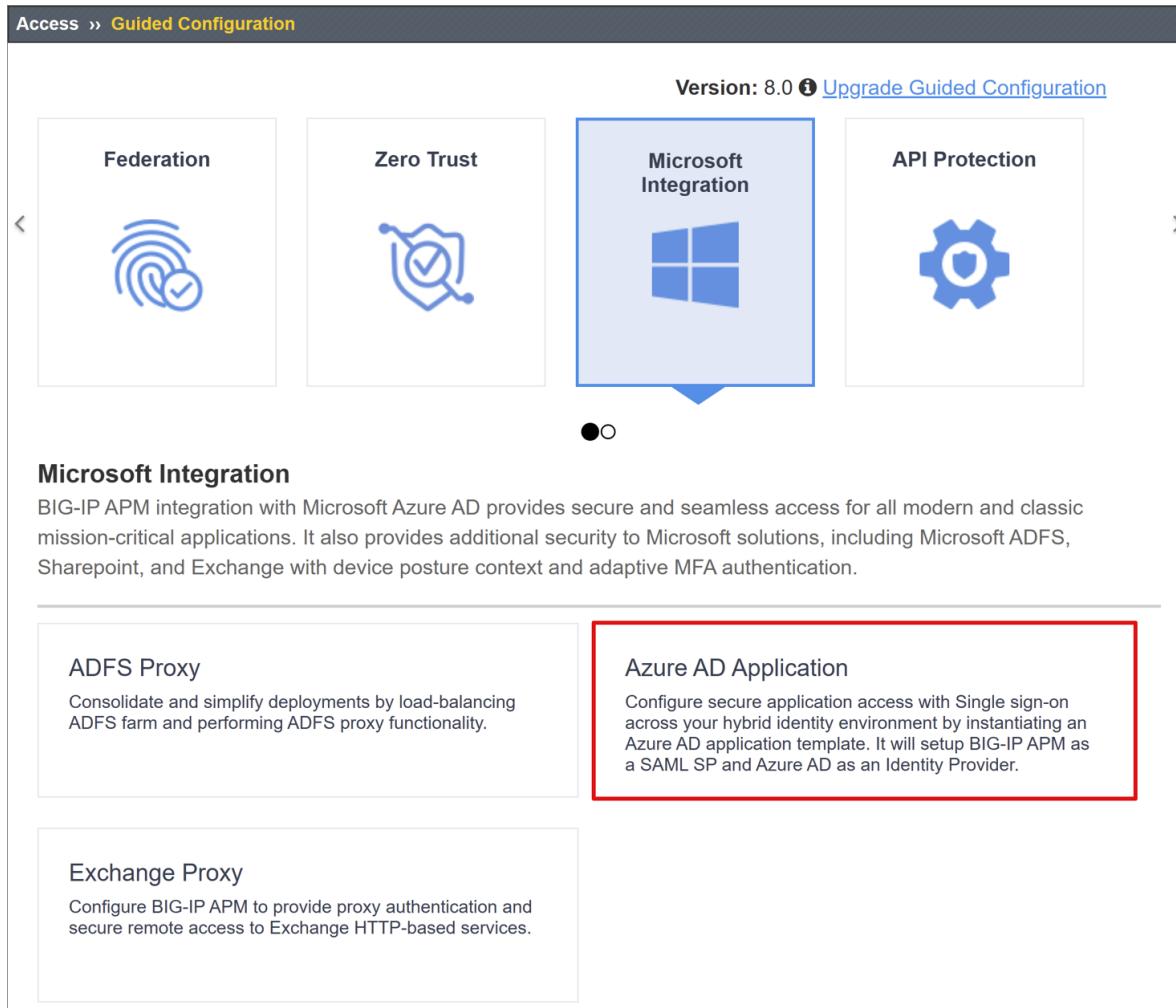
Learn more: [Quickstart: Register an application with the Microsoft identity platform](#)

The following instructions help you create a tenant app registration to authorize Easy Button access to Graph. With these permissions, the BIG-IP pushes the configurations to establish a trust between a SAML SP instance for published application, and Microsoft Entra ID as the SAML IdP.

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > App registrations > New registration**.
3. Enter an application **Name**.
4. For **Accounts in this organizational directory only**, specify who uses the application.
5. Select **Register**.
6. Navigate to **API permissions**.
7. Authorize the following Microsoft Graph **Application permissions**:
 - Application.ReadWrite.All
 - Application.ReadWrite.OwnedBy
 - Directory.Read.All
 - Group.Read.All
 - IdentityRiskyUser.Read.All
 - Policy.Read.All
 - Policy.ReadWrite.ApplicationConfiguration
 - Policy.ReadWrite.ConditionalAccess
 - User.Read.All
8. Grant admin consent to your organization.
9. Go to **Certificates & Secrets**.
10. Generate a new **Client Secret** and note it.
11. Go to **Overview** and note the **Client ID** and **Tenant ID**.

Configure the Easy Button

1. Initiate the APM Guided Configuration.
2. Launch the Easy Button template.
3. Navigate to Access > Guided Configuration.
4. Select Microsoft Integration.
5. Select Microsoft Entra Application.



The screenshot shows the 'Access > Guided Configuration' screen. At the top, it displays 'Version: 8.0' and a link to 'Upgrade Guided Configuration'. Below this, there are four cards representing different integration options: 'Federation' (fingerprint icon), 'Zero Trust' (shield with checkmark icon), 'Microsoft Integration' (Windows logo icon, highlighted with a blue border), and 'API Protection' (gear and shield icon). The 'Microsoft Integration' card has a blue arrow pointing down to its detailed description section. The section title is 'Microsoft Integration'. The text explains: 'BIG-IP APM integration with Microsoft Azure AD provides secure and seamless access for all modern and classic mission-critical applications. It also provides additional security to Microsoft solutions, including Microsoft ADFS, Sharepoint, and Exchange with device posture context and adaptive MFA authentication.' Below this, there are three other configuration steps: 'ADFS Proxy' (described as consolidating and simplifying ADFS farm deployments), 'Azure AD Application' (described as configuring secure application access with Single sign-on across hybrid identity environments, with this step highlighted by a red box), and 'Exchange Proxy' (described as providing proxy authentication and secure remote access to Exchange services).

6. Review the configuration sequence.

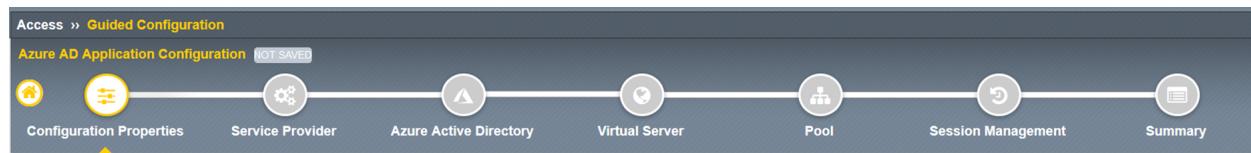
7. Select Next

Configuring the solution using the below steps will create the required objects:

	Configuration Properties Configure the Azure service account and the application settings.
	Service Provider Uniquely identify the SAML Service Provider and specify security settings.
	Azure Active Directory Select Azure application template and update configuration properties and user attributes.
	Virtual Server Provide the IP address and port for the network traffic and select a client-side SSL profile.
	Pool Configure a pool and pool members for load balancing network traffic.
	Single Sign-On (SSO) and HTTP Headers (Optional) Configure Single Sign-On properties and HTTP Headers.
	Endpoint Checks (Optional) Select client types and the endpoint inspections to perform on them.
	Session Management Configure session timeouts and user settings.

[Cancel](#) [Next](#)

8. Follow the configuration sequence.



Configuration Properties

Use the **Configuration Properties** tab to create new application configurations and SSO objects. The **Azure Service Account Details** section represents the client you registered in the Microsoft Entra tenant, as an application. Use the settings for BIG-IP OAuth client to register a SAML SP in the tenant, with SSO properties. Easy Button does this action for BIG-IP services published and enabled for SHA.

Note

Some of the following settings are global. You can reuse them to publish more applications.

1. Enter a **Configuration Name**. Unique names help distinguish configurations.
2. For **Single Sign-On (SSO) & HTTP Headers**, select **On**.
3. Enter the **Tenant ID**, **Client ID**, and **Client Secret** you noted.

4. Confirm the BIG-IP connects to the tenant.

5. Select **Next**.

Configuration Properties

General Properties ▾

Configuration Name

PeopleSoft

Type a name for this guided configuration.

Description ⓘ

On Single Sign-On (SSO) & HTTP Headers ⓘ

Endpoint Checks ⓘ

Additional Checks ⓘ

Azure Service Account Details ▾

Copy Account Info from Existing Configuration ⓘ

Tenant ID ⓘ

4e95c47f-b591-4ced-a1b1-c6ac7b3a9426

Client ID ⓘ

8b52b0b7-cf0e-4b9f-8f18-0a9ea3300e7e

Client Secret ⓘ

Test Connection

Connection is valid

Application Settings ▾

Use an existing Azure application ⓘ

Service Provider

Use the **Service Provider** settings to define SAML SP properties for the APM instance that represents the SHA-secured application.

1. For **Host**, enter the public FQDN of the secured application.

2. For **Entity ID**, enter the identifier Microsoft Entra ID uses to identify the SAML SP requesting a token.

Service Provider

Advanced Settings

Service Provider Properties ▾

Host ⓘ

Entity ID ⓘ

Description ⓘ

Relay State ⓘ

3. (Optional) For **Security Settings**, indicate Microsoft Entra ID encrypts issued SAML assertions. This option increases assurance that content tokens aren't intercepted, nor data compromised.
4. From the **Assertion Decryption Private Key** list, select **Create New**.

Security Settings ▾

Enable Encrypted Assertion ⓘ

Assertion Decryption Private Key ⓘ

--Select--

F5Demo
Key Type: rsa-private
Last Update Time: Thu, May 6, 2021
Security Type: normal

F5DemoCert
Key Type: rsa-private
Last Update Time: Fri, May 7, 2021
Security Type: normal

5. Select **OK**.
6. The **Import SSL Certificate and Keys** dialog appears in a new tab.
7. For **Import Type**, select **PKCS 12 (IIS)**. This option imports your certificate and private key.

8. Close the browser tab to return to the main tab.

System » Certificate Management : Traffic Certificate Management : SSL Certificate List » Import SSL Certificates and Keys

SSL Certificate/Key Source

Import Type	PKCS 12 (IIS) <input style="border: 1px solid red;" type="button" value="..."/>
Certificate and Key Name	<input checked="" type="radio"/> New <input type="radio"/> Overwrite Existing Contoso_SAML_Cert <input style="border: 2px solid red;" type="text"/>
Certificate and Key Source	Choose File No file chosen
Password	<input style="border: 1px solid red;" type="password"/>
Key Security	Normal <input style="border: 1px solid gray;" type="button" value="..."/>
Free Space on Disk	8022 MB

9. For **Enable Encrypted Assertion**, check the box.

10. If you enabled encryption, from the **Assertion Decryption Private Key** list, select your certificate. This private key is for the certificate that BIG-IP APM uses to decrypt Microsoft Entra assertions.
11. If you enabled encryption, from the **Assertion Decryption Certificate** list, select your certificate. BIG-IP uploads this certificate to Microsoft Entra ID to encrypt issued SAML assertions.

Security Settings ▾

<input checked="" type="checkbox"/> Enable Encrypted Assertion <small>i</small>
Assertion Decryption Private Key <small>i</small> <input style="border: 2px solid red;" type="button" value="Contoso_SAML_cert"/> <input style="border: 1px solid gray;" type="button" value="..."/>
Assertion Decryption Certificate <small>i</small> <input style="border: 2px solid red;" type="button" value="Contoso_SAML_cert"/> <input style="border: 1px solid gray;" type="button" value="..."/>

Microsoft Entra ID

The Easy Button has templates for Oracle PeopleSoft, Oracle E-Business Suite, Oracle JD Edwards, SAP ERP and a generic SHA template.

1. Select **Oracle PeopleSoft**.
2. Select **Add**.

Azure Configuration !

User Attributes & Claims Additional User Attributes

Conditional Access Policy

Configuration Properties ▾

Search

F5 BIG-IP APM
Azure AD...

Oracle
PeopleSoft -...

SAP ERP Central
Component...

Oracle E-
Business Suite...

JD Edwards -
Protected by F...

Add

Azure Configuration

1. Enter Display Name for the app BIG-IP creates in the tenant. The name appears on an icon in [My Apps](#).
2. (Optional) For Sign On URL enter the PeopleSoft application public FQDN.

Azure Configuration !

User Attributes & Claims Additional User Attributes

Conditional Access Policy

Advanced Settings OFF

Configuration Properties ▾

Oracle
PeopleSoft -...

Change

Display Name i

PeopleSoft

Sign On URL i

https://eportal.contoso.com

3. Next to the **Signing Key** and **Signing Certificate**, select **refresh**. This action locates the certificate you imported.

4. For **Signing Key Passphrase**, enter the certificate password.

5. (Optional) For **Signing Option**, select an option. This selection ensures BIG-IP accepts tokens and claims signed by Microsoft Entra ID.

SAML Signing Certificate ▾

Signing Key ⓘ	Contoso_Wildcard_Cert	↻
Signing Certificate ⓘ	Contoso_Wildcard_Cert	↻
Signing Key Passphrase ⓘ	
Signing Option ⓘ	Sign SAML assertion	
Signing Algorithm ⓘ	RSA-SHA256	

6. **User And User Groups** are dynamically queried from the Microsoft Entra tenant.

7. Add a user or group for testing, otherwise access is denied.

User And User Groups ▾

+ Add	Remove	
<input type="checkbox"/> Name	Type ▾	Description
<input type="checkbox"/> Contoso_Personnel	User Group	Contoso full time employees

User Attributes & Claims

When a user authenticates, Microsoft Entra ID issues a SAML token with default claims and attributes identifying the user. The **User Attributes & Claims** tab has default claims to issue for the new application. Use it to configure more claims. The Easy Button template has the employee ID claim required by PeopleSoft.

Azure Active Directory

Azure Configuration ⚠ User Attributes & Claims Additional User Attributes Conditional Access Policy

Required Claims ▾

Claim Name	Value
Unique User Identifier (Name ID)	user.userprincipalname

Additional Claims ▾

Claim Name	Value	Add
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	user.mail	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	user.givenname	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	user.userprinci...	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	user.surname	
EMPLID	user.employeeid	

If needed, include other Microsoft Entra attributes. The sample PeopleSoft application requires predefined attributes.

Additional User Attributes

The **Additional User Attributes** tab supports distributed systems that require attributes are stored in other directories for session augmentation. Attributes from an LDAP source are injected as more SSO headers to control access based on roles, Partner IDs, etc.

Azure Active Directory

The screenshot shows the 'Additional User Attributes' tab selected in the top navigation bar. Below it, there's an 'Advanced Settings' section with a toggle switch. A note indicates that LDAP Attributes are not selected. At the bottom are standard navigation buttons: 'Cancel', 'Save Draft', 'Back', and 'Save & Next'.

① Note

This feature has no correlation to Microsoft Entra ID; it's another attribute source.

Conditional Access Policy

Conditional Access policies are enforced after Microsoft Entra preauthentication to control access based on device, application, location, and risk signals. The **Available Policies** view has Conditional Access policies with no user actions. The **Selected Policies** view has policies targeting cloud apps. You can't deselect or move these policies to the Available Policies list because they're enforced at the tenant level.

Select a policy for the application.

1. In the **Available Policies** list, select a policy.
2. Select the **right arrow** and move the policy to **Selected Policies**.

Selected policies have the **Include** or **Exclude** option checked. If both options are checked, the policy isn't enforced.

Azure Configuration User Attributes & Claims Additional User Attributes **Conditional Access Policy**

Conditional Access Policy ▾

[View Conditional Access policies in the Azure Portal](#)

Available Policies ⓘ		Selected Policies ⓘ		
Items: 11	Filter by Name...	Items: 4	Filter by Name...	
□ Name	Apps	□ Name	Include	Exclude
<input type="checkbox"/> F5 Intranet - MCAS	Selected	<input type="checkbox"/> Block legacy authenticat...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> Intranet - Custom Control	Selected	<input type="checkbox"/> Term of Use	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> Intranet - Block downloads	Selected	<input type="checkbox"/> Session Controls	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> SPO - Managed endpoints only	Selected	<input type="checkbox"/> MFA for all	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> SharePoint Restricted Content	Selected			
<input type="checkbox"/> Breakglass	None			
<input type="checkbox"/> Managed or Compliant endpoints	None			
<input type="checkbox"/> Restrict office 365 to Register...	Selected			
<input type="checkbox"/> Block all	Selected			
<input type="checkbox"/> Block Legacy clients (Office, I...	Selected ▾			

ⓘ Note

The policy list appears once, when you select the tab. Use **Refresh** for the wizard to query the tenant. This option appears after the application is deployed.

Virtual Server Properties

A virtual server is a BIG-IP data plane object represented by a virtual IP address. The server listens for client requests to the application. Received traffic is processed and evaluated against the virtual server APM profile. Then, traffic is directed according to policy.

1. For **Destination Address**, enter the IPv4 or IPv6 address BIG-IP uses to receive client traffic. A corresponding record appears in DNS, which enables clients to resolve the published application's external URL to the IP. Use a test computer localhost DNS for testing.
2. For **Service Port**, enter **443** and select **HTTPS**.
3. For **Enable Redirect Port**, check the box.
4. For **Redirect Port**, enter **80** and select **HTTP**. This option redirects incoming HTTP client traffic to HTTPS.
5. For **Client SSL Profile**, select **Use Existing**.

6. Under **Common** select the option you created. If testing, leave the default. Client SSL Profile enables the virtual server for HTTPS, so client connections are encrypted over TLS.

Virtual Server Properties

Advanced Settings

General Properties ▾

Virtual Server
 Create New Use Existing

Destination Address

Service Port

Enable Redirect Port

Client SSL Profile Create new Use Existing

Available	Selected
Common	Common
clientssl	Contoso_clientssl
clientssl-insecure-compatible	

[Create Profile in BIG-IP UI](#)

Pool Properties

The **Application Pool** tab has services behind a BIG-IP, represented as a pool with application servers.

1. For **Select a Pool**, select **Create New**, or select one.
2. For **Load Balancing Method**, select **Round Robin**.
3. For **Pool Servers**, in **IP Address/Node Name** select a node, or enter an IP and port for servers hosting the PeopleSoft application.

Pool Properties

Advanced Settings

Application Pool ▾

Select a Pool

Create New

Select an existing pool or select Create New.

Resources Properties

Load Balancing Method ⓘ

Round Robin

Pool Servers ⓘ

IP Address/Node name	Port	Priority Group	Action
172.16.74.12	443	HTTPS	<input type="button" value="+"/> <input type="button" value="X"/>

Single sign-on & HTTP Headers

The Easy Button wizard supports Kerberos, OAuth Bearer, and HTTP authorization headers for SSO to published applications. The PeopleSoft application expects headers.

1. For **HTTP Headers**, check the box.
2. For **Header Operation**, select replace.
3. For **Header Name**, enter PS_SSO_UID.
4. For **Header Value**, enter %{session.sso.token.last.username}.

Single Sign-On & HTTP Headers

Single Sign-On

HTTP Headers ▾

SSO Headers ⓘ

Header Operation	Header Name	Header Value	Delimiter	Action
replace	PS_SSO_UID	%{session.saml.last.attr.name.EMPLID}	<input type="button" value="+"/> <input type="button" value="X"/>	

ⓘ Note

APM session variables in curly brackets are case-sensitive. For instance, if you enter OrclGUID, and the attribute name is orclguid, attribute mapping fails.

Session Management

Use BIG-IP session management settings to define conditions for user sessions termination or continuation. Set limits for users and IP addresses, and corresponding user info.

To learn more, go to support.f5.com for [K18390492: Security | BIG-IP APM operations guide](#)

Not covered in the operations guide is single log-out (SLO) functionality, which ensures IdP, BIG-IP, and user agent sessions terminate when users sign out. When the Easy Button instantiates a SAML application in the Microsoft Entra tenant, it populates the Logout URL with the APM SLO endpoint. IdP-initiated sign out from [My Apps](#) terminates BIG-IP and client sessions.

Published-application SAML federation data is imported from the tenant. This action provides the APM with the SAML sign-out endpoint for Microsoft Entra ID, which ensures SP-initiated sign out terminates client and Microsoft Entra sessions. The APM needs to know when a user signs out.

When the BIG-IP webtop portal accesses published applications, the APM processes a sign out to call the Microsoft Entra sign-out endpoint. If the BIG-IP webtop portal isn't used, the user can't instruct the APM to sign out. If the user signs out of the application, the BIG-IP is oblivious. SP-initiated sign out needs secure session termination. Add an SLO function to your application **Sign out** button, to redirect your client to the Microsoft Entra SAML or BIG-IP sign out endpoint. The SAML sign out endpoint URL for your tenant in **App Registrations > Endpoints**.

If you can't change the app, consider having the BIG-IP listen for application sign out calls, and trigger SLO. For more information, see **PeopleSoft Single Logout** in the following section.

Deployment

1. Select **Deploy**.
2. Verify the application in the tenant list of Enterprise applications.
3. The application is published and accessible with SHA.

Configure PeopleSoft

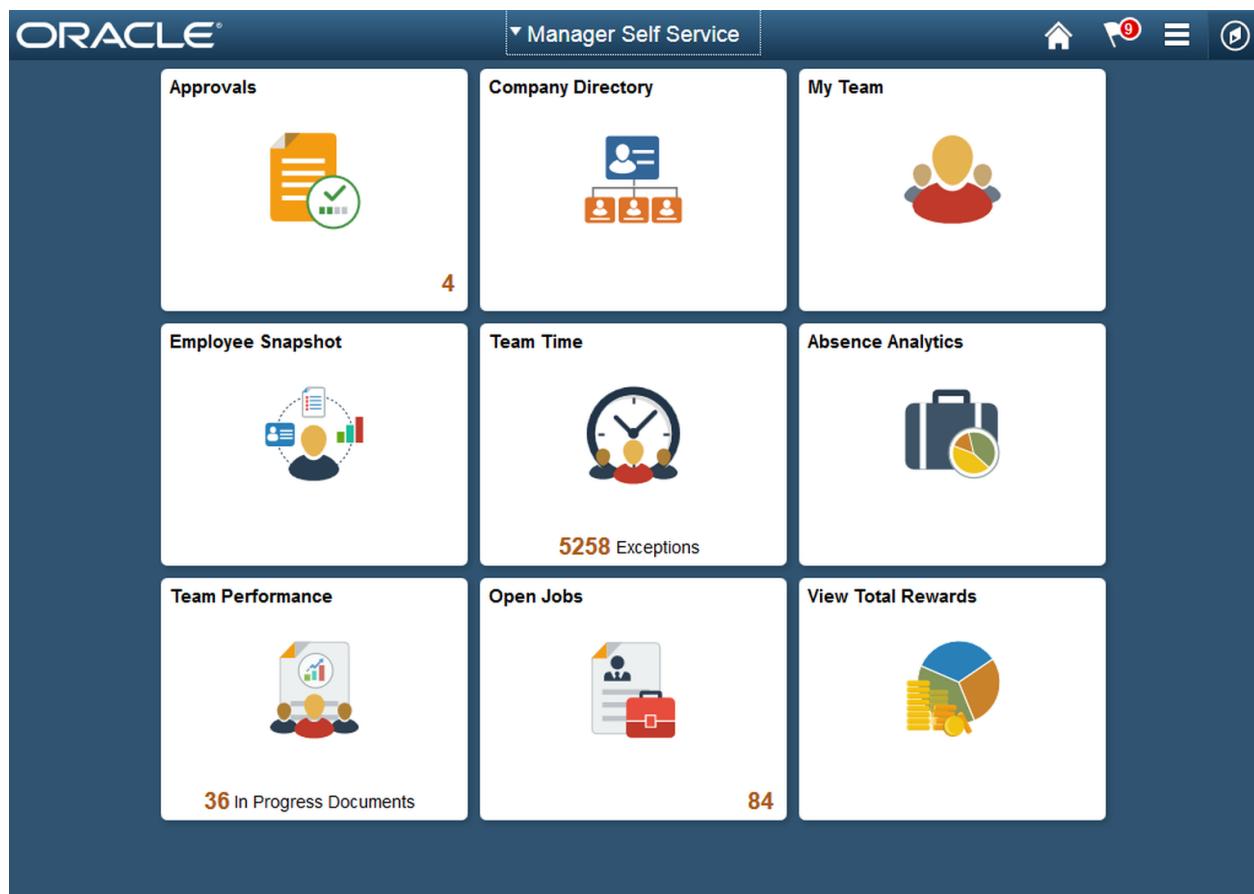
Use Oracle Access Manager for PeopleSoft application identity and access management.

To learn more, go to [docs.oracle.com](#) for [Oracle Access Manager Integration Guide, Integrating PeopleSoft](#).

Configure Oracle Access Manager SSO

Configure Oracle Access Manager to accept SSO from the BIG-IP.

1. Sign into the Oracle console with admin permissions.



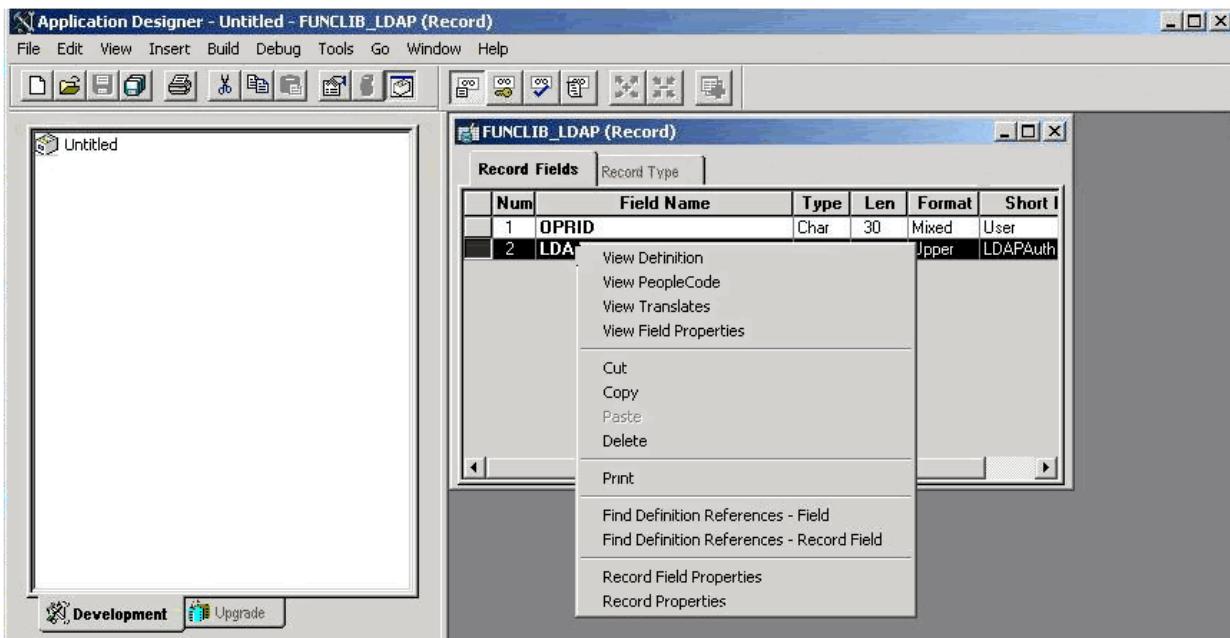
2. Navigate to **PeopleTools > Security**.
3. Select **User Profiles**.
4. Select **User Profiles**.
5. Create a new user profile.
6. For **User ID**, enter **OAMPSFT**
7. For **User Role**, enter **PeopleSoft User**.
8. Select **Save**.

The screenshot shows the 'User Profiles' screen in the Oracle PeopleSoft application. At the top, there's a navigation bar with links for 'General', 'ID', 'Roles', 'Workflow', 'Audit', 'Links', and 'User ID Queries'. A red box highlights the 'User ID' field, which contains 'OAMPSFT'. Below the navigation bar is a search bar with the placeholder 'User ID OAMPSFT'. To the right of the search bar is a 'Description' section. On the left, there's a 'Dynamic Role Rule' panel with buttons for 'Test Rule(s)', 'Refresh', 'Execute Rule(s)', 'Process Monitor', and 'Service Monitor'. To the right of this panel is a 'User Roles' table with one row: 'Role Name: PeopleSoft User' and 'Description: PeopleSoft User'. Below the table are buttons for 'Save', 'Return to Search', 'Add', and 'Update/Display'. At the bottom of the page are links for 'General', 'ID', 'Roles', 'Workflow', 'Audit', 'Links', and 'User ID Queries'.

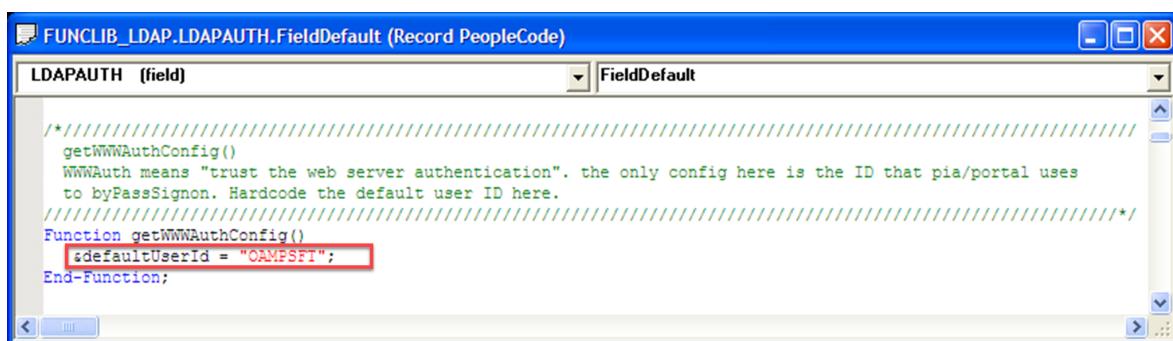
9. Navigate to **People Tools > Web Profile**.
10. Select the web profile.
11. On **Security** tab, in **Public Users**, select **Allow Public Access**.
12. For **User ID**, enter **OAMPSFT**.
13. Enter the **Password**.

The screenshot shows the 'Web Profile Configuration' screen in the Oracle PeopleTools application. The top navigation bar includes links for 'General', 'Security', 'Authorized Site', 'Virtual Addressing', 'Cookie Rules', 'Caching', 'Debugging', and 'Look and Feel'. The main configuration area has a 'Profile Name' set to 'PROD'. Under the 'Security' tab, the 'Public Users' section is highlighted with a red box. It contains fields for 'Allow Public Access' (checked), 'User ID' (set to 'OAMPSFT'), 'Password' (redacted), and 'HTTP Session Inactivity' (set to 0 seconds). Other sections visible include 'Authenticated Users' (with 'Inactivity Warning' at 1,080 seconds and 'Inactivity Logout' at 1,200 seconds) and 'Web Server Jolt Settings' (with 'Disconnect Timeout', 'Send Timeout', and 'Receive Timeout' all set to 0 seconds). There's also an 'XML Link' section with 'User ID' (set to 'PS') and 'Password' (redacted). At the bottom are buttons for 'Save', 'Return to Search', 'Notify', 'Previous tab', 'Next tab', 'Add', and 'Update/Display'. A link for 'Custom Properties' is also present.

14. Leave the Peoplesoft console.
15. Start **PeopleTools Application Designer**.
16. Right-click the **LDAPAUTH** field.
17. Select **View PeopleCode**.



18. The **LDAPAUTH** code windows opens.
19. Locate the **OAMSSO_AUTHENTICATION** function.
20. Replace the **&defaultUserId** value with **OAMPSFT**.



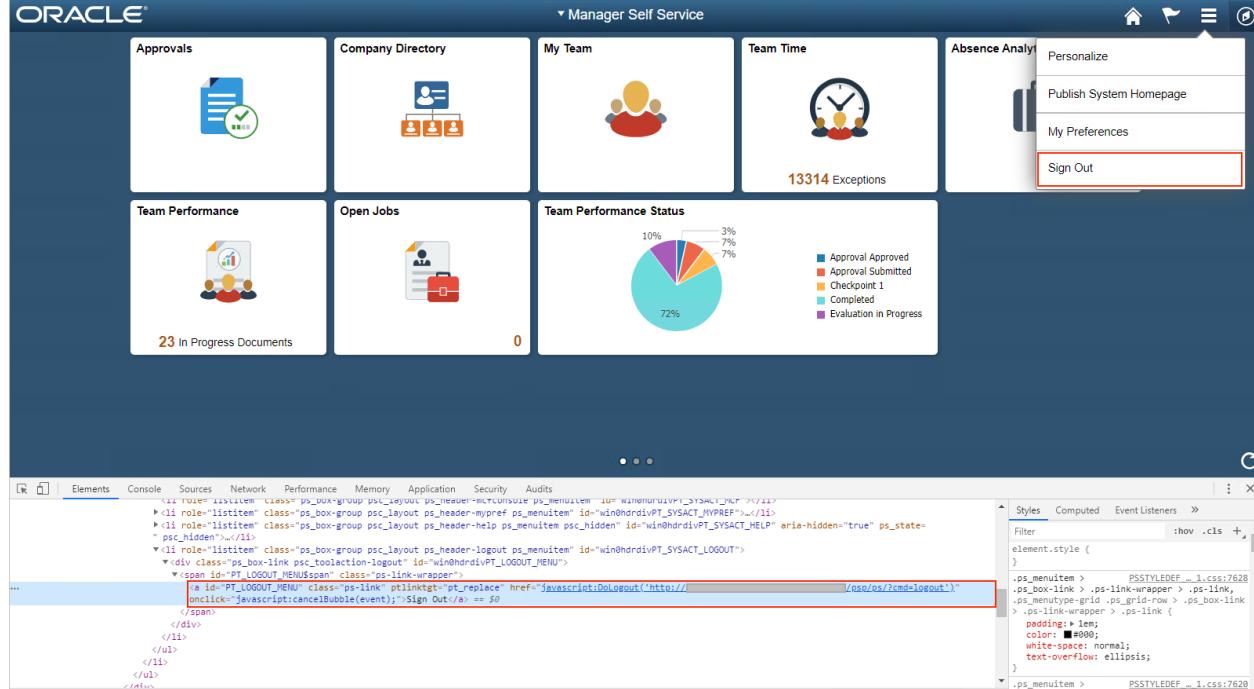
21. Save the record.
22. Navigate to **PeopleTools > Security.
23. Select **Security Objects**.
24. Select **Sign on PeopleCode**.
25. Enable **OAMSSO_AUTHENTICATION**.

PeopleSoft Single Logout

When you sign out of [My Apps](#), PeopleSoft SLO is initiated, which in turn calls the BIG-IP SLO endpoint. BIG-IP needs instructions to perform SLO on behalf of the application. Have the BIG-IP listen for user sign out requests to PeopleSoft, and then trigger SLO.

Add SLO support for PeopleSoft users.

1. Obtain the PeopleSoft portal sign-out URL.
2. Open the portal with a web browser.
3. Enable the debug tools.
4. Locate the element with the PT_LOGOUT_MENU ID.
5. Save the URL path with the query parameters. In this example: /psp/ps/?cmd=logout .



Create a BIG-IP iRule to redirect users to the SAML SP sign out endpoint:

/my.logout.php3 .

1. Navigate to **Local Traffic > iRules List.
2. Select **Create**.
3. Enter a rule **Name**.
4. Enter the following command lines.

```
when HTTP_REQUEST {switch -glob -- [HTTP::uri] { "/psp/ps/?cmd=logout"  
{HTTP::redirect "/my.logout.php3" }}}}
```

5. Select **Finished**.

Assign the iRule to the BIG-IP Virtual Server.

1. Navigate to **Access > Guided Configuration**.
2. Select the PeopleSoft application configuration link.

Configurations

Import	Name	Type	Filter Configurations by Name...
DEPLOYED	Oracle_PeopleSoft	Azure AD Application	

3. From the top navigation bar, select **Virtual Server**.

4. For **Advanced Settings**, select ***On**.

Virtual Server Properties

Advanced Settings On

4. Scroll to the bottom.

5. Under **Common**, add the iRule you created.

Virtual Server Configuration ▾

iRules

Available

Filter
Common
_sys_APM_activesync
_sys_APM_ExchangeSupport_helper
_sys_APM_ExchangeSupport_main

Selected

Common
irule_PeopleSoft

5. Select **Save**.

6. Select **Next**.

7. Continue to configure settings.

To learn more, go to support.f5.com for:

- K42052145: Configuring automatic session termination (logout) based on a URI-referenced file name ↗
- K12056: Overview of the Logout URI Include option ↗

Default to PeopleSoft landing page

Redirect user requests from the root ("/") to the external PeopleSoft portal, usually located in: "/psc/ps/EXTERNAL/HRMS/c/NUI_FRAMEWORK.PT_LANDINGPAGE.GBL"

1. Navigate to **Local Traffic > iRule**.
2. Select **iRule_PeopleSoft**.
3. Add the following command lines.

```
when HTTP_REQUEST {switch -glob -- [HTTP::uri] {"/" {HTTP::redirect  
"/psc/ps/EXTERNAL/HRMS/c/NUI_FRAMEWORK.PT_LANDINGPAGE.GB"/psc/ps/?cmd=logout"  
{HTTP::redirect "/my.logout.php3"} } }
```

4. Assign the iRule to the BIG-IP Virtual Server.

Confirm configuration

1. With a browser, go to the PeopleSoft application external URL, or select the application icon in [My Apps](#).
2. Authenticate to Microsoft Entra ID.
3. You're redirected to the BIG-IP virtual server and signed in with SSO.

 **Note**

You can block direct access to the application, thereby enforcing a path through the BIG-IP.

Advanced deployment

Sometimes, the Guided Configuration templates lack flexibility.

Learn more: [Tutorial: Configure F5 BIG-IP Access Policy Manager for header-based SSO](#)

Alternatively, in BIG-IP disable the Guided Configuration strict management mode. You can manually change configurations, although most configurations are automated with wizard templates.

1. Navigate to **Access > Guided Configuration**.
2. At the end of the row, select the **padlock**.



Changes with the wizard UI aren't possible, however BIG-IP objects associated with the application published instance are unlocked for management.

 **Note**

When you reenable strict mode and deploy a configuration, settings performed outside Guided Configuration are overwritten. We recommend advanced configuration for production services.

Troubleshooting

Use BIG-IP logging to isolate issues with connectivity, SSO, policy violations, or misconfigured variable mappings.

Log verbosity

1. Navigate to **Access Policy > Overview**.
2. Select **Event Logs**.
3. Select **Settings**.
4. Select the row of your published application.
5. Select **Edit**.
6. Select **Access System Logs**
7. From the SSO list, select **Debug**.
8. Select **OK**.
9. Reproduce your issue.
10. Inspect the logs.

When complete, revert this feature because verbose mode generates lots of data.

BIG-IP error message

If a BIG-IP error appears after Microsoft Entra preauthentication, it's possible the issue relates to Microsoft Entra ID to BIG-IP SSO.

1. Navigate to **Access > Overview**.

2. Select **Access reports**.
3. Run the report for the last hour.
4. Review the logs for clues.

Use the session's **View session** link to confirm the APM receives expected Microsoft Entra claims.

No BIG-IP error message

If no BIG-IP error message appears, the issue might be related to the back-end request, or BIG-IP to application SSO.

1. Navigate to **Access Policy > Overview**.
2. Select **Active Sessions**.
3. Select the active session link.

Use the **View Variables** link to determine SSO issues, particularly if BIG-IP APM obtains incorrect attributes from session variables.

Learn more:

- Go to devcentral.f5.com for [APM variable assign examples](#)
- Go to techdocs.f5.com for [Session Variables](#)

Feedback

Was this page helpful?



[Provide product feedback](#) | [Get help at Microsoft Q&A](#)

Tutorial: Configure F5 BIG-IP Easy Button for SSO to SAP ERP

Article • 06/28/2024

In this article, learn to secure SAP Enterprise Resource Planning (ERP) using Microsoft Entra ID, with F5 BIG-IP Easy Button Guided Configuration 16.1. Integrating a BIG-IP with Microsoft Entra ID has many benefits:

- Zero Trust framework to enable remote work [↗](#)
- What is Conditional Access?
- Single sign-on (SSO) between Microsoft Entra ID and BIG-IP published services
- Manage identities and access from the [Microsoft Entra admin center](#) [↗](#)

Learn more:

- [Integrate F5 BIG-IP with Microsoft Entra ID](#)
- [Enable SSO for an enterprise application.](#)

Scenario description

This scenario includes the SAP ERP application using Kerberos authentication to manage access to protected content.

Legacy applications lack modern protocols to support integration with Microsoft Entra ID. Modernization is costly, requires planning, and introduces potential downtime risk. Instead, use an F5 BIG-IP Application Delivery Controller (ADC) to bridge the gap between the legacy application and the modern ID control plane, through protocol transitioning.

A BIG-IP in front of the application enables overlay of the service with Microsoft Entra preauthentication and headers-based SSO. This configuration improves overall application security posture.

Scenario architecture

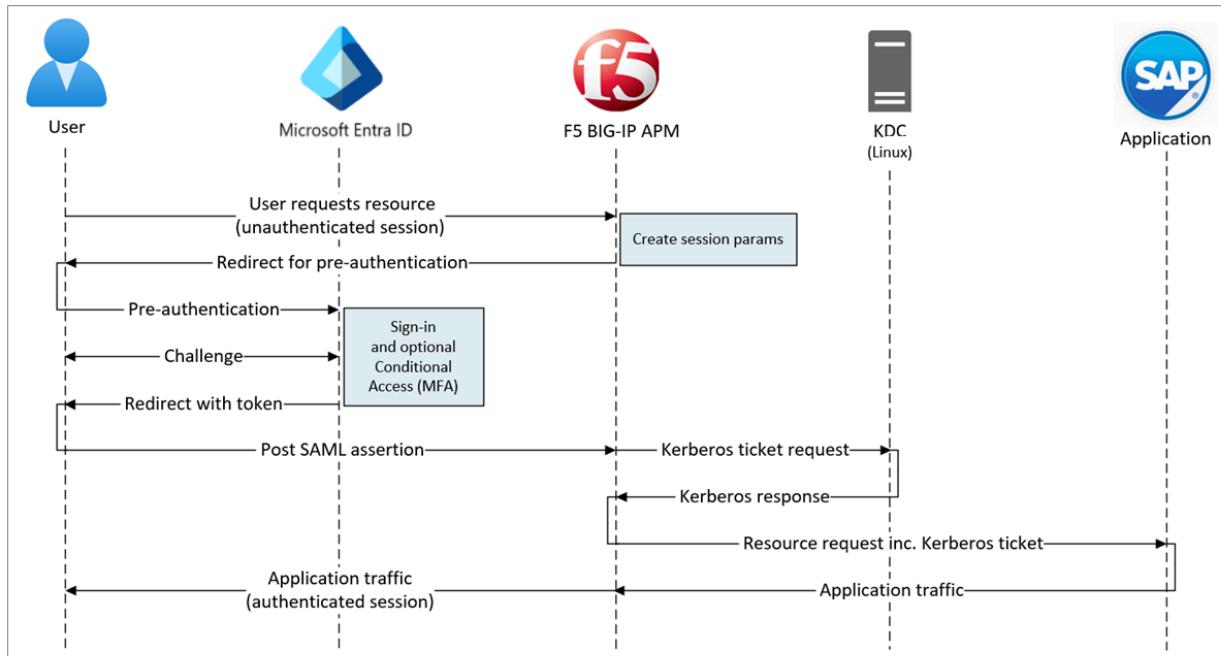
The secure hybrid access (SHA) solution has the following components:

- **SAP ERP application** - a BIG-IP published service protected by Microsoft Entra SHA
- **Microsoft Entra ID** - Security Assertion Markup Language (SAML) identity provider (IdP) that verifies user credentials, Conditional Access, and SAML-based SSO to the

BIG-IP

- BIG-IP - reverse-proxy and SAML service provider (SP) to the application. BIG-IP delegates authentication to the SAML IdP then performs header-based SSO to the SAP service

SHA supports SP and IdP initiated flows. The following image illustrates the SP-initiated flow.



Prerequisites

- A Microsoft Entra ID Free account, or higher
 - If you don't have one, get an [Azure free account](#)
- A BIG-IP or a BIG-IP Virtual Edition (VE) in Azure
 - See, [Deploy F5 BIG-IP Virtual Edition virtual machine \(VM\) in Azure](#)
- Any of the following F5 BIG-IP licenses:
 - F5 BIG-IP® Best bundle
 - F5 BIG-IP APM standalone license

- F5 BIG-IP APM add-on license on an existing BIG-IP F5 BIG-IP® Local Traffic Manager™ (LTM)
- 90-day BIG-IP full feature [trial license](#)
- User identities synchronized from an on-premises directory to Microsoft Entra ID, or created in Microsoft Entra ID and flowed back to the on-premises directory
 - See, [Microsoft Entra Connect Sync: Understand and customize synchronization](#)
- One of the following roles: Cloud Application Administrator, or Application Administrator.
- An SSL Web certificate to publish services over HTTPS, or use default BIG-IP certs for testing
 - See, [Deploy F5 BIG-IP Virtual Edition VM in Azure](#)
- An SAP ERP environment configured for Kerberos authentication

BIG-IP configuration methods

This tutorial uses Guided Configuration 16.1 with an Easy Button template. With the Easy Button, admins don't go between Microsoft Entra ID and a BIG-IP to enable services for SHA. The APM Guided Configuration wizard and Microsoft Graph handle deployment and policy management. This integration ensures applications support identity federation, SSO, and Conditional Access.

ⓘ Note

Replace example strings or values in this guide with those in your environment.

Register Easy Button

💡 Tip

Steps in this article might vary slightly based on the portal you start from.

Before a client or service accesses Microsoft Graph, the Microsoft identity platform must trust it.

See, [Quickstart: Register an application with the Microsoft identity platform](#)

Register the Easy Button client in Microsoft Entra ID, then establishes a trust between SAML SP instances of a BIG-IP published application, and Microsoft Entra ID as the SAML IdP.

1. Sign in to the Microsoft Entra admin center [↗](#) as at least a **Cloud Application Administrator**.
2. Browse to **Identity > Applications > App registrations > New registration**.
3. Enter a **Name** for the new application.
4. In **Accounts in this organizational directory only**, specify who can use the application.
5. Select **Register**.
6. Navigate to **API permissions**.
7. Authorize the following Microsoft Graph Application permissions:
 - Application.Read.All
 - Application.ReadWrite.All
 - Application.ReadWrite.OwnedBy
 - Directory.Read.All
 - Group.Read.All
 - IdentityRiskyUser.Read.All
 - Policy.Read.All
 - Policy.ReadWrite.ApplicationConfiguration
 - Policy.ReadWrite.ConditionalAccess
 - User.Read.All
8. Grant admin consent for your organization.
9. On **Certificates & Secrets**, generate a new **client secret**.
10. Note the secret.
11. From **Overview**, note the **Client ID** and **Tenant ID**.

Configure the Easy Button

1. Initiate the APM Guided Configuration.
2. Launch the Easy Button template.
3. From a browser, sign-in to the F5 BIG-IP management console.
4. Navigate to **Access > Guided Configuration > Microsoft Integration**.
5. Select **Microsoft Entra Application**.
6. Review the configuration list.
7. Select **Next**.

8. Follow the configuration sequence under **Microsoft Entra Application Configuration**.



Configuration Properties

The **Configuration Properties** tab has service account properties and creates a BIG-IP application config and SSO object. The **Azure Service Account Details** section represents the client you registered as an application, in the Microsoft Entra tenant. Use the settings for BIG-IP OAuth client to individually register a SAML SP in the tenant, with the SSO properties. Easy Button does this action for BIG-IP services published and enabled for SHA.

ⓘ Note

Some settings are global and can be re-used to publish more applications.

1. Enter a **Configuration Name**. Unique names differentiate Easy Button configurations.
2. For **Single Sign-On (SSO) & HTTP Headers**, select **On**.
3. For **Tenant ID**, **Client ID**, and **Client Secret**, enter the Tenant ID, Client ID, and Client Secret you noted during tenant registration.
4. Select **Test Connection**. This action confirms the BIG-IP connects to your tenant.
5. Select **Next**.

Configuration Properties

General Properties ▾

Configuration Name

 SAP

Type a name for this guided configuration.

Description ⓘ

 On

Single Sign-On (SSO) & HTTP Headers ⓘ



Endpoint Checks ⓘ



Additional Checks ⓘ

Azure Service Account Details ▾



Copy Account Info from Existing Configuration ⓘ

Tenant ID ⓘ

 4e95c47f-b591-4ced-a6b1-c7ac7b5a9436

Client ID ⓘ

 8b52b0b7-cf0e-4b9f-8f18-0a9wa3100e7e

Client Secret ⓘ

Connection is valid

Application Settings ▾



Use an existing Azure application ⓘ

Service Provider

Use the Service Provider settings to define SAML SP instance properties of the application secured by SHA.

1. For **Host**, enter the public fully qualified domain name (FQDN) of the application being secured.
2. For **Entity ID**, enter the identifier Microsoft Entra ID uses to identify the SAML SP requesting a token.

Service Provider

Advanced Settings

Service Provider Properties ▾

Host ⓘ

eportal.contoso.com

Entity ID ⓘ

https://eportal.contoso.com/

Description ⓘ

(empty)

Relay State ⓘ

(empty)

3. (Optional) Use **Security Settings** to indicate Microsoft Entra ID encrypts issued SAML assertions. Assertions encrypted between Microsoft Entra ID and the BIG-IP APM increase assurance that content tokens aren't intercepted, nor data compromised.

4. From **Assertion Decryption Private Key**, select **Create New**.

Security Settings ▾

Enable Encrypted Assertion ⓘ

Assertion Decryption Private Key ⓘ

Action	Key Details
<input type="button" value="Create New"/>	--Select--
<input type="button" value="Edit"/>	F5Demo Key Type: rsa-private Last Update Time: Thu, May 6, 2021 Security Type: normal
<input type="button" value="Delete"/>	F5DemoCert Key Type: rsa-private Last Update Time: Fri, May 7, 2021 Security Type: normal
<input type="button" value="Import"/>	(empty)
<input type="button" value="Create New"/>	(empty)

5. Select **OK**.

6. The **Import SSL Certificate and Keys** dialog appears in a new tab.

7. To import the certificate and private key, select **PKCS 12 (IIS)**.

8. Close the browser tab to return to the main tab.

SSL Certificate/Key Source

Import Type	PKCS 12 (IIS) <input checked="" type="checkbox"/>
Certificate and Key Name	<input checked="" type="radio"/> New <input type="radio"/> Overwrite Existing Contoso_SAML_Cert
Certificate and Key Source	Choose File <input type="file"/> No file chosen
Password	*****
Key Security	Normal <input type="button" value="▼"/>
Free Space on Disk	8022 MB

9. For **Enable Encrypted Assertion**, check the box.

10. If you enabled encryption, from the **Assertion Decryption Private Key** list, select the private key for the certificate BIG-IP APM uses to decrypt Microsoft Entra assertions.
11. If you enabled encryption, from the **Assertion Decryption Certificate** list, select the certificate BIG-IP uploads to Microsoft Entra ID to encrypt the issued SAML assertions.

Security Settings ▾

Enable Encrypted Assertion i

Assertion Decryption Private Key i
 Contoso_SAML_cert

Assertion Decryption Certificate i
 Contoso_SAML_cert

Microsoft Entra ID

Easy Button has application templates for Oracle PeopleSoft, Oracle E-Business Suite, Oracle JD Edwards, SAP ERP, and a generic SHA template.

1. To start Azure configuration, select **SAP ERP Central Component > Add**.

Azure Configuration ⚠

User Attributes & Claims Conditional Access Policy Additional User Attributes

Configuration Properties ▾

Search

Oracle
PeopleSoft -...

SAP ERP Central
Component...

Oracle E-
Business Suite...

JD Edwards -
Protected by F...

Add

! Note

You can use the information in the following sections when manually configuring a new BIG-IP SAML application in a Microsoft Entra tenant.

Azure Configuration

1. For **Display Name** enter the app BIG-IP creates in the Microsoft Entra tenant. The name appears on the icon in the [My Apps](#) portal.
2. (Optional) leave **Sign On URL (optional)** blank.

Azure Configuration ⚠ **User Attributes & Claims** **Conditional Access Policy**

Advanced Settings

Configuration Properties ▾



SAP ERP Central Component...

Change

Display Name SAP

Sign On URL https://eportal.contoso.com

SAML Signing Certificate ▾

Signing Key Contoso_Wildcard_Cert ↻

Signing Certificate Contoso_Wildcard_Cert ↻

Signing Key Passphrase

Signing Option Sign SAML assertion

Signing Algorithm RSA-SHA256

7. **User and User Groups** are dynamically queried from your Microsoft Entra tenant.
Groups help authorize application access.

8. Add a user or group for testing, otherwise access is denied.

Name	Type	Description
Contoso_Personnel	User Group	Contoso full time employees

User Attributes & Claims

When users authenticate to Microsoft Entra ID, it issues a SAML token with default claims and attributes identifying the user. The **User Attributes & Claims** tab shows the default claims to issue for the new application. Use it to configure more claims.

This tutorial is based on a .com domain suffix used internally and externally. No other attributes are required to achieve a functional Kerberos constrained delegation (KCD) SSO implementation.

Claim Name	Value
Unique User Identifier (Name ID)	user.userprincipalname
Identity	user.onpremisessamaccountname

Claim Name	Value	Add
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	user.mail	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	user.given...	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	user.user...	
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	user.surn...	

You can include more Microsoft Entra attributes. For this tutorial, SAP ERP requires the default attributes.

Learn more: [Tutorial: Configure F5 BIG-IP Access Policy Manager for Kerberos authentication](#). See, instructions on multiple domains or user sign in with alternate suffixes.

Additional User Attributes

The **Additional User Attributes** tab supports distributed systems requiring attributes stored in other directories, for session augmentation. Thus, attributes from a Lightweight Directory Access Protocol (LDAP) source are injected as more SSO headers to control role-based access, Partner IDs, and so on.

 **Note**

This feature has no correlation to Microsoft Entra ID but is another attribute source.

Conditional Access Policy

Conditional Access policies are enforced after Microsoft Entra preauthentication. This action controls access based on device, application, location, and risk signals.

The **Available Policies** view lists Conditional Access policies without user-based actions.

The **Selected Policies** view lists policies targeting cloud apps. You can't deselect policies enforced at the tenant level, nor move them to the Available Policies list.

To select a policy for the application being published:

1. From the **Available Policies** list, select the policy.
2. Select the right arrow.
3. Move the policy to the **Selected Policies** list.

Selected policies have an **Include** or **Exclude** option checked. If both options are checked, the selected policy isn't enforced.

Azure Configuration User Attributes & Claims Additional User Attributes **Conditional Access Policy**

Conditional Access Policy ▾

[View Conditional Access policies in the Azure Portal](#)

Available Policies ⓘ		Selected Policies ⓘ			
Items: 11	Filter by Name...	Name	Include	Exclude	Apps
		F5 Intranet - MCAS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
		Intranet - Custom Control	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
		Intranet - Block downloads	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
		SPO - Managed endpoints only	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
		SharePoint Restricted Content	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
		Breakglass	<input type="checkbox"/>	<input type="checkbox"/>	All
		Managed or Compliant endpoints	<input type="checkbox"/>	<input type="checkbox"/>	All
		Restrict office 365 to Registered users	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
		Block all	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All
		Block Legacy clients (Office, Internet)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	All

ⓘ Note

The policy list appears when you initially select this tab. Use the **refresh** button to query your tenant. Refresh appears when the application is deployed.

Virtual Server Properties

A virtual server is a BIG-IP data plane object represented by a virtual IP address. This server listens for client requests to the application. Received traffic is processed and evaluated against the APM profile associated with the virtual server. Traffic is then directed according to policy.

1. Enter a **Destination Address**. Use the IPv4/IPv6 address BIG-IP uses to receive client traffic. A corresponding record is in the domain name server (DNS), which enables clients to resolve the external URL of BIG-IP published application to this IP. You can use a test computer localhost DNS for testing.
2. For **Service Port**, enter 443.
3. Select **HTTPS**.
4. For **Enable Redirect Port**, check the box.
5. For **Redirect Port**, enter a number and select **HTTP**. This option redirects incoming HTTP client traffic to HTTPS.

6. Select the **Client SSL Profile** you created. Or, leave the default for testing. The Client SSL Profile enables the virtual server for HTTPS, so client connections are encrypted over Transport Layer Security (TLS).

Virtual Server Properties

Advanced Settings

General Properties ▾

Virtual Server
 Create New Use Existing

Destination Address ⓘ

Service Port ⓘ

Enable Redirect Port ⓘ

Redirect Port ⓘ

Client SSL Profile ⓘ
 Create new Use Existing

Available	Selected
<input type="button" value="Filter"/>	Common
Common	<input checked="" type="checkbox" value="Contoso_clientssl"/>
clientssl	
clientssl-insecure-compatible	
<input type="button" value="Create Profile in BIG-IP UI"/>	<input type="button" value=""/>

Pool Properties

The **Application Pool** tab has services behind a BIG-IP, represented as a pool with application servers.

1. For **Select a Pool**, select **Create New**, or select a pool.
2. For **Load Balancing Method**, select **Round Robin**.
3. For **Pool Servers** select a server node, or enter an IP and port for the back-end node hosting the header-based application.

Application Pool ▾

Select a Pool
Create New

Select an existing pool or select Create New.

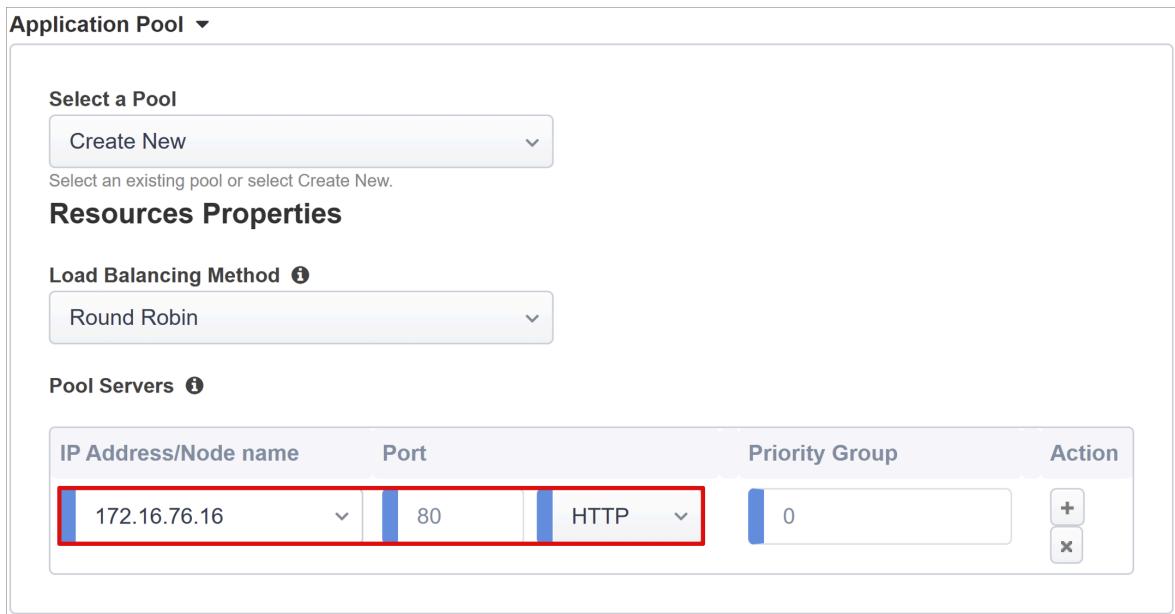
Resources Properties

Load Balancing Method ⓘ
Round Robin

Pool Servers ⓘ

IP Address/Node name	Port	Priority Group	Action
172.16.76.16	80	HTTP	0

+
x



Single Sign-On & HTTP Headers

Use SSO to enable access BIG-IP published services without entering credentials. The Easy Button wizard supports Kerberos, OAuth Bearer, and HTTP authorization headers for SSO. For the following instructions, you need the Kerberos delegation account you created.

1. On **Single Sign-On & HTTP Headers**, for **Advanced Settings**, select **On**.
2. For **Selected Single Sign-On Type**, select **Kerberos**.
3. For **Username Source**, enter a session variable as the user ID source.
`session.saml.last.identity` holds the Microsoft Entra claim with the signed-in user ID.
4. The **User Realm Source** option is required if the user domain differs from the BIG-IP kerberos realm. Thus, the APM session variable contains the signed in user domain. For example, `session.saml.last.attr.name.domain`.

Single Sign-On & HTTP Headers

Advanced Settings On

Single Sign-On ▾

Selected Single Sign-On Type i

Kerberos

SSO Configuration Object i

Create New



Credentials Source ▾

Username Source i

session.saml.last.identity

User Realm Source i

5. For **KDC**, enter a domain controller IP, or FQDN if the DNS is configured.
6. For **UPN Support**, check the box. The APM uses the User Principal Name (UPN) for kerberos ticketing.
7. For **SPN Pattern**, enter **HTTP/%h**. This action informs the APM to use the client-request host header and build the Service Principal Name (SPN) for which it's requesting a kerberos token.
8. For **Send Authorization**, disable the option for applications that negotiate authentication. For example, **Tomcat**.

SSO Method Configuration ▾

Kerberos Realm ⓘ
CONTOSO.COM

Account Name ⓘ
host/f5-big-ip.contoso.com@CONTOSO.COM

Account Password
.....

The password for the delegation account specified in the previous field.

Confirm Account Password
.....

Re-type the password for the delegation account specified in the previous field.

KDC ⓘ
172.16.76.4

UPN Support
Enable this to allow the User Principal Name to be used for SSO.

SPN Pattern ⓘ
HTTP/%h

Ticket Lifetime ⓘ
600

Send Authorization ⓘ
Always

This screenshot shows the configuration page for SSO Method. It includes fields for Kerberos Realm (CONTOSO.COM), Account Name (host/f5-big-ip.contoso.com@CONTOSO.COM), and Account Password (redacted). A note below the password field says 'The password for the delegation account specified in the previous field.' The KDC field is set to 172.16.76.4. The 'UPN Support' checkbox is checked. The SPN Pattern is set to HTTP/%h. The Ticket Lifetime is set to 600. The Send Authorization dropdown is set to 'Always'. The entire 'Kerberos Realm' and 'Account Name' section, as well as the 'KDC' and 'UPN Support' section, are highlighted with a red box.

Session Management

Use BIG-IP session management settings to define conditions when user sessions terminate or continue. Conditions include limits for users and IP addresses, and corresponding user info.

To learn more, go to [my.f5.com for K18390492: Security | BIG-IP APM operations guide ↗](#)

The operations guide doesn't cover Single Log-Out (SLO). This feature ensures sessions between the IdP, the BIG-IP, and the user agent terminate when users sign out. The Easy Button deploys a SAML application to the Microsoft Entra tenant. It populates the Logout URL with the APM SLO endpoint. IdP initiated sign out from the [My Apps ↗](#) portal terminates the BIG-IP and client session.

During deployment, the published-application SAML federation metadata is imported from the tenant. This action provides the APM the SAML sign out endpoint for Microsoft Entra ID and helps SP-initiated sign out terminate the client and Microsoft Entra session.

Deployment

1. Select Deploy.
2. Verify the application is in the tenant **Enterprise applications** list.
3. With a browser, connect to the application external URL or select the application icon in [My Apps](#).
4. Authenticate to Microsoft Entra ID.
5. Redirection takes you to the BIG-IP virtual server and signed in through SSO.

For increased security, you can block direct access to the application and enforce a path through the BIG-IP.

Advanced deployment

The Guided Configuration templates sometimes lack flexibility.

Learn more: [Tutorial: Configure F5 BIG-IP Access Policy Manager for Kerberos authentication.](#)

Disable strict management mode

Alternatively, in BIG-IP you can disable Guided Configuration strict management mode. You can change your configurations manually, although most configurations are automated with wizard templates.

1. Navigate to **Access > Guided Configuration**.
2. At the end of the row for your application configuration, select the **padlock**.
3. BIG-IP objects associated with the published application are unlocked for management. Changes via the wizard UI are no longer possible.



ⓘ Note

To re-enable strict management mode and deploy a configuration that overwrites settings outside the Guided Configuration UI, we recommend the advanced configuration method for production services.

Troubleshooting

If you're unable to access the SHA-secured application, see the following troubleshooting guidance.

- Kerberos is time sensitive. Ensure servers and clients are set to the correct time, and synchronized to a reliable time source.
- Ensure the domain controller and web app hostname resolve in DNS.
- Confirm no duplicate SPNs in the environment.
 - On a domain computer, at the command line, use the query: `setspn -q`
`HTTP/my_target_SPN`

To validate an Internet Information Services (IIS) application KCD configuration, see [Troubleshoot KCD configurations for Application Proxy](#)

Go to [techdocs.f5.com](#) for [Kerberos single sign-on method](#)

Log analysis

Log verbosity

BIG-IP logging isolates issues with connectivity, SSO, policy violations, or misconfigured variable mappings. To start troubleshooting, increase log verbosity.

1. Navigate to **Access Policy > Overview**.
2. Select **Event Logs**.
3. Select **Settings**.
4. Select the row for your published application.
5. Select **Edit**.
6. Select **Access System Logs**
7. From the SSO list, select **Debug**.
8. Select **OK**.
9. Reproduce your issue.
10. Inspect the logs.

When inspection is complete, revert log verbosity because this mode generates excessive data.

BIG-IP error message

If a BIG-IP error message appears after Microsoft Entra preauthentication, the issue might relate to Microsoft Entra ID to BIG-IP SSO.

1. Navigate to **Access > Overview**.
2. Select **Access reports**.
3. Run the report for the last hour.
4. Inspect the logs.

Use the current session's **View session variables** link to see if APM receives expected Microsoft Entra claims.

No BIG-IP error message

If no BIG-IP error message appeared, the issue might be related to the back-end request, or BIG-IP to application SSO.

1. Navigate to **Access Policy > Overview**.
2. Select **Active Sessions**.
3. Select the link for the current session.
4. Use the **View Variables** link to identify KCD issues, particularly if the BIG-IP APM doesn't obtain correct user and domain identifiers from session variables.

Learn more:

- Go to devcentral.f5.com for [APM variable assign examples](#)
- Go to techdocs.f5.com for [Session variables](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Tutorial: Configure F5 BIG-IP Access Policy Manager for header-based single sign-on

Article • 04/18/2024

Learn to implement secure hybrid access (SHA) with single sign-on (SSO) to header-based applications, using F5 BIG-IP advanced configuration. BIG-IP published applications and Microsoft Entra configuration benefits:

- Improved Zero Trust governance through Microsoft Entra preauthentication and Conditional Access
 - See, [What is Conditional Access?](#)
 - See, [Zero Trust security](#)
- Full SSO between Microsoft Entra ID and BIG-IP published services
- Managed identities and access from one control plane
 - See, the [Microsoft Entra admin center](#) ↗

Learn more:

- [Integrate F5 BIG-IP with Microsoft Entra ID](#)
- [Enable SSO for an enterprise application](#)

Scenario description

For this scenario, there's a legacy application using HTTP authorization headers to control access to protected content. Ideally, Microsoft Entra ID manages application access. However, legacy lacks a modern authentication protocol. Modernization takes effort and time, while introducing downtime costs and risks. Instead, deploy a BIG-IP between the public internet and the internal application to gate inbound access to the application.

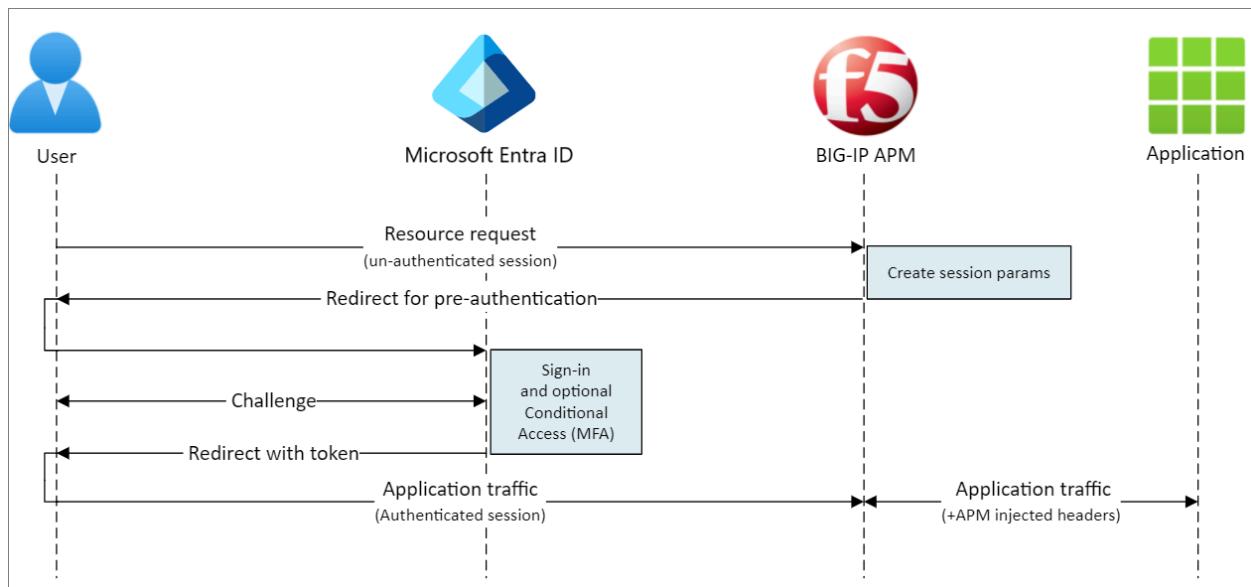
A BIG-IP in front of the application enables overlay of the service with Microsoft Entra preauthentication and header-based SSO. The configuration improves the application security posture.

Scenario architecture

The secure hybrid access solution for this scenario is made up of:

- **Application** - BIG-IP published service to be protected by Microsoft Entra SHA
- **Microsoft Entra ID** - Security Assertion Markup Language (SAML) identity provider (IdP) that verifies user credentials, Conditional Access, and SSO to the BIG-IP
 - With SSO, Microsoft Entra ID provides the BIG-IP required session attributes, including user identifiers
- **BIG-IP** - reverse-proxy and SAML service provider (SP) to the application, delegating authentication to the SAML IdP, before header-based SSO to the back-end application

The following diagram illustrates the user flow with Microsoft Entra ID, BIG-IP, APM, and an application.



1. User connects to application SAML SP endpoint (BIG-IP).
2. BIG-IP APM access policy redirects user to Microsoft Entra ID (SAML IdP).
3. Microsoft Entra preauthenticates user and applies ConditionalAccess policies.
4. User is redirected to BIG-IP (SAML SP) and SSO occurs using issued SAML token.
5. BIG-IP injects Microsoft Entra attributes as headers in request to the application.
6. Application authorizes request and returns payload.

Prerequisites

For the scenario you need:

- An Azure subscription
 - If you don't have one, get an [Azure free account](#)
- One of the following roles: Global Administrator, Cloud Application Administrator, or Application Administrator
- A BIG-IP or deploy a BIG-IP Virtual Edition (VE) in Azure
 - See, [Deploy F5 BIG-IP Virtual Edition Virtual Machine in Azure](#)

- Any of the following F5 BIG-IP licenses:
 - F5 BIG-IP® Best bundle
 - F5 BIG-IP Access Policy Manager™ (APM) standalone license
 - F5 BIG-IP Access Policy Manager™ (APM) add-on license on a BIG-IP F5 BIG-IP® Local Traffic Manager™ (LTM)
 - 90-day BIG-IP full feature trial. See, [Free Trials](#).
- User identities synchronized from an on-premises directory to Microsoft Entra ID
 - [Microsoft Entra Connect Sync: Understand and customize synchronization](#)
- An SSL certificate to publish services over HTTPS, or use default certificates while testing
 - See, [SSL profile](#)
- A header-based application or an IIS header app for testing

BIG-IP configuration method

The following instructions are an advanced configuration method, a flexible way to implement SHA. Manually create BIG-IP configuration objects. Use this method for scenarios not included in the Guided Configuration templates.

 **Note**

Replace example strings or values with those from your environment.

Add F5 BIG-IP from the Microsoft Entra gallery

 **Tip**

Steps in this article might vary slightly based on the portal you start from.

To implement SHA, the first step is to set up a SAML federation trust between BIG-IP APM and Microsoft Entra ID. The trust establishes the integration for BIG-IP to hand off preauthentication and Conditional Access to Microsoft Entra ID, before granting access to the published service.

Learn more: [What is Conditional Access?](#)

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.

3. On the top ribbon, select + New application.
4. In the gallery, search for F5.
5. Select **F5 BIG-IP APM Microsoft Entra ID integration**.
6. Enter an application Name.
7. Select **Add/Create**.
8. The name reflects the service.

Configure Microsoft Entra SSO

1. The new **F5** application properties appear
2. Select **Manage > Single sign-on**
3. On the **Select a single sign-on method** page, select **SAML**.
4. Skip the prompt to save the single sign-on settings.
5. Select **No, I'll save later**.
6. On **Set up single sign-on with SAML, for Basic SAML Configuration**, select the **pen** icon.
7. Replace the **Identifier URL** with the BIG-IP published service URL. For example,
`https://mytravel.contoso.com`
8. Repeat for **Reply URL** and include the APM SAML endpoint path. For example,
`https://mytravel.contoso.com/saml/sp/profile/post/acs`

 **Note**

In this configuration, the SAML flow operates in IdP mode: Microsoft Entra ID issues the user a SAML assertion before being redirected to the BIG-IP service endpoint for the application. The BIG-IP APM supports IdP and SP modes.

9. For **Logout URI** enter the BIG-IP APM Single Logout (SLO) endpoint, prepended by the service host header. The SLO URI ensures user BIG-IP APM sessions end after Microsoft Entra sign-out. For example,
`https://mytravel.contoso.com/saml/sp/profile/redirect/slur`

1 Basic SAML Configuration

Identifier (Entity ID)	https://mytravel.contoso.com
Reply URL (Assertion Consumer Service URL)	https://mytravel.contoso.com/saml/sp/profile/post/acs
Sign on URL	Optional
Relay State	Optional
Logout Url	https://mytravel.contoso.com/saml/sp/profile/redirect/slr

① Note

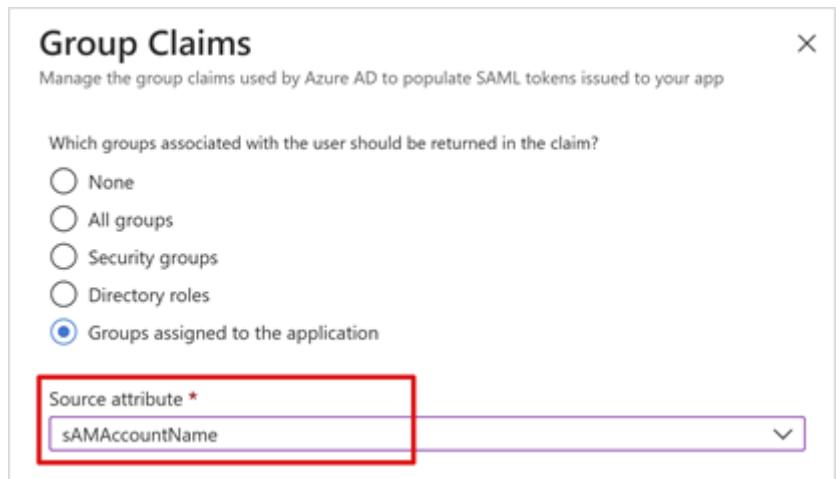
From Traffic Management operating system (TMOS) v16 onward, the SAML SLO endpoint changed to `/saml/sp/profile/redirect/slo`.

10. Select Save.
11. Exit SAML configuration.
12. Skip the SSO test prompt.
13. To edit the **User Attributes & Claims > + Add new claim**, select the **pen icon**.
14. For **Name** select **Employeeid**.
15. For **Source attribute** select **user.employeeid**.
16. Select **Save**

Manage claim

Name *	employeeid
Namespace	
Source *	<input checked="" type="radio"/> Attribute <input type="radio"/> Transformation
Source attribute *	user.employeeid

17. Select **+ Add a group claim**
18. Select **Groups assigned to the application > Source Attribute > sAMAccountName**.



19. Select **Save** the configuration.
20. Close the view.
21. Observe the **User Attributes & Claims** section properties. Microsoft Entra ID issues users properties for BIG-IP APM authentication and SSO to the back-end application.

The screenshot shows the 'User Attributes & Claims' configuration page. On the left, under 'User Attributes', there is a list of attributes: surname, emailaddress, identity, givenname, employeeid, Unique User Identifier, and Group. On the right, under 'Issued Claims', there is a list of claims: user.surname, user.mail, user.onpremisesamaccountname, user.givenname, user.employeeid, user.userprincipalname, and user.groups. An arrow points from the 'Group' attribute in the 'User Attributes' column to the 'user.groups' claim in the 'Issued Claims' column. A blue circle with the number '2' is in the top-left corner of the screenshot area.

ⓘ Note

Add other claims the BIG-IP published application expects as headers. More defined claims are issued if they're in Microsoft Entra ID. Define directory memberships and user objects in Microsoft Entra ID before claims can be issued. See, [Configure group claims for applications by using Microsoft Entra ID](#).

22. In the **SAML Signing Certificate** section, select **Download**.
23. The **Federation Metadata XML** file is saved on your computer.

3 SAML Signing Certificate

Status	Active
Thumbprint	9285F77DC7A9582B1C4FFC690EA8660844E02A56
Expiration	19/03/2023, 2:12:02 PM
Notification Email	rodney@contoso.com
App Federation Metadata Url	https://login.microsoftonline.com/536279f6-15cc-4...
Certificate (Base64)	Download
Certificate (Raw)	Download
Federation Metadata XML	Download

SAML signing certificates created by Microsoft Entra ID have a lifespan of three years.

Microsoft Entra authorization

By default, Microsoft Entra ID issues tokens to users granted access to an application.

1. In the application's configuration view, select **Users and groups**.
2. Select **+ Add user** and in **Add Assignment**, select **Users and groups**.
3. In the **Users and groups** dialog, add the user groups authorized to access the header-based application.
4. Select **Select**.
5. Select **Assign**.

Microsoft Entra SAML federation trust is complete. Next, set up BIG-IP APM to publish the web application, configured with properties to complete SAML preauthentication trust.

Advanced configuration

Use the following sections to configure SAML, header SSO, access profile, and more.

SAML configuration

To federate the published application with Microsoft Entra ID, create the BIG-IP SAML service provider and corresponding SAML IdP objects.

1. Select **Access > Federation > SAML Service Provider > Local SP Services > Create**.

Access » Federation : SAML Service Provider : Local SP Services

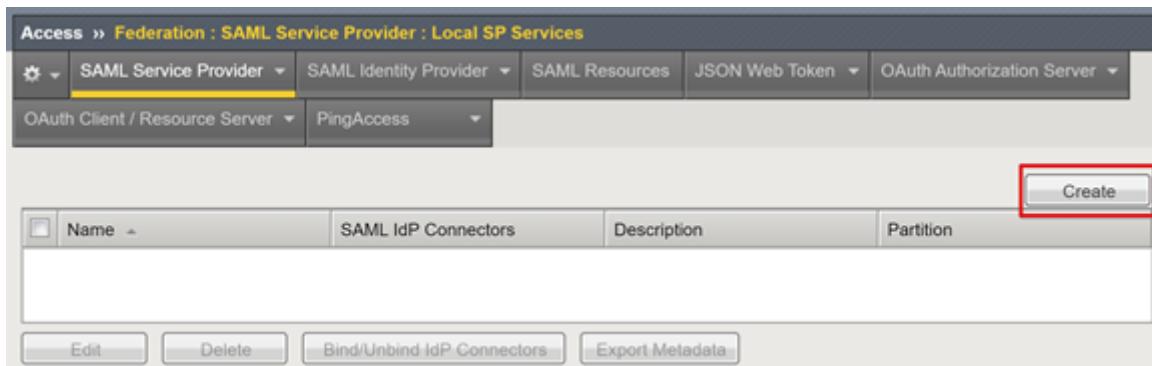
SAML Service Provider SAML Identity Provider SAML Resources JSON Web Token OAuth Authorization Server

OAuth Client / Resource Server PingAccess

Create

Name	SAML IdP Connectors	Description	Partition

Edit Delete Bind/Unbind IdP Connectors Export Metadata



2. Enter a Name.
3. Enter the Entity ID defined in Microsoft Entra ID.

Create New SAML SP Service

General Settings
Endpoint Settings
Security Settings
Authentication Cont...
Requested Attributes
Advanced Settings

Name*: MyTravel

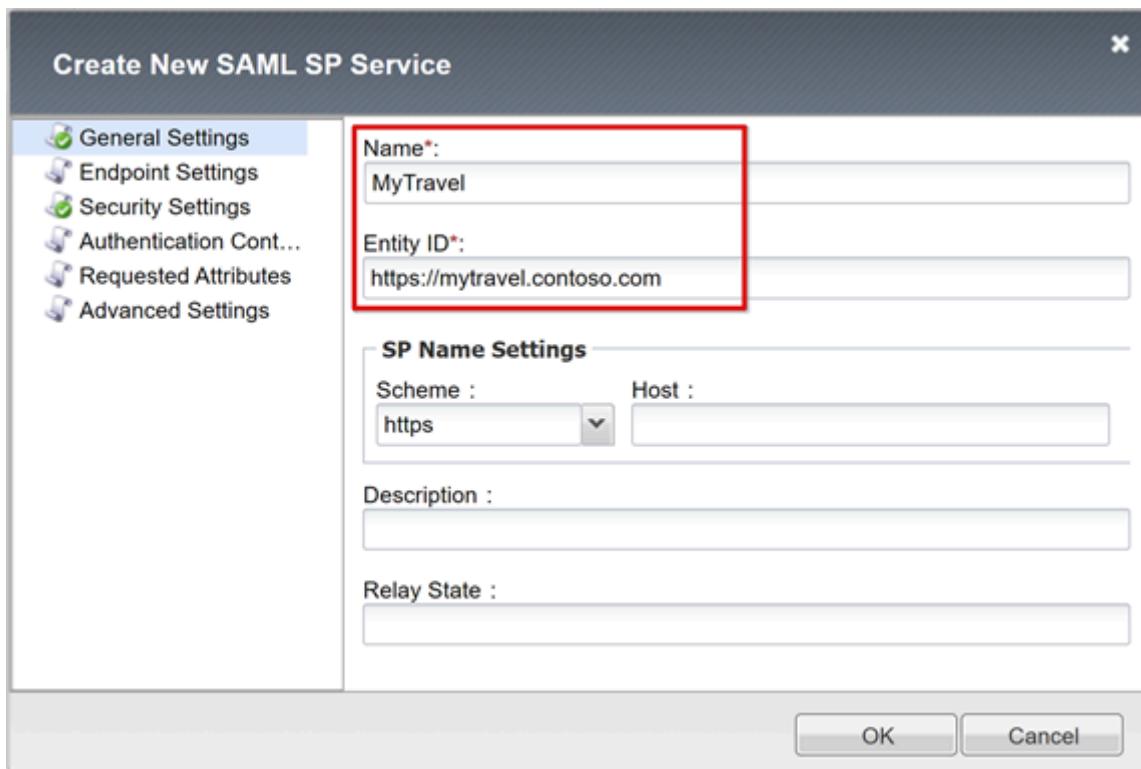
Entity ID*: https://mytravel.contoso.com

SP Name Settings
Scheme : https Host :

Description :

Relay State :

OK Cancel



4. For SP Name Settings, make selections if the Entity ID doesn't match the hostname of the published URL, or make selections if it isn't in regular hostname-based URL format. Provide the external scheme and application hostname if entity ID is urn:mytravel:contosoonline.
5. Scroll down to select the new SAML SP object.
6. Select Bind/UnBind IdP Connectors.

7. Select **Create New IdP Connector**.

8. From the drop-down, select **From Metadata**.

9. Browse to the federation metadata XML file you downloaded.

10. Enter an **Identity Provider Name** for the APM object for the external SAML IdP. For example, `MyTravel_AzureAD`

11. Select **Add New Row**.

12. Select the new SAML IdP Connector.

13. Select **Update**.

Edit SAML IdPs that use this SP

IdP Connectors associated with this SP Service		
	Add New Row	Create New IdP Connector
<input checked="" type="checkbox"/> SAML IdP Connectors	Matching Source	Matching Value
/Common/MyTravel_Az		
	Update	Cancel
Edit	Delete	
OK Cancel		

14. Select OK.

Edit SAML IdPs that use this SP

IdP Connectors associated with this SP Service		
	Add New Row	Create New IdP Connector
<input checked="" type="checkbox"/> SAML IdP Connectors	Matching Source	Matching Value
<input checked="" type="checkbox"/> /Common/MyTravel_A...		
Edit	Delete	
OK Cancel		

Header SSO configuration

Create an APM SSO object.

1. Select Access > Profiles/Policies > Per-Request Policies > Create.
2. Enter a Name.
3. Add at least one Accepted Language.
4. Select Finished.

Access > Profiles / Policies : Per-Request Policies

General Properties

Name	SSO_Headers
Policy Type	All
Incomplete Action	Deny
Customization Type	Modern

Language Settings

Additional Languages	Afar (aa) <input type="button" value="Add"/>
Accepted Languages	
English (en)	
Languages	<input type="button" value="<<"/> <input type="button" value=">>"/>
Default Language	English (en) <input type="button" value=""/>
Factory BuiltIn Languages	
Japanese (ja) Chinese (Simplified) (zh-cn) Chinese (Traditional) (zh-tw) Korean (ko) Spanish (es) French (fr) German (de)	

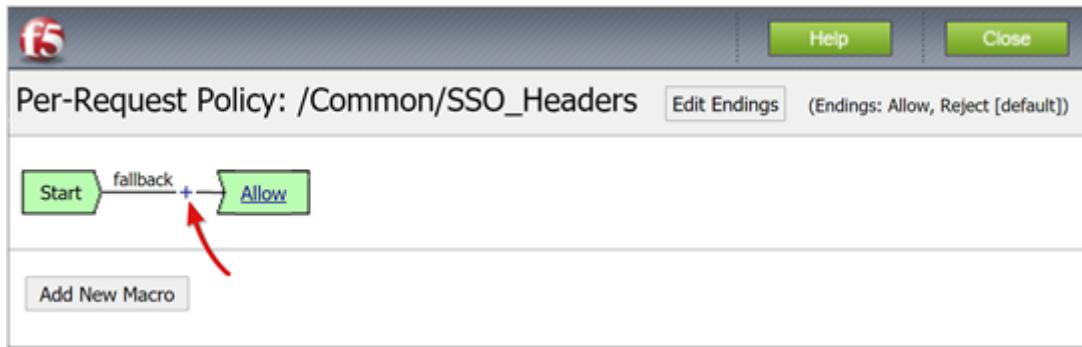
5. For the new per-request policy, select **Edit**.

Access > Profiles / Policies : Per-Request Policies

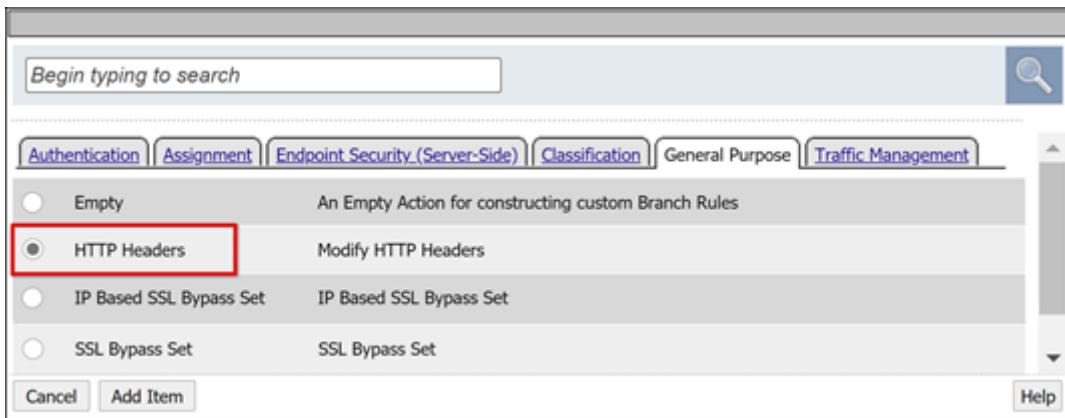
Per-Request Policy Name	Policy Type	Per-Request Policy	Export	Copy	Customization	Virtual Servers	Partition / Path
SSO_Headers	All	<input checked="" type="button" value="Edit..."/> <input type="button" value="Export..."/> <input type="button" value="Copy..."/>	Modern	Common			

6. The visual policy editor starts.

7. Under **fallback**, select the + symbol.



8. On the **General Purpose** tab, select **HTTP Headers > Add Item**.



9. Select **Add new entry**.

10. Create three HTTP and Header modify entries.

11. For **Header Name**, enter **upn**.

12. For **Header Value**, enter `%{session.saml.last.identity}`.

13. For **Header Name**, enter **employeeid**.

14. For **Header Value**, enter `%{session.saml.last.attr.name.employeeid}`.

15. For **Header Name**, enter **group_authz**.

16. For **Header Value**, enter %

```
{session.saml.last.attr.name. http://schemas.microsoft.com/ws/2008/06/identity/cl  
aims/groups }.
```

Note

APM session variables in curly brackets are case sensitive. We recommend you define attributes in lowercase.

Properties* Branch Rules

Name: HTTP Headers

HTTP Header Modify

Add new entry Insert Before: 1 ▾

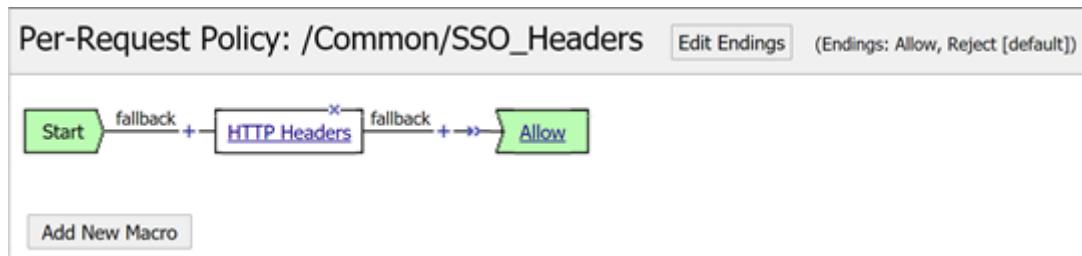
Header Operation	Header Name	Header Value	Header Delimiter
1 insert	upn	%{session.saml.last.identity}	
2 insert	employeeid	.last.id_token.extension_employeeid	
3 insert	group_authz	/ws/2008/06/identity/claims/groups}	

HTTP Cookie Modify

Add new entry Insert Before:

17. Select Save.

18. Close the visual policy editor.



Access profile configuration

An access profile binds many APM elements managing access to BIG-IP virtual servers, including access policies, SSO configuration, and UI settings.

1. Select Access > Profiles / Policies > Access Profiles (Per-Session Policies) > Create.
2. For Name, enter MyTravel.
3. For Profile Type, select All.
4. For Accepted Language, select at least one language.
5. select Finished.

Access » Profiles / Policies : Access Profiles (Per-Session Policies) » New Profile...

General Properties

Name	MyTravel
Parent Profile	access
Profile Type	All
Profile Scope	Profile
Customization Type	Modern

Settings

Custom

Configurations

SSO Across Authentication Domains (Single Domain mode)

Domain Cookie	
Cookie Options	<input checked="" type="checkbox"/> Secure <input type="checkbox"/> Persistent <input type="checkbox"/> HTTP Only <input type="checkbox"/> Samesite
SSO Configuration	None

Language Settings

Additional Languages	Afar (aa) <input type="button" value="Add"/>
Languages	<input type="button" value="Accepted Languages"/> <div style="border: 1px solid #ccc; padding: 5px; width: 200px;"> <input type="checkbox"/> English (en) </div>
Default Language	English (en) <input type="button" value=""/>
<div style="float: right; margin-right: 20px;"> Factory BuiltIn Languages Japanese (ja) Chinese (Simplified) (zh-cn) Chinese (Traditional) (zh-tw) Korean (ko) Spanish (es) French (fr) German (de) </div>	

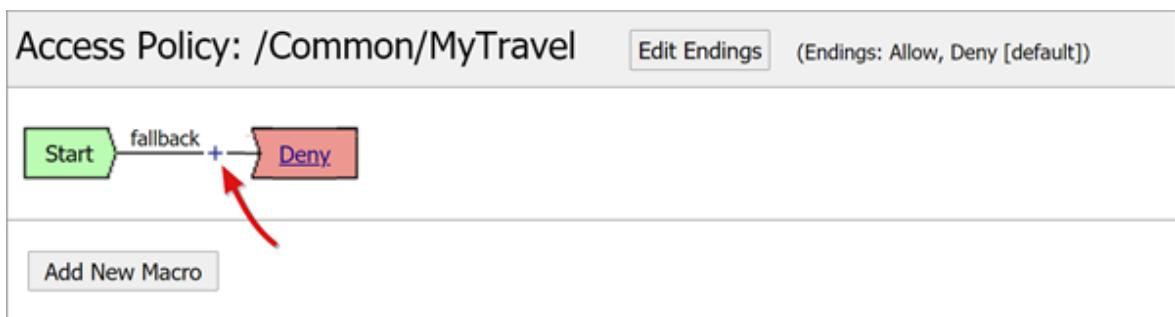
6. For the per-session profile you created, select Edit.

Access » Profiles / Policies : Access Profiles (Per-Session Policies)

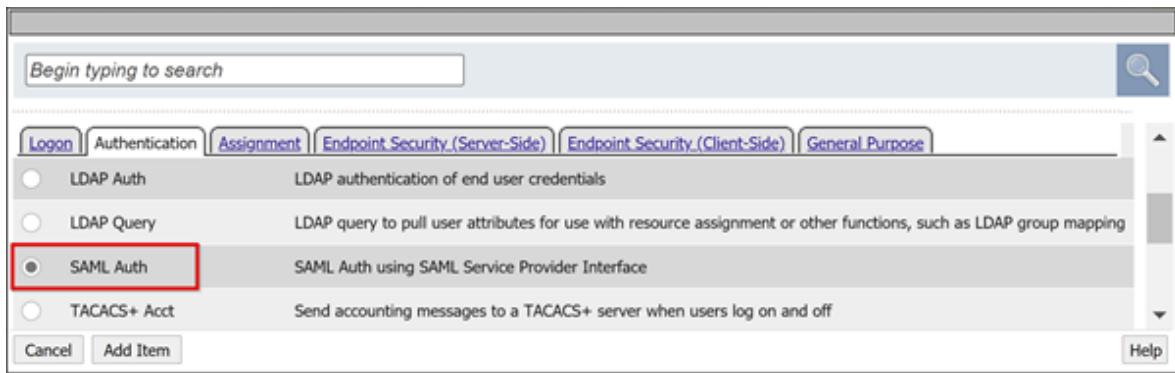
		Status	Access Profile Name	Application	Profile Type	Per-Session Policy	Export	Copy	Customization	Logs	Virtual Servers	Partition / Path
<input checked="" type="checkbox"/>	<input type="checkbox"/>	MyTravel	All	<input type="button" value="Edit..."/>		<input type="button" value="Export..."/>	<input type="button" value="Copy..."/>	Modern	<input type="button" value="default-log-setting"/>		<input type="button" value="Common"/>	
<input type="checkbox"/>	<input type="checkbox"/>	access	All	(none)	(none)	(none)						Common

7. The visual policy editor starts.

8. Under fallback, select the + symbol.



9. Select Authentication > SAML Auth > Add Item.



10. For the **SAML authentication SP** configuration, from the **AAA Server** dropdown, select the SAML SP object you created.

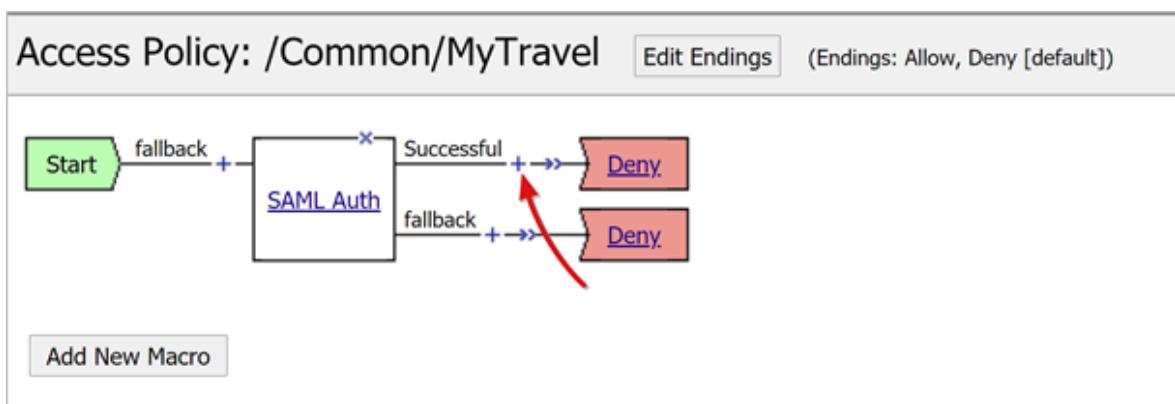
11. Select **Save**.

The screenshot shows the 'SAML Authentication SP' configuration dialog. It has two tabs: 'Properties*' and 'Branch Rules'. The 'Properties*' tab is selected. The 'Name' field contains 'SAML Auth'. Under the 'SAML Authentication SP' section, the 'AAA Server' dropdown is set to '/Common/MyTravel' and is highlighted with a red box. The 'Attribute Consuming Service' dropdown is set to 'None'. The 'Force Authentication' dropdown is set to 'Use AAA server setting'. At the bottom, there are 'Cancel' and 'Save' buttons, with a note '(*Data in tab has been changed, please don't forget to save)' between them.

Attribute mapping

The following instructions are optional. With a LogonID_Mapping configuration, the BIG-IP active sessions list has the signed-in user principal name (UPN), not a session number. Use this data when analyzing logs or troubleshooting.

1. For the SAML Auth Successful branch, select the + symbol.



2. In the pop-up, select **Assignment > Variable Assign > Add Item**.

The screenshot shows a search bar at the top with the placeholder "Begin typing to search". Below it is a navigation bar with tabs: Logon, Authentication, Assignment, Endpoint Security.(Server-Side), Endpoint Security.(Client-Side), and General Purpose. The "Assignment" tab is selected. A list of policy items follows:

- LDAP Group Resource Assign: Map ACLs and resources based on user LDAP group membership
- Links Sections and Webtop Assign: Assign a Webtop, Webtop Links and Webtop Sections
- SSO Credential Mapping: Enables Single Sign-On (SSO) credentials caching and assigns SSO variables
- Variable Assign**: Assign custom variables, configuration variables, or predefined session variables (this item is highlighted with a red border)
- VMware View Policy: Specify a policy that will apply to VMware View connections

At the bottom are "Cancel" and "Add Item" buttons, and a "Help" link.

3. Enter a Name

4. In the Variable Assign section, select Add new entry > change. For example, LogonID_Mapping.

The screenshot shows the "Variable Assign" dialog. It has tabs for "Properties*" and "Branch Rules", with "Properties*" selected. The "Name:" field contains "LogonID_Mapping". Below it is a table titled "Assignment" with one row. The first column of the row contains the number "1", the second column contains the word "empty", and the third column contains a blue link labeled "change". A red arrow points from the text "click here" in the previous step to this "change" link.

5. For Custom Variable, set session.saml.last.identity.

6. For Session Variable, set session.logon.last.username.

7. Select Finished.

8. Select Save.

9. On the Access Policy Successful branch, select the Deny terminal.

10. Select Allow.

11. Select Save.

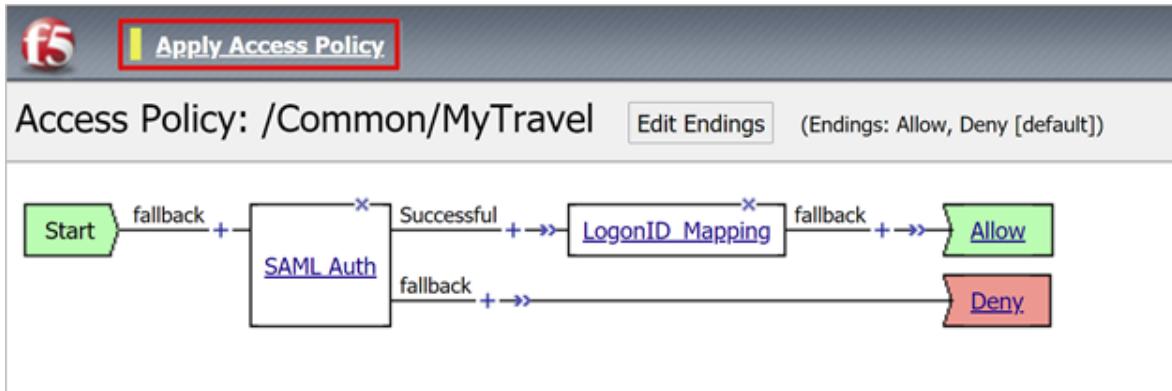
12. Select Apply Access Policy.

13. Close the visual policy editor.

Back-end pool configuration

To enable BIG-IP to forward client traffic correctly, create an APM node object representing the back-end server hosting your application. Place the node in an APM pool.

1. Select Local Traffic > Pools > Pool List > Create.
2. For a server pool object, enter a **Name**. For example, **MyApps_VMs**.



3. Add a pool member object.
4. For **Node Name**, enter a name for the server hosting the back-end web application.
5. For **Address**, enter the IP address of the server hosting the application.
6. For **Service Port** enter the HTTP/S port the application is listening on.

7. Select **Add**.

The screenshot shows the F5 Resources screen under the "New Members" tab. A red box highlights the "Add" button, which has a red arrow pointing to it from the previous step's description. The "Add" button is located next to input fields for "Node Name" (MyTravel_VM1), "Address" (172.16.76.16), and "Service Port" (80).
Below the "Add" button is a table showing the new member details:

Node Name	Address/FQDN	Service Port	Auto Populate	Priority
MyTravel_VM1	172.16.76.16	80		0

At the bottom of the screen are buttons for "Cancel", "Repeat", and "Finished".

① Note

To learn more go to my.f5.com for [K13397: Overview of HTTP health monitor request formatting for the BIG-IP DNS system](#).

Virtual server configuration

A virtual server is a BIG-IP data plane object represented by a virtual IP address listening for clients requests to the application. Received traffic is processed and evaluated with the APM access profile associated with the virtual server. Traffic is directed according to policy.

1. Select Local Traffic > Virtual Servers > Virtual Server List > Create.
2. Enter a virtual server Name.
3. For Destination Address/Mask, select Host
4. Enter an unused IP IPv4 or IPv6 to be assigned to the BIG-IP to receive client traffic.
5. For Service Port, select Port, 443, and HTTPS.

The screenshot shows the 'General Properties' section of the 'New Virtual Server...' dialog. The 'Name' field is set to 'MyTravel'. Under 'Destination Address/Mask', the 'Host' radio button is selected, and the IP address '172.16.76.26' is entered. For 'Service Port', the 'Port' radio button is selected, and the port number '443' is entered, with 'HTTPS' checked. Other fields like 'Description', 'Type', 'Source Address', 'Notify Status to Virtual Address', and 'State' are also visible.

6. For HTTP Profile (Client), select http.
7. For SSL Profile (Client), select the client SSL profile you created, or leave the default for testing.

Configuration: Basic

Protocol	TCP
Protocol Profile (Client)	tcp
Protocol Profile (Server)	(Use Client Profile)
HTTP Profile (Client)	http
HTTP Profile (Server)	(Use Client Profile)
HTTP Proxy Connect Profile	None
FTP Profile	None
RTSP Profile	None
SSL Profile (Client)	<div style="display: flex; align-items: center;"> <div style="flex: 1;"> <p>Selected</p> <ul style="list-style-type: none"> /Common Contoso_clientssl </div> <div style="margin: 0 10px;"> << >> </div> <div style="flex: 1;"> <p>Available</p> <ul style="list-style-type: none"> /Common clientssl clientssl-insecure-compatible clientssl-quic clientssl-secure crypto-server-default-clientssl splitsession-default-clientssl </div> </div>
SSL Profile (Server)	<div style="display: flex; align-items: center;"> <div style="flex: 1;"> <p>Selected</p> <ul style="list-style-type: none"> /Common </div> <div style="margin: 0 10px;"> << >> </div> <div style="flex: 1;"> <p>Available</p> <ul style="list-style-type: none"> /Common apm-default-serverssl cloud-service-default-ssl crypto-client-default-serverssl f5aas-default-ssl pcoip-default-serverssl serverssl-insecure-compatible </div> </div>

8. For **Source Address Translation**, select **Auto Map**.



9. For **Access Policy**, select the **Access Profile** created earlier. This action binds the Microsoft Entra SAML preauthentication profile and headers SSO policy to the virtual server.

10. For **Per-Request Policy**, select **SSO_Headers**.

Access Policy	
Access Profile	MyTravel <input type="button" value="▼"/>
Connectivity Profile	+ None <input type="button" value="▼"/>
Per-Request Policy	SSO_Headers <input type="button" value="▼"/>
VDI Profile	None <input type="button" value="▼"/>
Application Tunnels (Java & Per-App VPN)	<input type="checkbox"/> Enabled
OAM Support	<input type="checkbox"/> Enabled
ADFS Proxy	<input type="checkbox"/> Enabled
PingAccess Profile	None <input type="button" value="▼"/>

11. For **Default Pool**, select the back-end pool objects you created.

12. Select **Finished**.

Resources											
iRules	<table border="1"> <thead> <tr> <th>Enabled</th> <th>Available</th> </tr> </thead> <tbody> <tr> <td><input type="button" value="<<"/></td><td>/Common _sys_APM_ExchangeSupport_OA _sys_APM_ExchangeSupport_OA _sys_APM_ExchangeSupport_he _sys_APM_ExchangeSupport_ma</td></tr> <tr> <td><input type="button" value=">>"/></td><td></td></tr> <tr> <td><input type="button" value="Up"/></td><td></td></tr> <tr> <td><input type="button" value="Down"/></td><td></td></tr> </tbody> </table>	Enabled	Available	<input type="button" value="<<"/>	/Common _sys_APM_ExchangeSupport_OA _sys_APM_ExchangeSupport_OA _sys_APM_ExchangeSupport_he _sys_APM_ExchangeSupport_ma	<input type="button" value=">>"/>		<input type="button" value="Up"/>		<input type="button" value="Down"/>	
Enabled	Available										
<input type="button" value="<<"/>	/Common _sys_APM_ExchangeSupport_OA _sys_APM_ExchangeSupport_OA _sys_APM_ExchangeSupport_he _sys_APM_ExchangeSupport_ma										
<input type="button" value=">>"/>											
<input type="button" value="Up"/>											
<input type="button" value="Down"/>											
Policies	<table border="1"> <thead> <tr> <th>Enabled</th> <th>Available</th> </tr> </thead> <tbody> <tr> <td><input type="button" value="<<"/></td><td></td></tr> <tr> <td><input type="button" value=">>"/></td><td></td></tr> </tbody> </table>	Enabled	Available	<input type="button" value="<<"/>		<input type="button" value=">>"/>					
Enabled	Available										
<input type="button" value="<<"/>											
<input type="button" value=">>"/>											
Default Pool	<input type="button" value="+"/> <input type="button" value="MyApps_VMs"/> <input type="button" value="▼"/>										
Default Persistence Profile	None <input type="button" value="▼"/>										
Fallback Persistence Profile	None <input type="button" value="▼"/>										

Session management

Use the BIG-IPs session management setting to define the conditions for user session termination or continuation. Create policy with **Access Policy > Access Profiles**. Select an application from the list.

Regarding SLO functionality, a SLO URI in Microsoft Entra ID ensures an IdP-initiated sign-out from the MyApps portal terminates the session between the client and the BIG-IP APM. The imported application federation metadata.xml provides the APM with the Microsoft Entra SAML sign-out endpoint, for SP initiated sign-out. Therefore, enable the APM to know when a user signs out.

If there's no BIG-IP web portal, the user can't instruct the APM to sign out. If the user signs out of the application, the BIG-IP is oblivious to the action. The application session can be reinstated through SSO. Therefore, SP-initiated sign out needs careful consideration.

To ensure sessions terminate securely, add an SLO function to your application **Sign out** button. Enable it to redirect the client to the Microsoft Entra SAML sign-out endpoint. For the SAML sign out endpoint for your tenant, go to **App Registrations > Endpoints**.

If you can't change the app, enable the BIG-IP to listen for the app sign-out call and trigger SLO. To learn more:

- Go to support.f5.com for [K42052145: Configuring automatic session termination \(logout\) based on a URI-referenced file name ↗](#)
- Go to my.f5.com for [K12056: Overview of the Logout URI Include option ↗](#)

Deploy

1. Select **Deploy** to commit settings.
2. Verify the application appears in your tenant.
3. The application is published and accessible via SHA, with its URL or Microsoft portals.

Test

Perform the following test as a user.

1. Select the application external URL, or in the MyApps portal select the application icon.
2. Authenticate to Microsoft Entra ID.
3. A redirection occurs to the BIG-IP virtual server for the app and signed in with SSO.
4. The injected header output appears by the header-based application.

The screenshot shows a browser window titled "My Travel" with the URL "https://mytravel.contoso.com". The page displays "Request Details" and "Server Variables".

Request Details

Session Id:	rpxszrbzwvam3mthtbodrgii	Request Type:	GET
Time of Request:	8/15/2021 8:45:17 PM	Status Code:	200
Request Encoding:	Unicode (UTF-8)	Response Encoding:	Unicode (UTF-8)

Server Variables

Name	Value
REMOTE_ADDR	172.16.76.6
REMOTE_PORT	37206
REQUEST_METHOD	GET
SCRIPT_NAME	/default.aspx
SERVER_NAME	mytravel.contoso.com
SERVER_PORT	80
SERVER_PROTOCOL	HTTP/1.1
SERVER_SOFTWARE	Microsoft-IIS/8.5
URL	/default.aspx
HTTP_CACHE_CONTROL	max-age=0
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,app
HTTP_ACCEPT_ENCODING	gzip, deflate, br
HTTP_ACCEPT_LANGUAGE	en-GB,en;q=0.9
HTTP_COOKIE	TIN=20000; LastMRH_Session=120b554e; F5_ST=1z1z1z1629056717z604800
HTTP_HOST	mytravel.contoso.com
HTTP_REFERER	https://login.microsoftonline.com/
HTTP_USER_AGENT	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/9
HTTP_SEC_FETCH_SITE	cross-site
HTTP_UPN	ronie.s@contoso.com
HTTP_EMPLOYEEID	GC45865
HTTP_GROUP_AUTHZ	Travel Approvers Logistics

Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.8.4330.0

For increased security, block direct access to the application, enforcing a path through the BIG-IP.

Troubleshooting

Use the following guidance for troubleshooting.

Log verbosity

BIG-IP logs have information to help isolate authentication and SSO issues. Increase the log verbosity level:

1. Go to Access Policy > Overview > Event Logs.
2. Select Settings.
3. Select the row of your published application.
4. Select Edit > Access System Logs.
5. From the SSO list, select Debug.
6. Select OK.
7. Reproduce the issue.
8. Review the logs.
9. When finished, revert the settings.

BIG-IP error message

If a BIG-IP error appears after redirection, the issue likely relates to SSO from Microsoft Entra ID to the BIG-IP.

1. Navigate to **Access Policy > Overview**.
2. Select **Access reports**.
3. Run the report for the last hour.
4. Review the logs for clues.
5. For your session, select the **View session variables** link.
6. Verify the APM receives the expected claims from Microsoft Entra ID.

No BIG-IP error message

If no BIG-IP error message appears, then the issue is probably more related to SSO from the BIG-IP to the backend application.

1. Navigate to **Access Policy > Overview**.
2. Select **Active Sessions**.
3. Select the link for your active session.
4. Select the **View Variables** link to determine any SSO issues.
5. Confirm the BIG-IP APM fails or succeeds to obtain the correct user and domain identifiers.

Learn more:

- Go to devcentral.f5.com for [APM variable assign examples](#)
- Go to techdocs.f5.com for [BIG-IP Access Policy Manager: Visual Policy Editor](#)

Resources

- [Passwordless authentication](#)
- [What is Conditional Access?](#)
- [Zero Trust framework to enable remote work](#)

Tutorial: Configure F5 BIG-IP Access Policy Manager for Kerberos authentication

Article • 04/18/2024

In this tutorial, you'll learn to implement secure hybrid access (SHA) with single sign-on (SSO) to Kerberos applications by using the F5 BIG-IP advanced configuration. Enabling BIG-IP published services for Microsoft Entra SSO provides many benefits, including:

- Improved [Zero Trust](#) governance through Microsoft Entra preauthentication, and use of the Conditional Access security policy enforcement solution.
 - See, [What is Conditional Access?](#)
- Full SSO between Microsoft Entra ID and BIG-IP published services
- Identity management and access from a single control plane, the [Microsoft Entra admin center](#)

To learn more about benefits, see [Integrate F5 BIG-IP with Microsoft Entra ID](#).

Scenario description

For this scenario, you'll configure a line-of-business application for Kerberos authentication, also known as Integrated Windows Authentication.

To integrate the application with Microsoft Entra ID requires support from a federation-based protocol, such as Security Assertion Markup Language (SAML). Because modernizing the application introduces the risk of potential downtime, there are other options.

While you're using Kerberos Constrained Delegation (KCD) for SSO, you can use [Microsoft Entra application proxy](#) to access the application remotely. You can achieve the protocol transition to bridge the legacy application to the modern, identity control plane.

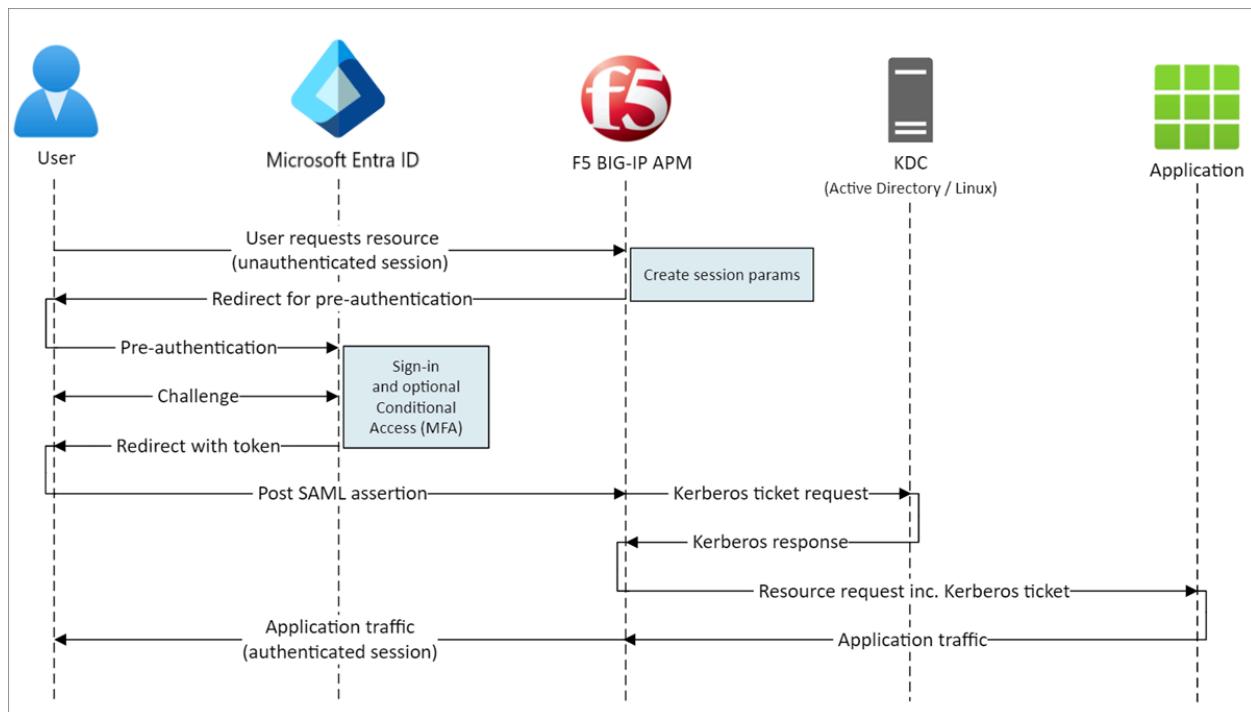
Another approach is to use an F5 BIG-IP Application Delivery Controller. This approach enables overlay of the application with Microsoft Entra preauthentication and KCD SSO. It improves the overall Zero Trust posture of the application.

Scenario architecture

The SHA solution for this scenario has the following elements:

- **Application:** Back-end Kerberos-based service externally published by BIG-IP and protected by SHA
- **BIG-IP:** Reverse proxy functionality for publishing back-end applications. The Access Policy Manager (APM) overlays published applications with SAML service provider (SP) and SSO functionality.
- **Microsoft Entra ID:** Identity provider (IdP) that verifies user credentials, Microsoft Entra Conditional Access, and SSO to the BIG-IP APM through SAML
- **KDC:** Key Distribution Center role on a domain controller (DC) that issues Kerberos tickets

The following image illustrates the SAML SP-initiated flow for this scenario, but IdP-initiated flow is also supported.



User flow

1. User connects to the application endpoint (BIG-IP)
2. BIG-IP access policy redirects the user to Microsoft Entra ID (SAML IdP)
3. Microsoft Entra ID preauthenticates the user and applies enforced Conditional Access policies
4. User is redirected to BIG-IP (SAML SP), and SSO is performed via the issued SAML token
5. BIG-IP authenticates the user and requests a Kerberos ticket from KDC

6. BIG-IP sends the request to the back-end application with the Kerberos ticket for SSO
7. Application authorizes the request and returns the payload

Prerequisites

Prior BIG-IP experience isn't necessary. You need:

- An [Azure free account ↗](#), or a higher-tier subscription.
- A BIG-IP, or [deploy BIG-IP Virtual Edition in Azure](#).
- Any of the following F5 BIG-IP licenses:
 - F5 BIG-IP Best bundle
 - F5 BIG-IP APM standalone license
 - F5 BIG-IP APM add-on license on a BIG-IP Local Traffic Manager (LTM)
 - 90-day BIG-IP [Free Trial ↗](#) license
- User identities [synchronized](#) from an on-premises directory to Microsoft Entra ID, or created in Microsoft Entra ID and flowed back to your on-premises directory.
- One of the following roles in Microsoft Entra tenant: Global Administrator, Cloud Application Administrator, or Application Administrator.
- A web server [certificate](#) for publishing services over HTTPS, or use default BIG-IP certificates while testing.
- A Kerberos application, or go to [active-directory-wp.com](#) to learn to configure [SSO with IIS on Windows ↗](#).

BIG-IP configuration methods

This article covers the advanced configuration, a flexible SHA implementing that creates BIG-IP configuration objects. You can use this approach for scenarios the Guided Configuration templates don't cover.

 **Note**

Replace all example strings or values in this article with those for your actual environment.

Register F5 BIG-IP in Microsoft Entra ID

 **Tip**

Steps in this article might vary slightly based on the portal you start from.

Before BIG-IP can hand off preauthentication to Microsoft Entra ID, register it in your tenant. This process initiates SSO between both entities. The app you create from the F5 BIG-IP gallery template is the relying party that represents the SAML SP for the BIG-IP published application.

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**, then select **New application**.
3. The **Browse Microsoft Entra Gallery** pane appears with tiles for cloud platforms, on-premises applications, and featured applications. Applications in the **Featured applications** section have icons that indicate whether they support federated SSO and provisioning.
4. In the Azure gallery, search for **F5**, and select **F5 BIG-IP APM Microsoft Entra ID integration**.
5. Enter a name for the new application to recognize the application instance.
6. Select **Add/Create** to add it to your tenant.

Enable SSO to F5 BIG-IP

Configure the BIG-IP registration to fulfill SAML tokens that the BIG-IP APM requests.

1. In the **Manage** section of the left menu, select **Single sign-on**. The **Single sign-on** pane appears.
2. On the **Select a single sign-on method** page, select **SAML**. Select **No, I'll save later** to skip the prompt.
3. On the **Set up single sign-on with SAML** pane, select the **pen** icon to edit **Basic SAML Configuration**.
4. Replace the predefined **Identifier** value with the full URL for the BIG-IP published application.
5. Replace the **Reply URL** value, but retain the path for the application's SAML SP endpoint.

 **Note**

In this configuration, the SAML flow operates in IdP-initiated mode. Microsoft Entra ID issues a SAML assertion before the user is redirected to the BIG-IP endpoint for the application.

6. To use SP-initiated mode, enter the application URL in **Sign on URL**.
7. For **Logout Url**, enter the BIG-IP APM single logout (SLO) endpoint prepended by the host header of the service being published. This action ensures the user BIG-IP APM session ends after the user signs out of Microsoft Entra ID.

The screenshot shows the 'Basic SAML Configuration' pane. It includes fields for 'Identifier (Entity ID)', 'Reply URL (Assertion Consumer Service URL)', 'Sign on URL' (with optional notes), 'Relay State' (with optional notes), and 'Logout Url'. The 'Logout Url' field contains the value 'https://myexpenses.contoso.com/saml/sp/profile/redirect/slo', which is highlighted with a red box.

① Note

From BIG-IP traffic management operating system (TMOS) v16, the SAML SLO endpoint has changed to `/saml/sp/profile/redirect/slo`.

8. Before you close the SAML configuration, select **Save**.
9. Skip the SSO test prompt.
10. Note the properties of the **User Attributes & Claims** section. Microsoft Entra ID issues properties to users for BIG-IP APM authentication and for SSO to the back-end application.
11. To save the Federation Metadata XML file to your computer, on the **SAML Signing Certificate** pane, select **Download**.

The screenshot shows the 'SAML Signing Certificate' pane. It displays properties like Status (Active), Thumbprint (9285F77 EA8660844E02A56C4FFC690 DC7A9582B1), and Expiration (19/03/2023, 2:12:02 PM). Under the 'App Federation Metadata Url' field, there are three download links: 'Download' (Base64), 'Download' (Raw), and 'Download' (Federation Metadata XML). The 'Download' link for the Federation Metadata XML is highlighted with a red box.

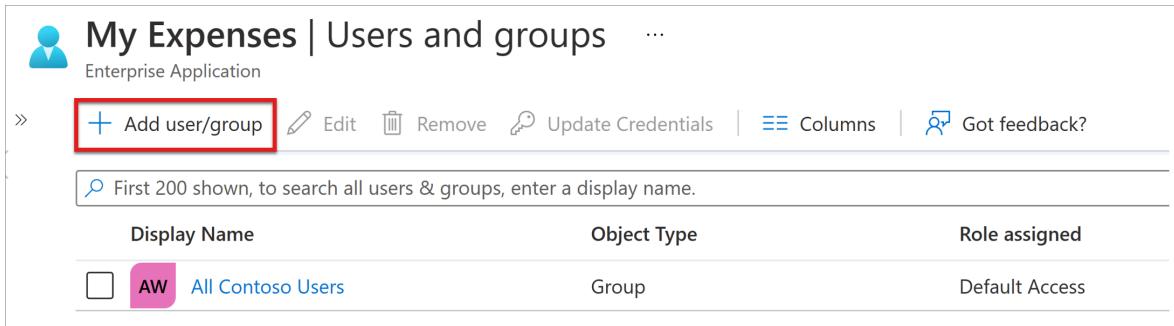
Note

SAML signing certificates that Microsoft Entra ID creates have a lifespan of three years. For more information, see [Managed certificates for federated single sign-on](#).

Grant access to users and groups

By default, Microsoft Entra ID issues tokens for users granted access to an application. To grant users and groups access to the application:

1. On the F5 BIG-IP application's overview pane, select **Assign Users and groups**.
2. Select **+ Add user/group**.



The screenshot shows the Microsoft Entra ID interface for managing users and groups. At the top, it says "My Expenses | Users and groups" and "Enterprise Application". Below the header, there's a search bar with placeholder text "First 200 shown, to search all users & groups, enter a display name." and a "Display Name" filter. A red box highlights the "Add user/group" button. The main area is a table with columns: "Display Name", "Object Type", and "Role assigned". One row is visible, showing "AW" (with a checkbox), "All Contoso Users" (with a pink background), "Group", and "Default Access".

Display Name	Object Type	Role assigned
<input type="checkbox"/> AW All Contoso Users	Group	Default Access

3. Select users and groups, and then select **Assign**.

Configure Active Directory Kerberos constrained delegation

For the BIG-IP APM to perform SSO to the back-end application on behalf of users, configure KCD in the target Active Directory (AD) domain. Delegating authentication requires you to provision the BIG-IP APM with a domain service account.

For this scenario, the application is hosted on server APP-VM-01 and runs in the context of a service account named `web_svc_account`, not the computer identity. The delegating service account assigned to the APM is F5-BIG-IP.

Create a BIG-IP APM delegation account

The BIG-IP doesn't support group Managed Service Accounts (gMSA), therefore create a standard user account for the APM service account.

1. Enter the following PowerShell command. Replace the **UserPrincipalName** and **SamAccountName** values with your environment values. For better security, use a dedicated service principal name (SPN) that matches the host header of the application.

```
New-ADUser -Name "F5 BIG-IP Delegation Account" UserPrincipalName $HOST_SPN  
SamAccountName "f5-big-ip" -PasswordNeverExpires $true Enabled $true -  
AccountPassword (Read-Host -AsSecureString "Account Password")
```

HOST_SPN = host/f5-big-ip.contoso.com@contoso.com

 **Note**

When the Host is used, any application running on the host will delegate the account whereas when HTTPS is used, it will allow only HTTP protocol-related operations.

2. Create a **Service Principal Name (SPN)** for the APM service account to use during delegation to the web application service account:

```
Set-AdUser -Identity f5-big-ip -ServicePrincipalNames @Add="host/f5-big-  
ip.contoso.com"}
```

 **Note**

It is mandatory to include the host/ part in the format of UserPrincipleName (host/name.domain@domain) or ServicePrincipleName (host/name.domain).

3. Before you specify the target SPN, view its SPN configuration. Ensure the SPN shows against the APM service account. The APM service account delegates for the web application:

- Confirm your web application is running in the computer context or a dedicated service account.
- For the Computer context, use the following command to query the account object in the Active Directory to see its defined SPNs. Replace <name_of_account> with the account for your environment.

```
Get-ADComputer -identity <name_of_account> -properties  
ServicePrincipalNames | Select-Object -ExpandProperty  
ServicePrincipalNames
```

For example: Get-ADUser -identity f5-big-ip -properties ServicePrincipalNames | Select-Object -ExpandProperty ServicePrincipalNames

- For the dedicated service account, use the following command to query the account object in Active Directory to see its defined SPNs. Replace <name_of_account> with the account for your environment.

```
Get-ADUser -identity <name_of_account> -properties ServicePrincipalNames  
| Select-Object -ExpandProperty ServicePrincipalNames
```

For example: Get-ADComputer -identity f5-big-ip -properties ServicePrincipalNames | Select-Object -ExpandProperty ServicePrincipalNames

4. If the application ran in the machine context, add the SPN to the object of the computer account in Active Directory:

```
Set-ADComputer -Identity APP-VM-01 -ServicePrincipalNames  
@{Add="http/myexpenses.contoso.com"}
```

With SPNs defined, establish trust for the APM service account delegate to that service. The configuration varies depending on the topology of your BIG-IP instance and application server.

Configure BIG-IP and the target application in the same domain

1. Set trust for the APM service account to delegate authentication:

```
Get-ADUser -Identity f5-big-ip | Set-ADAccountControl -  
TrustedToAuthForDelegation $true
```

2. The APM service account needs to know the target SPN that it's trusted to delegate to. Set the target SPN to the service account running your web application:

```
Set-ADUser -Identity f5-big-ip -Add @{ 'msDS-  
AllowedToDelegateTo'=@('HTTP/myexpenses.contoso.com') }
```

(!) Note

You can complete these tasks with the Active Directory Users and Computers, Microsoft Management Console (MMC) snap-in, on a domain controller.

Configure BIG-IP and the target application in different domains

In the Windows Server 2012 version, and higher, cross-domain KCD uses Resource-Based Constrained Delegation (RBCD). The constraints for a service are transferred from the domain administrator to the service administrator. This delegation allows the backend service administrator to allow or deny SSO. This situation creates a different approach at configuration delegation, which is possible when you use PowerShell or Active Directory Service Interfaces Editor (ADSI Edit).

You can use the `PrincipalsAllowedToDelegateToAccount` property of the application service account (computer or dedicated service account) to grant delegation from BIG-IP. For this scenario, use the following PowerShell command on a domain controller (Windows Server 2012 R2, or later) in the same domain as the application.

Use an SPN defined against a web application service account. For better security, use a dedicated SPN that matches the host header of the application. For example, because the web application host header in this example is `myexpenses.contoso.com`, add `HTTP/myexpenses.contoso.com` to the application service account object in Active Directory (AD):

```
Set-AdUser -Identity web_svc_account -ServicePrincipalNames  
@{Add="http/myexpenses.contoso.com"}
```

For the following commands, note the context.

If the `web_svc_account` service runs in the context of a user account, use these commands:

```
$big-ip= Get-ADComputer -Identity f5-big-ip -server dc.contoso.com Set-ADUser -  
Identity web_svc_account -PrincipalsAllowedToDelegateToAccount $big-ip Get-ADUser  
web_svc_account -Properties PrincipalsAllowedToDelegateToAccount
```

If the `web_svc_account` service runs in the context of a computer account, use these commands:

```
$big-ip= Get-ADComputer -Identity f5-big-ip -server dc.contoso.com Set-ADComputer  
-Identity web_svc_account -PrincipalsAllowedToDelegateToAccount $big-ip Get-  
ADComputer web_svc_account -Properties PrincipalsAllowedToDelegateToAccount
```

For more information, see [Kerberos Constrained Delegation across domains](#).

BIG-IP advanced configuration

Use the following section to continue setting up the BIG-IP configurations.

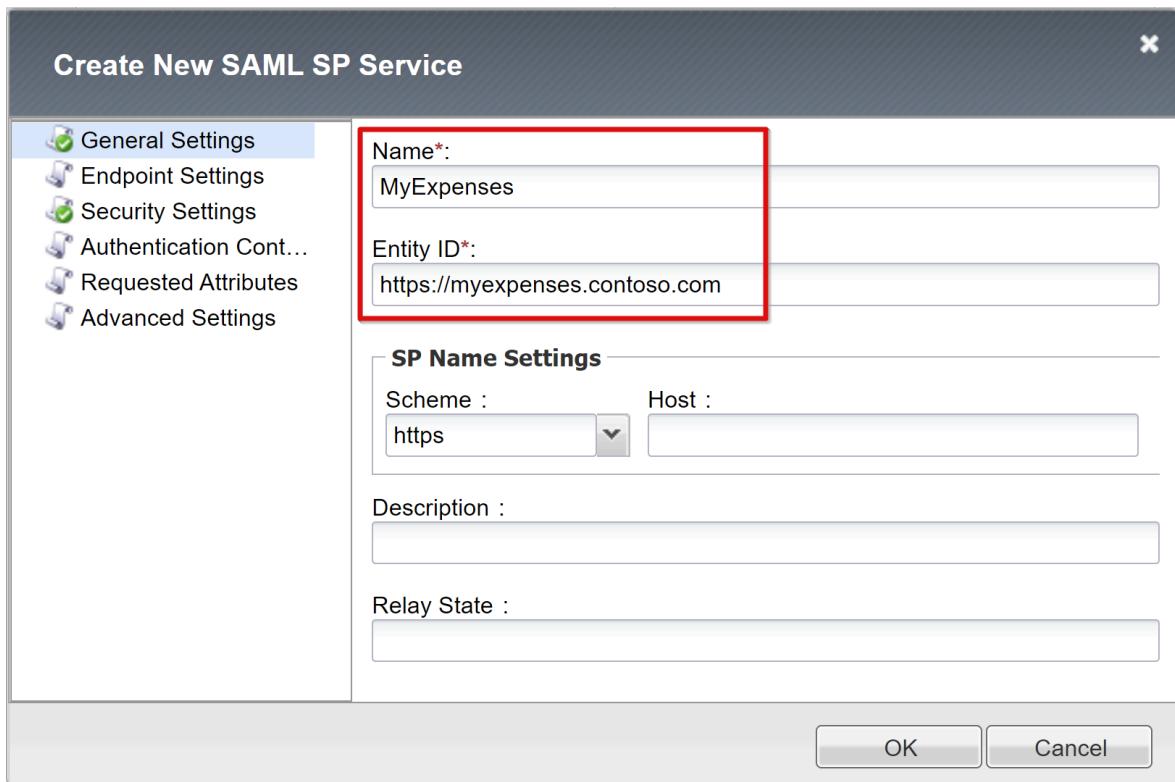
Configure SAML service provider settings

SAML service provider settings define the SAML SP properties that APM uses for overlaying the legacy application with SAML preauthentication. To configure them:

1. From a browser, sign in to the F5 BIG-IP management console.
2. Select **Access > Federation > SAML Service Provider > Local SP Services > Create**.



3. Provide the **Name** and **Entity ID** values you saved when you configured SSO for Microsoft Entra ID.



4. If the SAML entity ID is an exact match of the URL for the published application, you can skip **SP Name Settings**. For example, if the entity ID is `urn:myexpenses:contosoonline`, the **Scheme** value is **https**; the **Host** value is **myexpenses.contoso.com**. If the entity ID is "`https://myexpenses.contoso.com`", you don't need to provide this information.

Configure an external IdP connector

A SAML IdP connector defines the settings for the BIG-IP AP to trust Microsoft Entra ID as its SAML IdP. These settings map the SAML SP to a SAML IdP, establishing the federation trust between the AP and Microsoft Entra ID. To configure the connector:

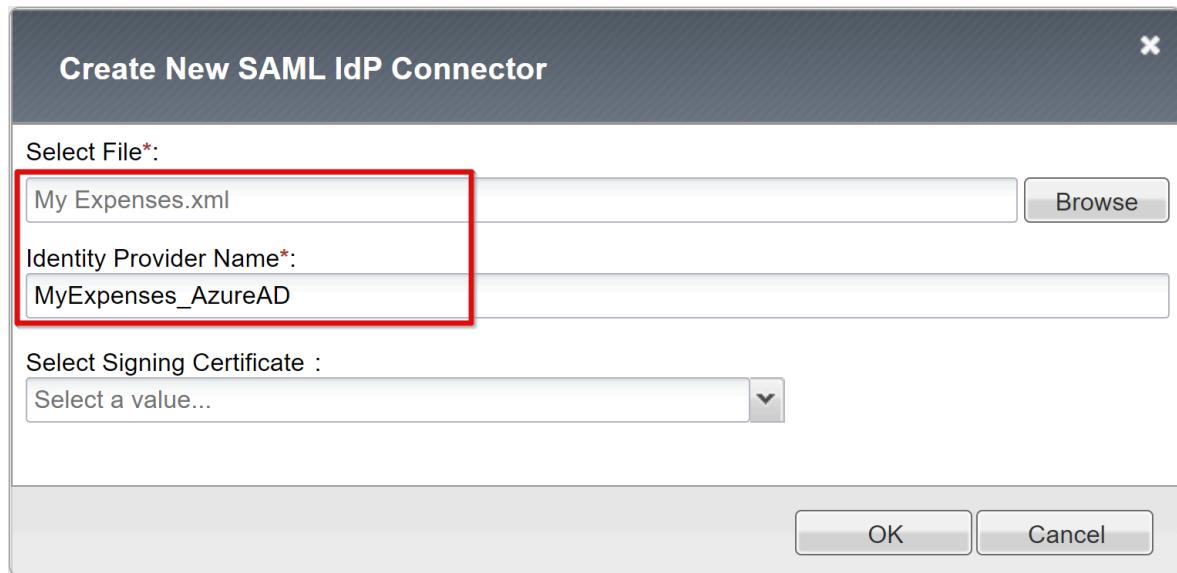
1. Scroll down to select the new SAML SP object, and then select **Bind/Unbind IdP Connectors**.

Name	Description	Partition
MyExpenses	MyExpenses BIG-IP SAML Service Provider	Common

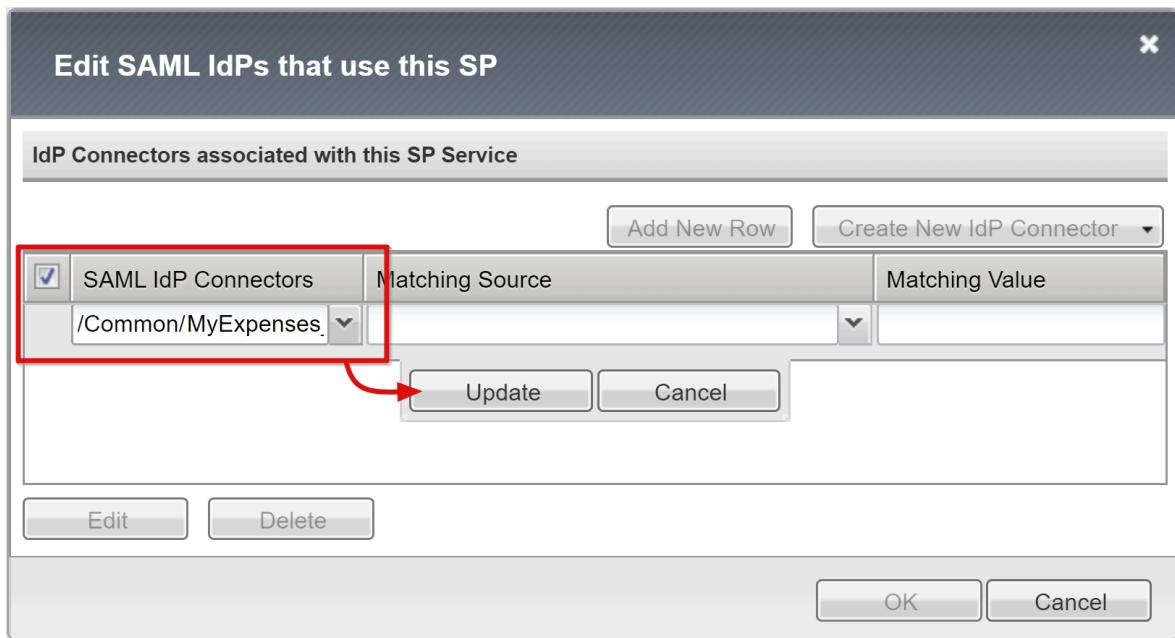
2. Select **Create New IdP Connector > From Metadata**.



3. Browse to the federation metadata XML file you downloaded, and provide an **Identity Provider Name** for the APM object that represents the external SAML IdP. The following example shows **MyExpenses_AzureAD**.



4. Select **Add New Row** to choose the new **SAML IdP Connectors** value, and then select **Update**.



5. Select OK.

Configure Kerberos SSO

Create an APM SSO object for KCD SSO to back-end applications. Use the APM delegation account that you created.

1. Select Access > Single Sign-on > Kerberos > Create and provide the following information:

- **Name:** After you create it, other published applications can use the Kerberos SSO APM object. For example, use Contoso_KCD_sso for multiple published applications for the Contoso domain. Use MyExpenses_KCD_sso for a single application.
- **Username Source:** Specify the user ID source. Use an APM session variable as the source. Use of `session.saml.last.identity` is advised because it contains the logged-in user ID from the Microsoft Entra claim.
- **User Realm Source:** Required when the user domain differs from the Kerberos realm for KCD. If users are in a separate trusted domain, you make the APM aware by specifying the APM session variable with the logged-in user domain. An example is `session.saml.last.attr.name.domain`. You do this action in scenarios when the user principal name (UPN) is based on an alternative suffix.
- **Kerberos Realm:** User domain suffix in uppercase
- **KDC:** Domain controller IP address. Or enter a fully qualified domain name if DNS is configured and efficient.

- **UPN Support:** Select this checkbox if the source for username is in UPN format, for instance the session.saml.last.identity variable.
- **Account Name and Account Password:** APM service account credentials to perform KCD
- **SPN Pattern:** If you use HTTP/%h, APM uses the host header of the client request to build the SPN for which it's requesting a Kerberos token.
- **Send Authorization:** Disable this option for applications that prefer negotiating authentication, instead of receiving the Kerberos token in the first request (for example, Tomcat).

Access » Single Sign-On » New SSO Configuration...

General Properties:	
Name	Contoso_KCD_sso
SSO Method	Kerberos
Log Settings	+ From Access Profile

Credentials Source	
Username Source	session.saml.last.identity
User Realm Source	

SSO Method Configuration	
Kerberos Realm	CONTOSO.COM
KDC	172.16.76.4
UPN Support	<input checked="" type="checkbox"/> Enable
Account Name	host/f5-big-ip.contoso.com@CONTOSO.COM
Account Password
Confirm Account Password
SPN Pattern	HTTP/%h
Ticket Lifetime	600
Send Authorization	Always

You can leave KDC undefined if the user realm is different from the back-end server realm. This rule applies to multiple-domain realm scenarios. If you leave KDC undefined, BIG-IP tries to discover a Kerberos realm through a DNS lookup of SRV records for the back-end server domain. It expects the domain name to be the same as the realm name. If the domain name differs, specify it in the [/etc/krb5.conf](#) file.

Kerberos SSO processing is faster when an IP address specifies a KDC. Kerberos SSO processing is slower if a host name specifies a KDC. Because of more DNS queries, processing is slower when a KDC is undefined. Ensure your DNS is performing optimally before moving a proof-of-concept into production.

 **Note**

If back-end servers are in multiple realms, create a separate SSO configuration object for each realm.

You can inject headers as part of the SSO request to the back-end application. Change the **General Properties** setting from **Basic** to **Advanced**.

For more information on configuring an APM for KCD SSO, see the F5 article [K17976428: Overview of Kerberos constrained delegation ↗](#).

Configure an access profile

An access profile binds APM elements that manage access to BIG-IP virtual servers. These elements include access policies, SSO configuration, and UI settings.

1. Select **Access > Profiles / Policies > Access Profiles (Per-Session Policies) > Create** and enter the following properties:

- **Name:** For example, enter MyExpenses
- **Profile Type:** Select All
- **SSO Configuration:** Select the KCD SSO configuration object you created
- **Accepted Languages:** Add at least one language

Access » Profiles / Policies : Access Profiles (Per-Session Policies) » New Profile...

General Properties

Name	MyExpenses
Parent Profile	access
Profile Type	All
Profile Scope	Profile
Customization Type	Modern

Settings Custom

Configurations

SSO Across Authentication Domains (Single Domain mode)

Domain Cookie	
Cookie Options	<input checked="" type="checkbox"/> Secure <input type="checkbox"/> Persistent <input type="checkbox"/> HTTP Only <input type="checkbox"/> Samesite
SSO Configuration	Contoso_KCD_sso

Language Settings

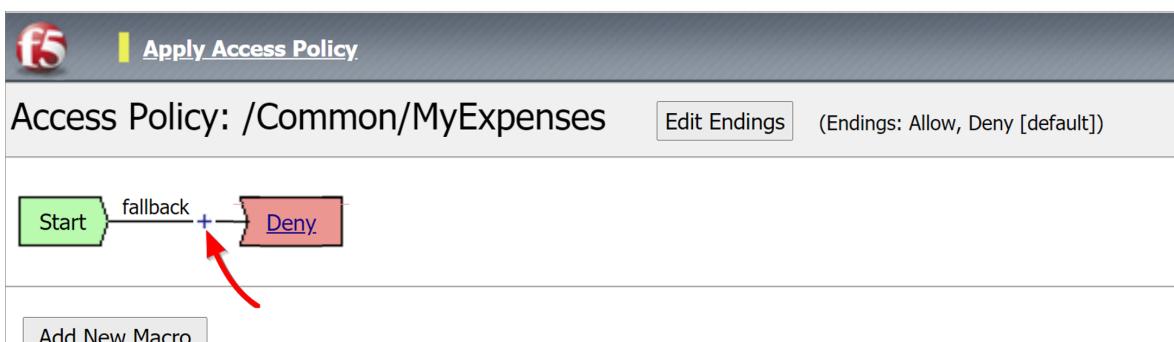
Additional Languages	Afar (aa) <input type="button" value="Add"/>
Accepted Languages	
<input type="checkbox"/> English (en)	
Languages	
Default Language	English (en) <input type="button" value=""/>
Factory BuiltIn Languages	
<input type="checkbox"/> Japanese (ja) <input type="checkbox"/> Chinese (Simplified) (zh-cn), <input type="checkbox"/> Korean (ko) <input type="checkbox"/> Spanish (es) <input type="checkbox"/> French (fr) <input type="checkbox"/> German (de)	

2. For the per-session profile you created, select **Edit**.

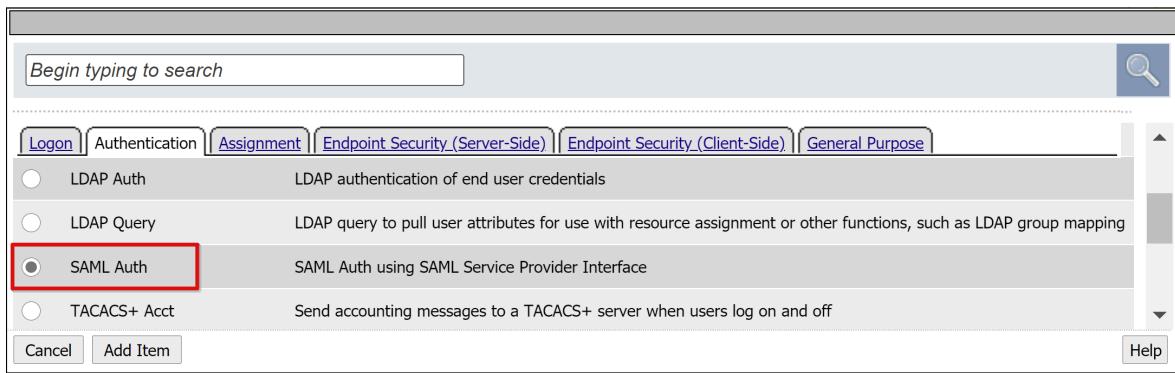
Access » Profiles / Policies : Access Profiles (Per-Session Policies)

Status	Access Profile Name	Application	Profile Type	Per-Session Policy	Export	Copy	Customization	Logs	Virtual Servers
<input type="checkbox"/>	MyExpenses	All	<input type="checkbox"/> Edit... <input type="button" value="Edit..."/>		<input type="button" value="Export..."/>	<input type="button" value="Copy..."/>	Modern		default-log-setting
<input type="checkbox"/>	access	All	(none)	(none)	(none)	(none)			

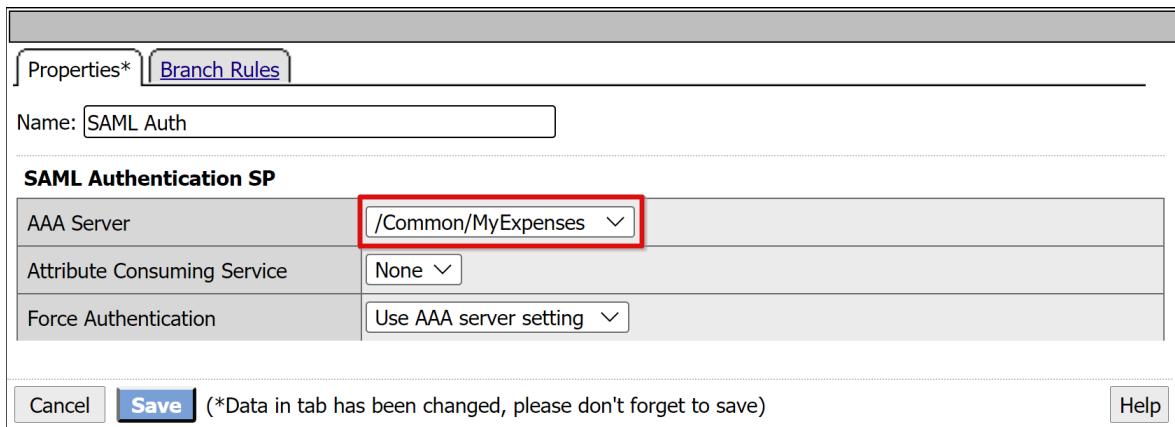
3. The visual policy editor opens. Select the plus sign next to the fallback.



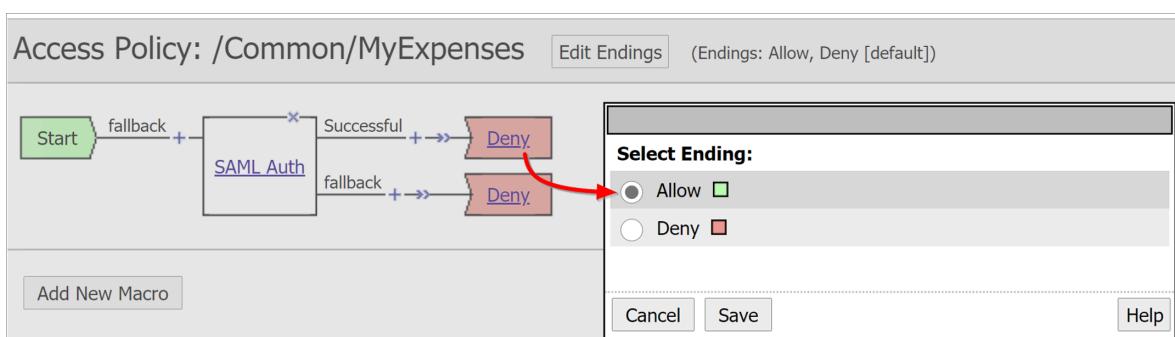
4. In the dialog, select **Authentication > SAML Auth > Add Item**.



5. In the **SAML authentication SP** configuration, set the **AAA Server** option to use the SAML SP object you created.



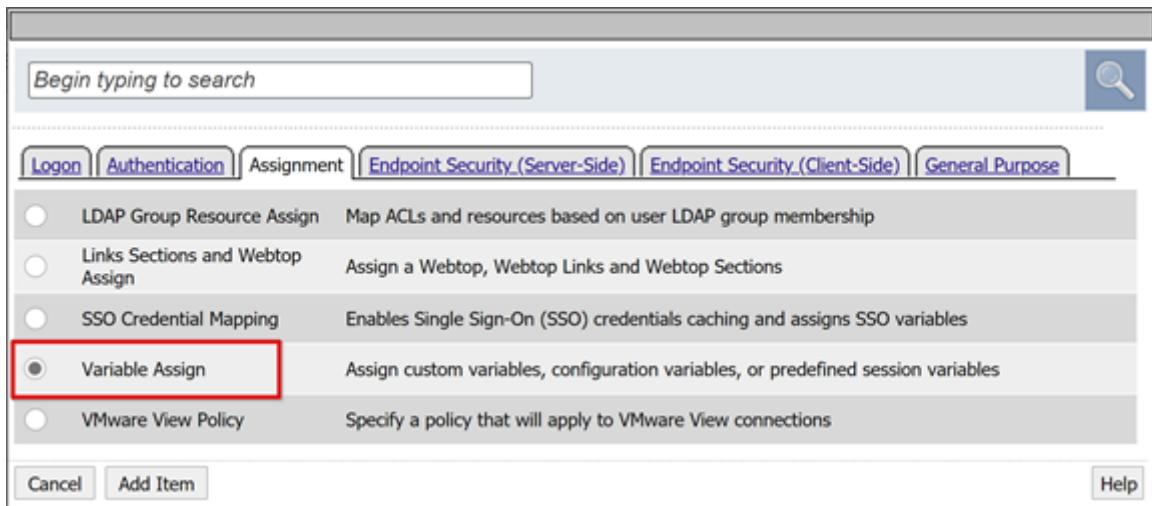
6. To change the **Successful** branch to **Allow**, select the link in the upper **Deny** box.
7. Select **Save**.



Configure attribute mappings

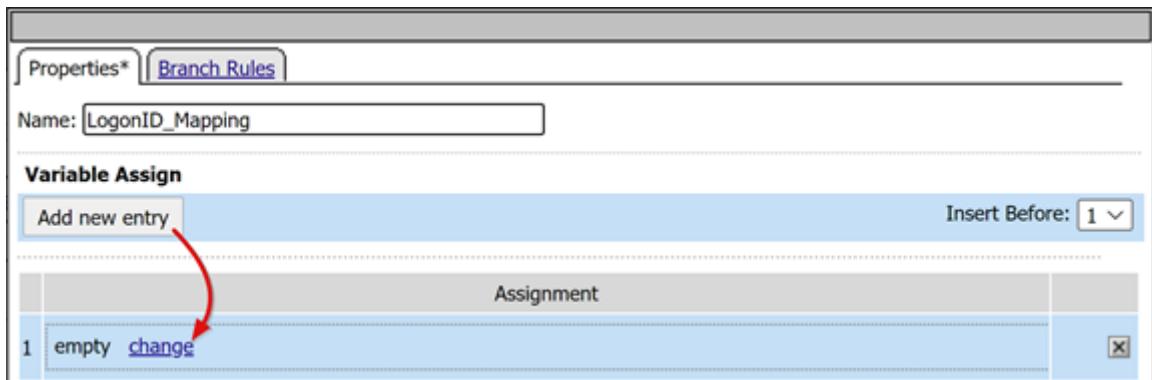
Although it's optional, you can add a **LogonID_Mapping** configuration to enable the BIG-IP active sessions list to display the UPN of the logged-in user, instead of a session number. This information is useful for analyzing logs or troubleshooting.

1. For the **SAML Auth Successful** branch, select the **plus sign**.
2. In the dialog, select **Assignment > Variable Assign > Add Item**.



3. Enter a Name.

4. On the **Variable Assign** pane, select **Add new entry > change**. The following example shows LogonID_Mapping in the Name box.



5. Set both variables:

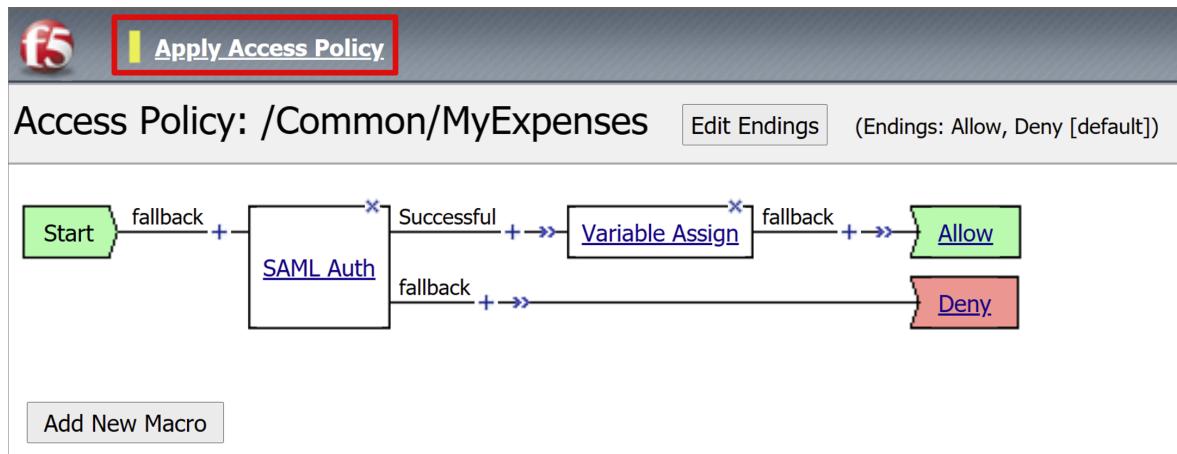
- **Custom Variable:** Enter session.logon.last.username
- **Session Variable:** Enter session.saml.last.identity

6. Select **Finished > Save**.

7. Select the **Deny** terminal of the access policy **Successful** branch. Change it to **Allow**.

8. Select **Save**.

9. Select **Apply Access Policy**, and close the editor.



Configure the back-end pool

For BIG-IP to forward client traffic accurately, create a BIG-IP node object that represents the back-end server hosting your application. Then, place that node in a BIG-IP server pool.

1. Select **Local Traffic > Pools > Pool List > Create** and provide a name for a server pool object. For example, enter `MyApps_VMs`.

Configuration: Basic	
Name	MyApps_VMs
Description	
Health Monitors	Active: Available: /Common gateway_icmp http http2 http2_head_f5

2. Add a pool member object with the following resource details:

- **Node Name:** Display name for the server hosting the back-end web application
- **Address:** IP address of the server hosting the application
- **Service Port:** HTTP/S port the application is listening on

Resources

Load Balancing Method	Round Robin									
Priority Group Activation	Disabled									
New Members	<input checked="" type="radio"/> New Node <input type="radio"/> New FQDN Node <input type="radio"/> Node List Node Name: MyExpenses_VM1 (Optional) Address: 172.16.76.16 Service Port: 80 HTTP									
	<input type="button" value="Add"/> <table border="1"> <thead> <tr> <th>Node Name</th> <th>Address/FQDN</th> <th>Service Port</th> <th>Auto Populate</th> <th>Priority</th> </tr> </thead> <tbody> <tr> <td>MyExpenses_VM1</td> <td>172.16.76.16</td> <td>80</td> <td></td> <td>0</td> </tr> </tbody> </table> <input type="button" value="Edit"/> <input type="button" value="Delete"/>	Node Name	Address/FQDN	Service Port	Auto Populate	Priority	MyExpenses_VM1	172.16.76.16	80	
Node Name	Address/FQDN	Service Port	Auto Populate	Priority						
MyExpenses_VM1	172.16.76.16	80		0						
<input type="button" value="Cancel"/> <input type="button" value="Repeat"/> <input type="button" value="Finished"/>										

(!) Note

This article doesn't cover the additional configuration health monitors require. See, [K13397: Overview of HTTP health monitor request formatting for the BIG-IP DNS system ↗](#).

Configure the virtual server

A virtual server is a BIG-IP data plane object represented by a virtual IP address listening for client requests to the application. Received traffic is processed and evaluated against the APM access profile associated with the virtual server, before being directed according to policy.

To configure a virtual server:

1. Select Local Traffic > Virtual Servers > Virtual Server List > Create.
2. Enter a **Name** and an IPv4/IPv6 address not allocated to a BIG-IP object, or device, on the connected network. The IP address is dedicated to receive client traffic for the published back-end application.
3. Set **Service Port** to 443.

General Properties

Name	<input type="text" value="MyExpenses"/>
Description	<input type="text" value="APM Listener for MyExpenses web application"/>
Type	Standard <input type="button" value="▼"/>
Source Address	<input checked="" type="radio"/> Host <input type="radio"/> Address List <input type="text"/>
Destination Address/Mask	<input checked="" type="radio"/> Host <input type="radio"/> Address List <input type="text" value="172.16.76.26"/>
Service Port	<input checked="" type="radio"/> Port <input type="radio"/> Port List <input type="text" value="443"/> <input type="button" value="HTTPS"/> <input type="button" value="▼"/>
Notify Status to Virtual Address	<input checked="" type="checkbox"/>
State	Enabled <input type="button" value="▼"/>

4. Set **HTTP Profile (Client)** to http.
5. Enable a virtual server for Transport Layer Security (TLS) to allow services to be published over HTTPS.
6. For **SSL Profile (Client)**, select the profile you created for the prerequisites. Or use the default if you're testing.

Configuration: Basic

DoH Profile Type	None
Protocol	TCP
Protocol Profile (Client)	tcp
Protocol Profile (Server)	(Use Client Profile)
HTTP Profile (Client)	http
HTTP Profile (Server)	(Use Client Profile)
HTTP Proxy Connect Profile	None
FTP Profile	None
RTSP Profile	None
PPTP Profile	None
SSL Profile (Client)	<div style="display: flex; align-items: center;"> <div style="flex: 1;"> <p>Selected</p> <ul style="list-style-type: none"> /Common Contoso_clientssl </div> <div style="margin: 0 10px;"> << >> </div> <div style="flex: 1;"> <p>Available</p> <ul style="list-style-type: none"> /Common F5Demo F5DemoClient clientssl clientssl-insecure-compatible clientssl-quic clientssl-secure </div> </div>

7. Change Source Address Translation to Auto Map.

Source Address Translation	Auto Map
----------------------------	----------

8. Under Access Policy, set Access Profile based on the profile you created. This selection binds the Microsoft Entra SAML preauthentication profile and KCD SSO policy to the virtual server.

Access Policy

Access Profile	MyExpenses
Connectivity Profile	None
Per-Request Policy	None
VDI Profile	None
Application Tunnels (Java & Per-App VPN)	<input type="checkbox"/> Enabled
OAM Support	<input type="checkbox"/> Enabled
ADFS Proxy	<input type="checkbox"/> Enabled
PingAccess Profile	None

9. Set Default Pool to use the back-end pool objects you created in the previous section.

10. Select Finished.

The screenshot shows the 'Resources' section of the F5 BIG-IP configuration interface. It includes sections for 'iRules', 'Policies', and session management settings. The session management settings include fields for 'Default Pool', 'Default Persistence Profile', and 'Fallback Persistence Profile'. The 'Default Pool' field has a dropdown menu open, showing 'MyApps_VMs' selected. This dropdown is highlighted with a red box. To the right of the dropdown, there are two lists: 'Enabled' and 'Available', both containing a list of objects starting with '/Common' and '_sys_APM_ExchangeSupport_OA'. Below the dropdown are 'Up' and 'Down' buttons.

Configure session management settings

BIG-IP session-management settings define the conditions for which user sessions are terminated or allowed to continue, limits for users and IP addresses, and error pages. You can create policy here.

Go to **Access Policy > Access Profiles > Access Profile** and select an application from the list.

If you defined a Single Logout URI value in Microsoft Entra ID, it ensures an IdP-initiated sign out from the MyApps portal ends the session between the client and the BIG-IP APM. The imported application federation metadata XML file provides the APM with the Microsoft Entra SAML sign-out endpoint for SP-initiated sign out. For effective results, the APM needs to know when a user signs out.

Consider a scenario when a BIG-IP web portal isn't used. The user can't instruct the APM to sign out. Even if the user signs out of the application, BIG-IP is oblivious, so the application session could be reinstated through SSO. SP-initiated sign-out needs consideration to ensure sessions terminate securely.

ⓘ Note

You can add an SLO function to your application Sign-out button. This function redirects your client to the Microsoft Entra SAML sign-out endpoint. Find the SAML sign-out endpoint at **App Registrations > Endpoints**.

If you can't change the app, consider having BIG-IP listen for the app sign-out call. When it detects the request, it triggers SLO.

For more information, see the F5 articles:

- [K42052145: Configuring automatic session termination \(logout\) based on a URI-referenced file name ↗](#)
- [K12056: Overview of the Logout URI Include option ↗](#).

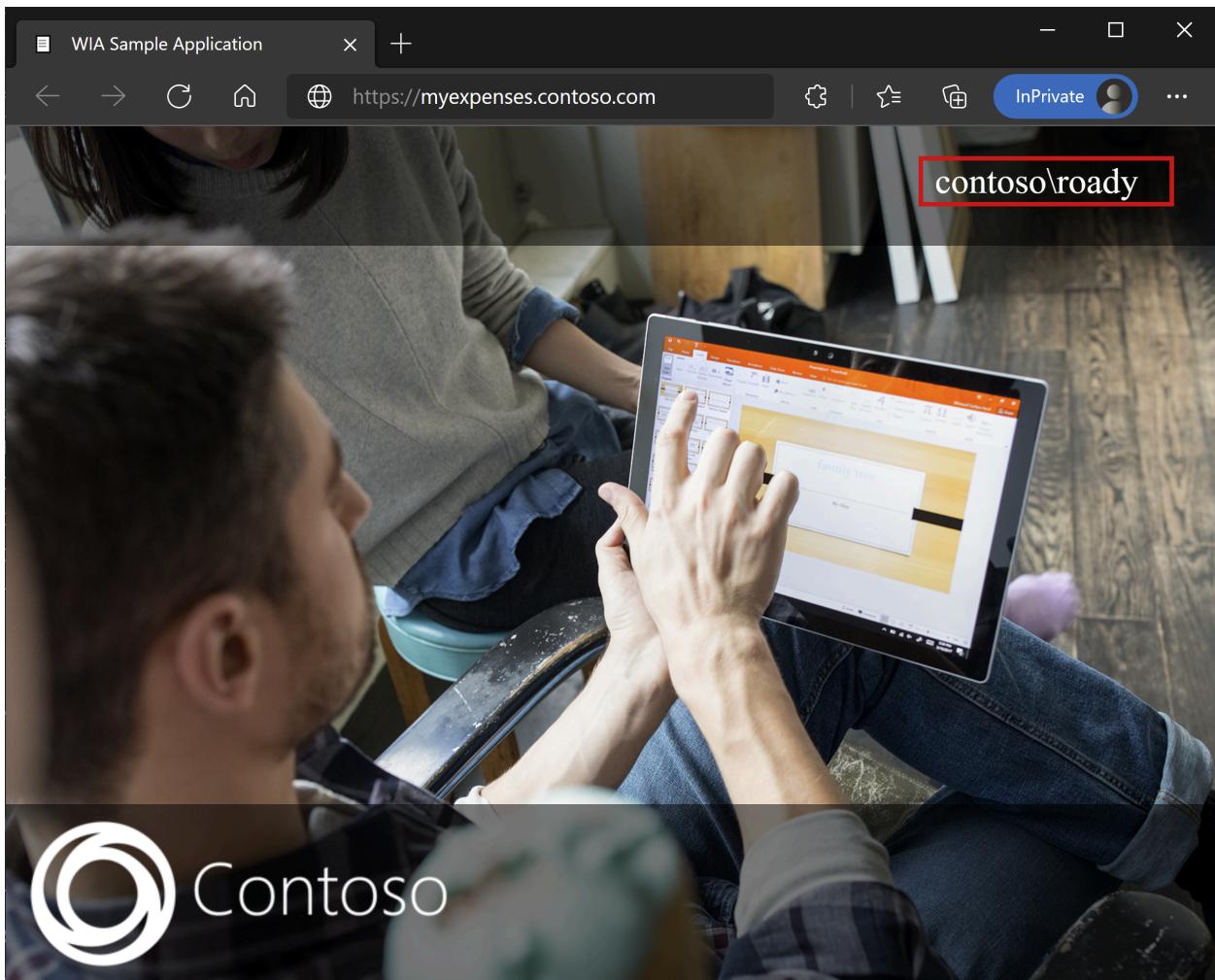
Summary

Your application is published and accessible via SHA, by its URL or through Microsoft application portals. The application is visible as a target resource in [Microsoft Entra Conditional Access](#).

For increased security, organizations that use this pattern can block direct access to the application, which forces a strict path through BIG-IP.

Next steps

As a user, open a browser and connect to the application external URL. You can select the application icon in the [Microsoft MyApps portal ↗](#). After you authenticate against your Microsoft Entra tenant, you're redirected to the BIG-IP endpoint for the application and signed in via SSO.



Microsoft Entra B2B guest access

SHA supports [Microsoft Entra B2B guest access](#). Guest identities are synchronized from your Microsoft Entra tenant to your target Kerberos domain. Have a local representation of guest objects for BIG-IP to perform KCD SSO to the back-end application.

Troubleshooting

While troubleshooting, consider the following points:

- Kerberos is time sensitive. It requires servers and clients set to the correct time and, when possible, synchronized to a reliable time source.
- Ensure the host names for the domain controller and web application are resolvable in DNS
- Ensure there are no duplicate SPNs in your environment. Run the following query at the command line: `setspn -q HTTP/my_target_SPN`.

Note

To validate an IIS application is configured for KCD, see [Troubleshoot Kerberos constrained delegation configurations for Application Proxy](#). See also the AskF5 article, [Kerberos Single Sign-On Method](#).

Increase log verbosity

BIG-IP logs are a reliable source of information. To increase the log verbosity level:

1. Go to **Access Policy > Overview > Event Logs > Settings**.
2. Select the row for your published application.
3. Select **Edit > Access System Logs**.
4. Select **Debug** from the SSO list.
5. Select **OK**.

Reproduce your problem before you look at the logs. Then revert this feature, when finished. Otherwise the verbosity is significant.

BIG-IP error

If a BIG-IP error appears after Microsoft Entra preauthentication, the problem might relate to SSO, from Microsoft Entra ID to BIG-IP.

1. Go to **Access > Overview > Access reports**.
2. To see if logs have any clues, run the report for the last hour.
3. Use the **View session variables** link for your session to understand if the APM receives the expected claims from Microsoft Entra ID.

Back-end request

If no BIG-IP error appears, the issue is probably related to the back-end request, or related to SSO from BIG-IP to the application.

1. Go to **Access Policy > Overview > Active Sessions**.
2. Select the link for your active session.
3. Use the **View Variables** link to determine root-cause KCD problems, particularly if the BIG-IP APM fails to get the right user and domain identifiers.

For help with diagnosing KCD-related problems, see archived the F5 BIG-IP deployment guide [Configuring Kerberos Constrained Delegation](#).

Resources

- My F5 article, [Active Directory Authentication](#)
- [Forget passwords, go passwordless](#)

- What is Conditional Access?
- Zero Trust framework to enable remote work ↗

Configure F5 BIG-IP Access Policy Manager for form-based SSO

Article • 06/28/2024

Learn to configure F5 BIG-IP Access Policy Manager (APM) and Microsoft Entra ID for secure hybrid access (SHA) to form-based applications. BIG-IP published services for Microsoft Entra single sign-on (SSO) has benefits:

- Improved Zero Trust governance through Microsoft Entra preauthentication and Conditional Access
 - See [What is Conditional Access?](#)
 - See [Zero Trust security](#)
- Full SSO between Microsoft Entra ID and BIG-IP published services
- Managed identities and access from one control plane
 - See the [Microsoft Entra admin center](#) ↗

Learn more:

- [Integrate F5 BIG-IP with Microsoft Entra ID](#)
- [Enable SSO for an enterprise application](#)

Scenario description

For the scenario, there's an internal legacy application configured for form-based authentication (FBA). Ideally, Microsoft Entra ID manages application access, because legacy lacks modern authentication protocols. Modernization takes time and effort, introducing the risk of downtime. Instead, deploy a BIG-IP between the public internet and the internal application. This configuration gates inbound access to the application.

With a BIG-IP in front of the application, you can overlay the service with Microsoft Entra preauthentication and header-based SSO. The overlay improves application security posture.

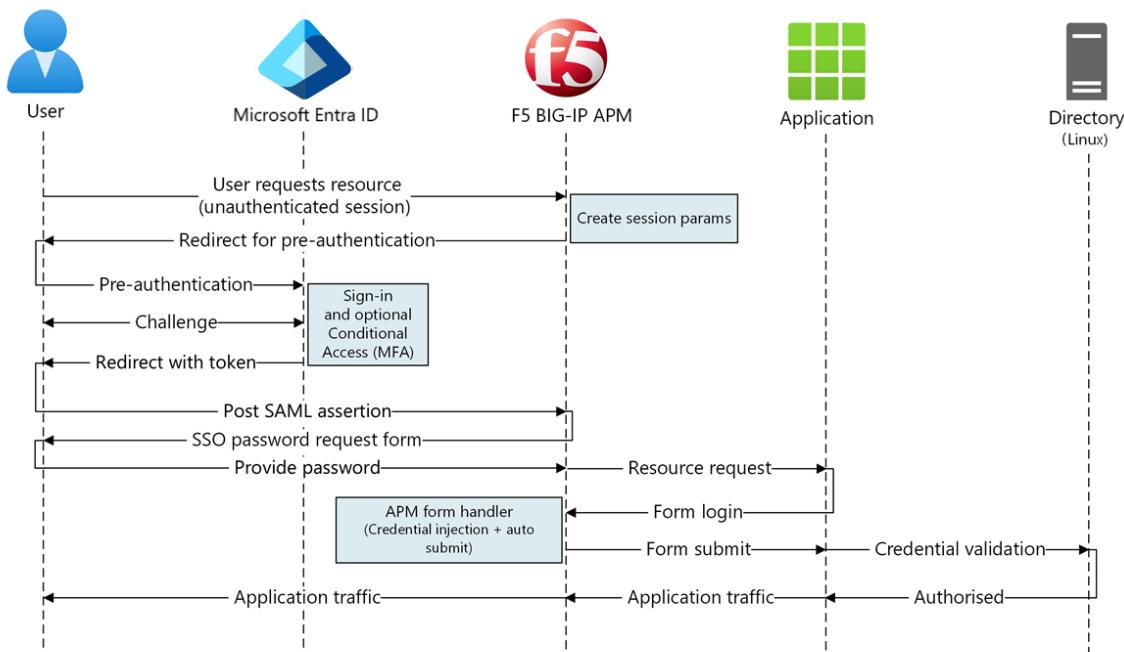
Scenario architecture

The SHA solution has the following components:

- **Application** - BIG-IP published service protected by SHA.
 - The application validates user credentials
 - Use any directory, open source, and so on

- **Microsoft Entra ID** - Security Assertion Markup Language (SAML) identity provider (IdP) that verifies user credentials, Conditional Access, and SSO to the BIG-IP.
 - With SSO, Microsoft Entra ID provides attributes to the BIG-IP, including user identifiers
- **BIG-IP** - reverse-proxy and SAML service provider (SP) to the application.
 - BIG-IP delegating authentication to the SAML IdP then performs header-based SSO to the back-end application.
 - SSO uses the cached user credentials against other forms-based authentication applications

SHA supports SP- and IdP-initiated flows. The following diagram illustrates the SP-initiated flow.



Prerequisites

You need the following components:

- An Azure subscription
 - If you don't have one, get an [Azure free account](#)
- One of the following roles: Cloud Application Administrator, or Application Administrator
- A BIG-IP or deploy a BIG-IP Virtual Edition (VE) in Azure
 - See [Deploy F5 BIG-IP Virtual Edition Virtual Machine in Azure](#)
- Any of the following F5 BIG-IP licenses:
 - F5 BIG-IP® Best bundle
 - F5 BIG-IP Access Policy Manager™ (APM) standalone license
 - F5 BIG-IP Access Policy Manager™ (APM) add-on license on a BIG-IP F5 BIG-IP® Local Traffic Manager™ (LTM)
 - 90-day BIG-IP full feature trial. See [Free Trials](#)
- User identities synchronized from an on-premises directory to Microsoft Entra ID
 - See [Microsoft Entra Connect Sync: Understand and customize synchronization](#)
- An SSL certificate to publish services over HTTPS, or use default certificates while testing
 - See [SSL profile](#)
- A form-based authentication application, or set up an Internet Information Services (IIS) form-based authentication (FBA) app for testing
 - See [Forms-based authentication](#)

BIG-IP configuration

The configuration in this article is a flexible SHA implementation: manual creation of BIG-IP configuration objects. Use this approach for scenarios the Guided Configuration templates don't cover.

Note

Replace example strings or values with those from your environment.

Register F5 BIG-IP in Microsoft Entra ID

Tip

Steps in this article might vary slightly based on the portal you start from.

BIG-IP registration is the first step for SSO between entities. The app you create from the F5 BIG-IP gallery template is the relying party, representing the SAML SP for the BIG-IP

published application.

1. Sign in to the Microsoft Entra admin center [↗](#) as at least a **Cloud Application Administrator**.
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. In the **All applications** pane, select **New application**.
4. The **Browse Microsoft Entra Gallery** pane opens.
5. Tiles appear for cloud platforms, on-premises applications, and featured applications. **Featured applications** icons indicate support of federated SSO and provisioning.
6. In the Azure gallery, search for **F5**.
7. Select **F5 BIG-IP APM Microsoft Entra ID integration**.
8. Enter a **Name** the new application uses to recognize the application instance.
9. Select **Add**.
10. Select **Create**.

Enable SSO to F5 BIG-IP

Configure the BIG-IP registration to fulfill SAML tokens that BIG-IP APM requests.

1. In left menu, in the **Manage** section, select **Single sign-on**.
2. The **Single sign-on** pane appears.
3. On the **Select a single sign-on method** page, select **SAML**.
4. Select **No, I'll save later**.
5. On the **Set up single sign-on with SAML** pane, select the **pen** icon.
6. For **Identifier**, replace the value with the BIG-IP published application URL.
7. For **Reply URL**, replace the value, but retain the path for the application SAML SP endpoint. With this configuration, SAML flow operates in IdP-initiated mode.
8. Microsoft Entra ID issues a SAML assertion, then the user is redirected to the BIG-IP endpoint.
9. For SP-initiated mode, for **Sign on URL**, enter the application URL.
10. For **Logout Url**, enter the BIG-IP APM single logout (SLO) endpoint prepended by the service host header.
11. Then, BIG-IP APM user sessions end when users sign out of Microsoft Entra ID.
12. Select **Save**.
13. Close the SAML configuration pane.
14. Skip the SSO test prompt.
15. Make a note of the **User Attributes & Claims** section properties. Microsoft Entra ID issues the properties for BIG-IP APM authentication, and SSO to the back-end application.
16. On the **SAML Signing Certificate** pane, select **Download**.

17. The Federation Metadata XML file is saved to your computer.

1 Basic SAML Configuration

Identifier (Entity ID) https://myvacation.contoso.com
Reply URL (Assertion Consumer Service URL) https://myvacation.contoso.com/saml/sp/profile/post/acs
Sign on URL *Optional*
Relay State *Optional*
Logout Url https://myvacation.contoso.com/saml/sp/profile/redirect/slr

① Note

From Traffic Management Operating System (TMOS) v16 onward, the SAML SLO endpoint is `/saml/sp/profile/redirect/slo`.

3 SAML Signing Certificate

Status Active
Thumbprint AB1AEB45AF1C26828764D87A450C4984F114CF3A
Expiration 8/9/2024, 6:12:20 PM
Notification Email rodney@contoso.com
App Federation Metadata Url https://login.microsoftonline.com/4e95c47f-b5... 
Certificate (Base64) [Download](#)
Certificate (Raw) [Download](#)
Federation Metadata XML [Download](#)

① Note

Microsoft Entra SAML signing certificates have a lifespan of three years.

Learn more: [Tutorial: Manage certificates for federated single sign-on](#)

Assign users and groups

Microsoft Entra ID issues tokens for users granted access to an application. To grant specific users and groups application access:

1. On the F5 BIG-IP application's overview pane, select **Assign Users and groups**.
2. Select **+ Add user/group**.
3. Select the users and groups you want.
4. Select **Assign**.

BIG-IP advanced configuration

Use the following instructions to configure BIG-IP.

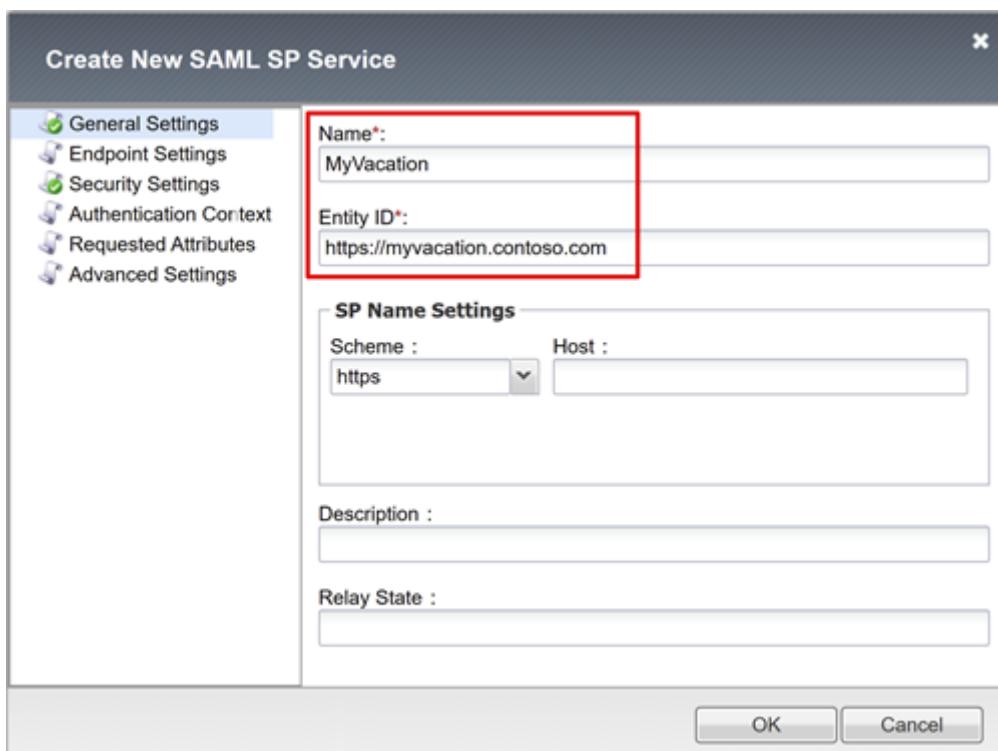
Configure SAML service provider settings

SAML SP settings define the SAML SP properties that the APM uses to overlay the legacy application with SAML preauthentication. To configure them:

1. Select Access > Federation > SAML Service Provider.
2. Select Local SP Services.
3. Select Create.



4. On **Create New SAML SP Service**, for **Name** and **Entity ID**, enter the defined name and entity ID.



Note

SP Name Settings values are required if the entity ID doesn't match the hostname portion of the published URL. Or, values are required if the entity ID isn't in regular hostname-based URL format.

5. If the entity ID is `urn:myvacation:contosoonline`, enter the application external scheme and hostname.

Configure an external IdP connector

A SAML IdP connector defines settings for the BIG-IP APM to trust Microsoft Entra ID as its SAML IdP. The settings connect the SAML service provider to a SAML IdP, which establishes the federation trust between the APM and Microsoft Entra ID.

To configure the connector:

1. Select the new SAML service provider object.
2. Select Bind/UnBind IdP Connectors.

The screenshot shows the 'Federation : SAML Service Provider : Local SP Services' configuration page. The 'SAML Service Provider' tab is selected. In the main table, there is one entry: 'MyVacation' with 'MyVacation' in the 'Description' column and 'Common' in the 'Partition' column. At the bottom of the table, there are buttons for 'Edit', 'Delete', 'Bind/Unbind IdP Connectors', and 'Export Metadata'. The 'Bind/Unbind IdP Connectors' button is highlighted with a red box.

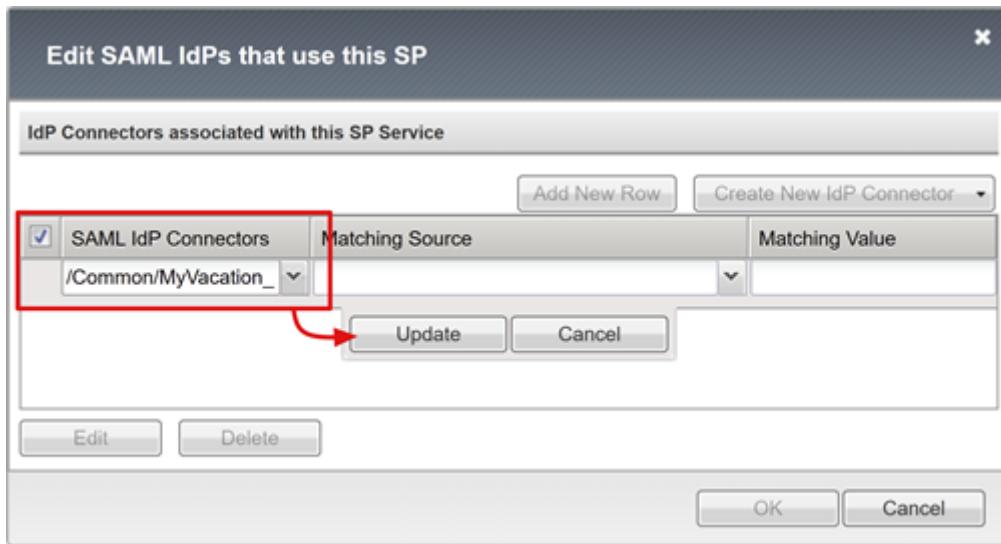
3. In the Create New IdP Connector list, select From Metadata.

The screenshot shows the 'Edit SAML IdPs that use this SP' dialog. In the top right corner, there is a 'Create New IdP Connector' dropdown menu. The 'From Metadata' option is highlighted with a red box. Other options in the menu include 'Custom' and 'From Template'. At the bottom of the dialog, there are 'OK' and 'Cancel' buttons.

4. On the **Create New SAML IdP Connector** pane, browse for the Federation Metadata XML file you downloaded.
5. Enter an **Identity Provider Name** for the APM object that represents the external SAML IdP. For example, MyVacation_EntralID.



6. Select **Add New Row**.
7. Select the new **SAML IdP Connector**.
8. Select **Update**.



9. Select **OK**.



Configure forms-based SSO

Create an APM SSO object for FBA SSO to back-end applications.

Perform FBA SSO in client-initiated mode or BIG-IP-initiated mode. Both methods emulate a user sign-on by injecting credentials into the username and password tags. The form is submitted. Users provide password to access an FBA application. The password is cached and reused for other FBA applications.

1. Select Access > Single Sign-on.
2. Select Forms Based.
3. Select Create.
4. For Name, enter a descriptive name. For example, Contoso\FBA\sso.
5. For Use SSO Template, select None.
6. For Username Source, enter the username source to prefill the password collection form. The default `session.sso.token.last.username` works well, because it has the signed-in user Microsoft Entra User Principal Name (UPN).
7. For Password Source, keep the default `session.sso.token.last.password` the APM variable BIG-IP uses to cache user passwords.

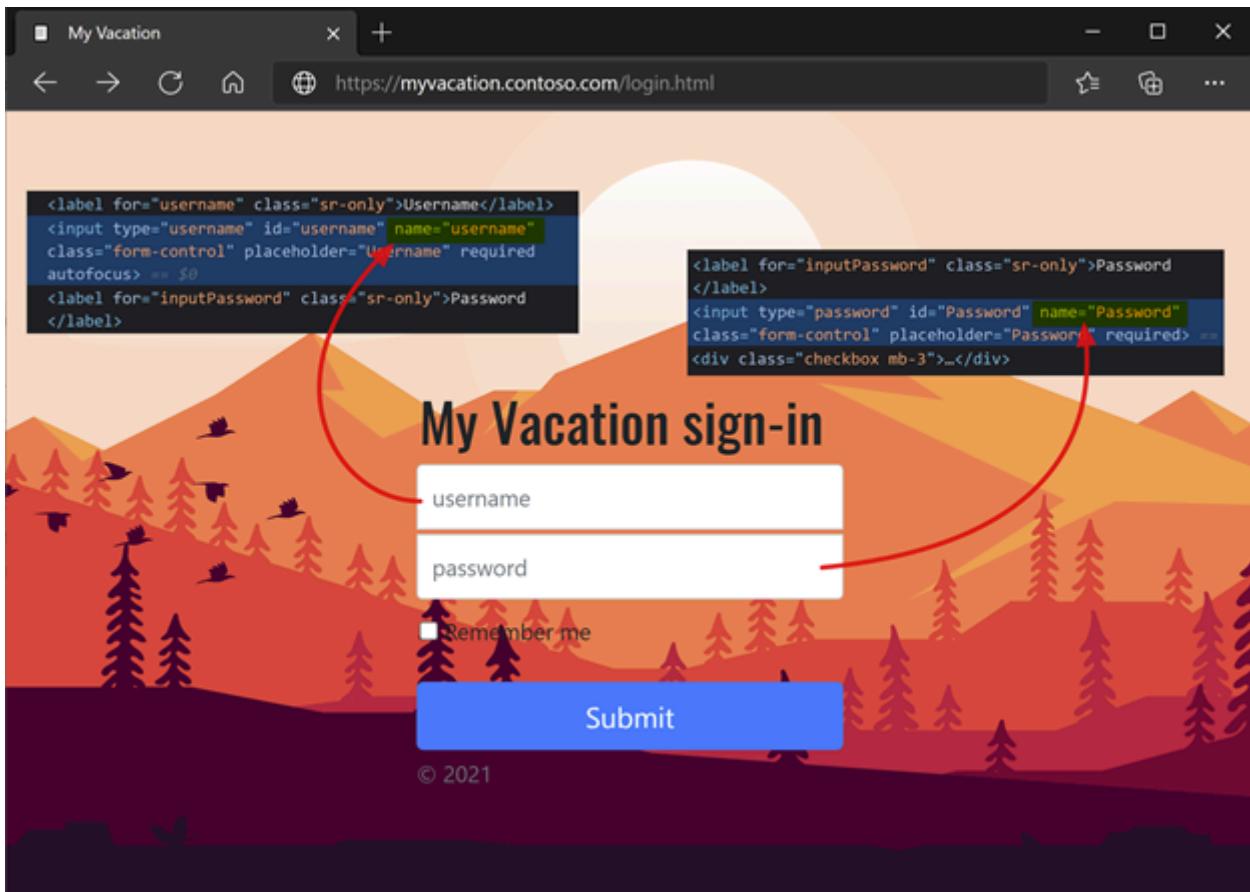
Access » Single Sign-On » New SSO Configuration...

General Properties: Basic	
Name	Contoso_FBA_sso
SSO Method	Forms
Use SSO Template	None
Log Settings	+ From Access Profile
Credentials Source	
Username Source	session.sso.token.last.username
Password Source	session.sso.token.last.password

8. For **Start URI**, enter the FBA application logon URI. If the request URI matches this URI value, the APM form-based authentication executes SSO.
9. For **Form Action**, leave it blank. Then, the original request URL is used for SSO.
10. For **Form Parameter for Username**, enter the sign in form username field element.
Use the browser dev tools to determine the element.
11. For **Form Parameter for Password**, enter the sign in form password field element.
Use the browser dev tools to determine the element.

SSO Method Configuration

Start URI	/login.html
Pass Through	<input type="checkbox"/> Enable
Form Method	POST
Form Action	
Form Parameter For User Name	username
Form Parameter For Password	password
Hidden Form Parameters/Values	
Successful Logon Detection Match Type	None
Successful Logon Detection Match Value	



To learn more, go to [techdocs.f5.com](#) for [Manual Chapter: Single sign-on methods](#).

Configure an access profile

An access profile binds the APM elements that manage access to BIG-IP virtual servers, including access policies, SSO configuration, and UI settings.

1. Select Access > Profiles / Policies.
2. Select Access Profiles (Per-Session Policies).
3. Select Create.
4. Enter a Name.
5. For Profile Type, select All.
6. For SSO Configuration, select the FBA SSO configuration object you created.
7. For Accepted Language, select at least one language.

Access » Profiles / Policies : Access Profiles (Per-Session Policies) » New Profile...

General Properties

Name	MyVacation
Parent Profile	access
Profile Type	All
Profile Scope	Profile
Customization Type	Modern

Settings

Custom

Configurations

SSO Across Authentication Domains (Single Domain mode)

Domain Cookie	
Cookie Options	<input checked="" type="checkbox"/> Secure <input type="checkbox"/> Persistent <input type="checkbox"/> HTTP Only <input type="checkbox"/> Samesite
SSO Configuration	Contoso_FBA_sso

Language Settings

Additional Languages	Afar (aa) <input type="button" value="Add"/>
Languages	Accepted Languages English (en)
Default Language	English (en) <input type="button" value=""/>
<input type="button" value="Cancel"/> <input type="button" value="Finished"/>	

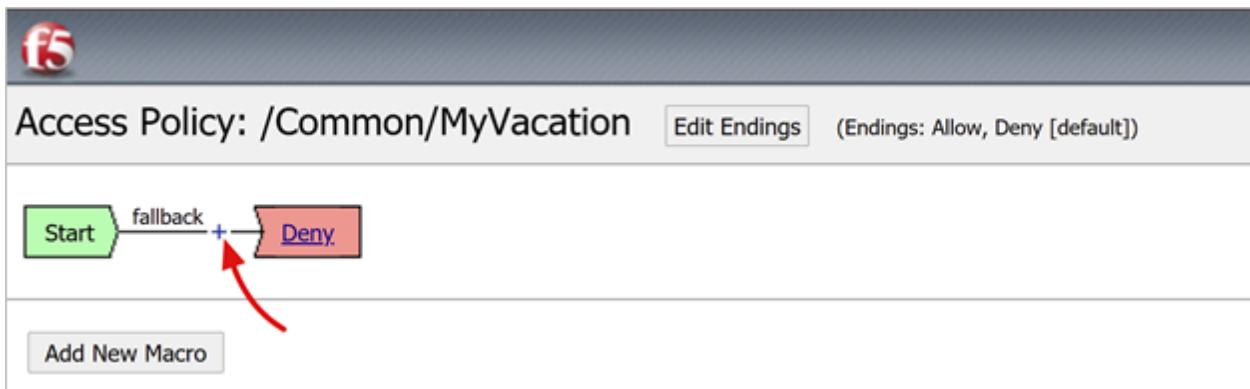
8. In the Per-Session Policy column, for the profile, select **Edit**.

9. The APM Visual Policy Editor starts.

Access » Profiles / Policies : Access Profiles (Per-Session Policies)

Status	Access Profile Name	Application	Profile Type	Per-Session Policy	Export	Copy	Customization
<input type="checkbox"/>	<input type="checkbox"/> MyVacation	All	<input type="checkbox"/> Edit...	<input type="button" value="Export..."/>	<input type="button" value="Copy..."/>	<input type="button" value="Modern"/>	
<input type="checkbox"/>	<input type="checkbox"/> access	All	(none)	(none)	(none)	(none)	

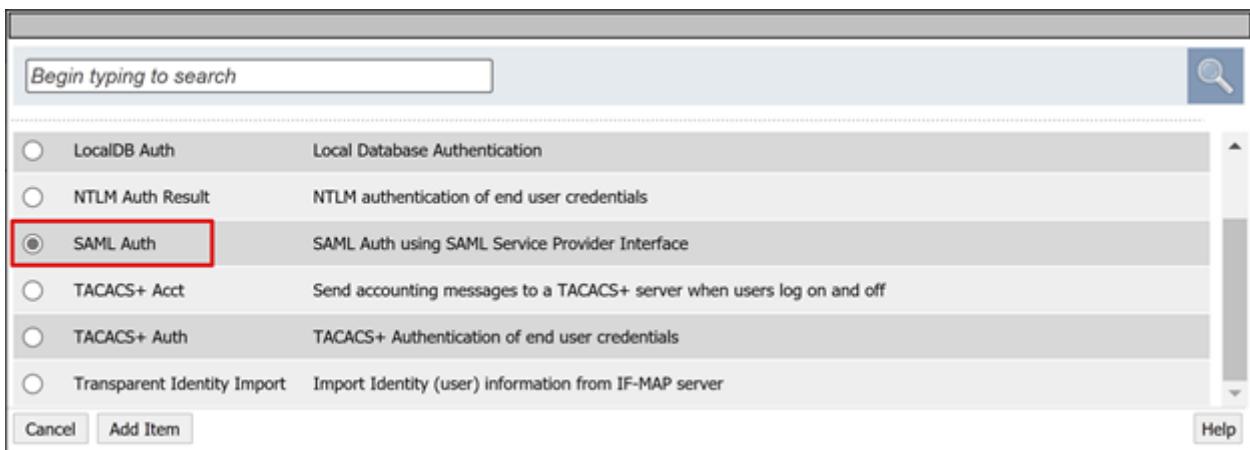
10. Under **fallback**, select the + sign.



11. In the pop-up, select **Authentication**.

12. Select **SAML Auth**.

13. Select **Add Item**.



14. On **SAML authentication SP**, change the **Name** to **Microsoft Entra auth**.

15. In the **AAA Server** dropdown, enter the SAML service provider object you created.

Properties*		Branch Rules
Name:		Microsoft Entra auth
SAML Authentication SP		
AAA Server	/Common/MyVacation ▾	
Attribute Consuming Service	None ▾	
Force Authentication	Use AAA server setting ▾	
Cancel Save (*Data in tab has been changed, please don't forget to save)		Help

16. On the **Successful** branch, select the **+** sign.

17. In the pop-up, select **Authentication**.

18. Select **Logon Page**.

19. Select **Add Item**.

Begin typing to search

Logon Authentication Assignment Endpoint Security (Server-Side) Endpoint Security (Client-Side) General Purpose

- Citrix Logon Prompt Configure logon options for Citrix clients
- External Logon Page Redirect user to externally hosted form-based web logon page
- HTTP 401 Response HTTP 401 Response for Basic or SPNEGO/Kerberos authentication
- HTTP 407 Response HTTP 407 Response for Basic or SPNEGO/Kerberos authentication
- Logon Page** Web form-based logon page for collecting end user credentials (used with most deployments)
- OAuth Logon Page OAuth Logon Page used for OAuth Client authentication
- VMware View Logon Page Display logon screen on VMware View clients

Cancel Add Item Help

20. For **username**, in the **Read Only** column, select **Yes**.

Properties* Branch Rules

Name: Logon Page

Logon Page Agent

Split domain from full Username	No
CAPTCHA Configuration	None

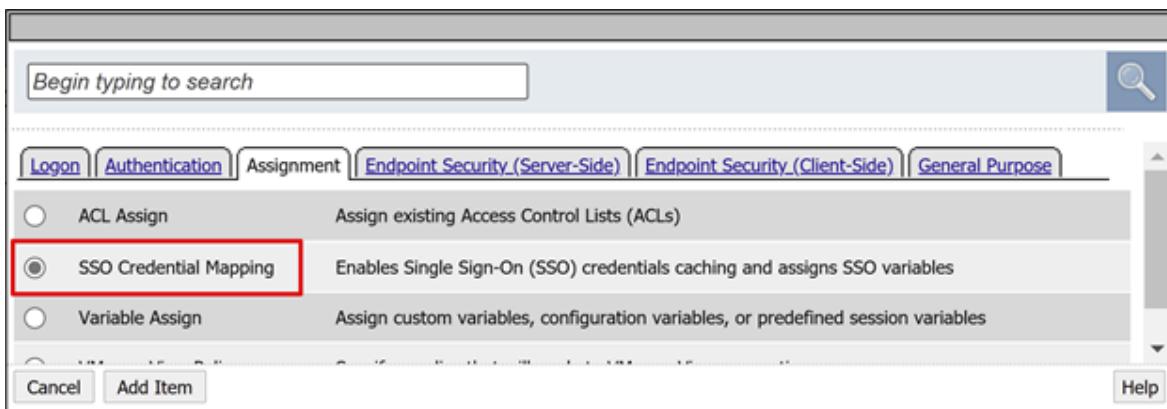
	Type	Post Variable Name	Session Variable Name	Clean Variable	Values	Read Only
1	text	username	username	No		Yes
2	password	password	password	No		No
3	none	field3	field3	No		No
4	none	field4	field4	No		No
5	none	field5	field5	No		No

21. For the sign in page fallback, select the + sign. This action adds an SSO credential mapping object.

22. In the pop-up, select the **Assignment** tab.

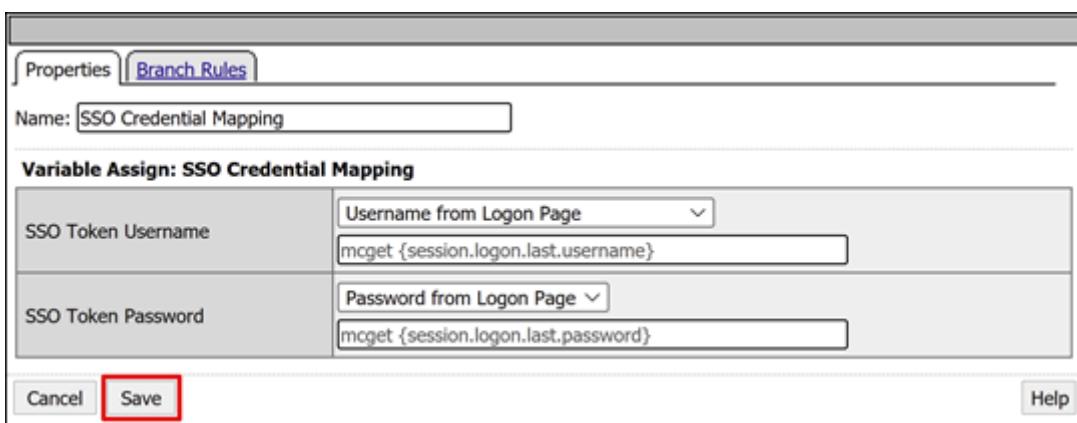
23. Select **SSO Credential Mapping**.

24. Select **Add Item**.



25. On Variable Assign: SSO Credential Mapping, keep the default settings.

26. Select Save.



27. In the upper Deny box, select the link.

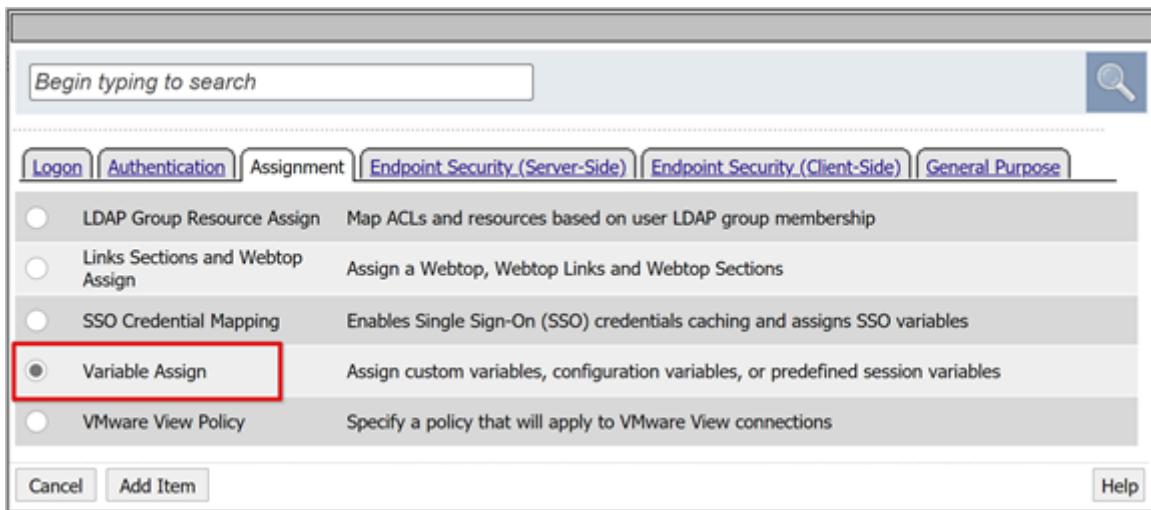
28. The Successful branch changes to Allow.

29. Select Save.

(Optional) Configure attribute mappings

You can add a LogonID_Mapping configuration. Then, the BIG-IP active sessions list has the signed-in user UPN, not a session number. Use this information for analyzing logs or troubleshooting.

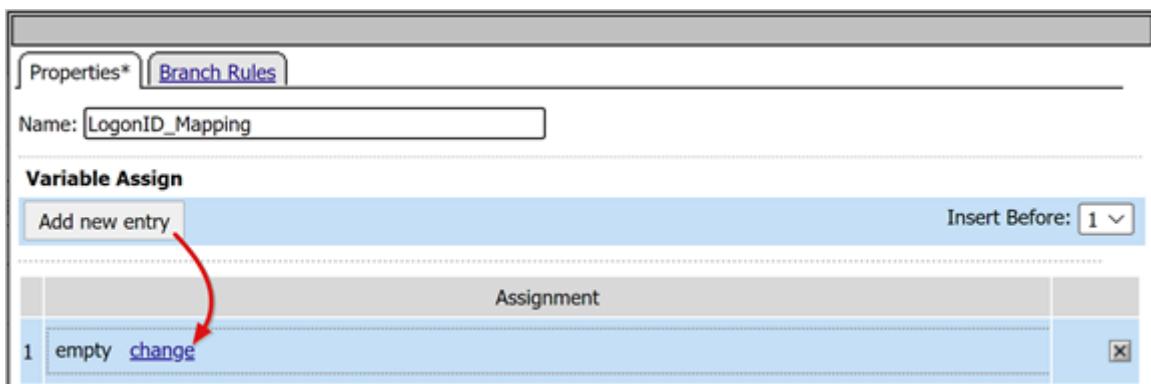
1. For the SAML Auth Successful branch, select the + sign.
2. In the pop-up, select Assignment.
3. Select Variable Assign.
4. Select Add Item.



5. On the Properties tab, enter a Name. For example, LogonID_Mapping.

6. Under Variable Assign, select Add new entry.

7. Select change.



8. For Custom Variable, use `session.logon.last.username`.

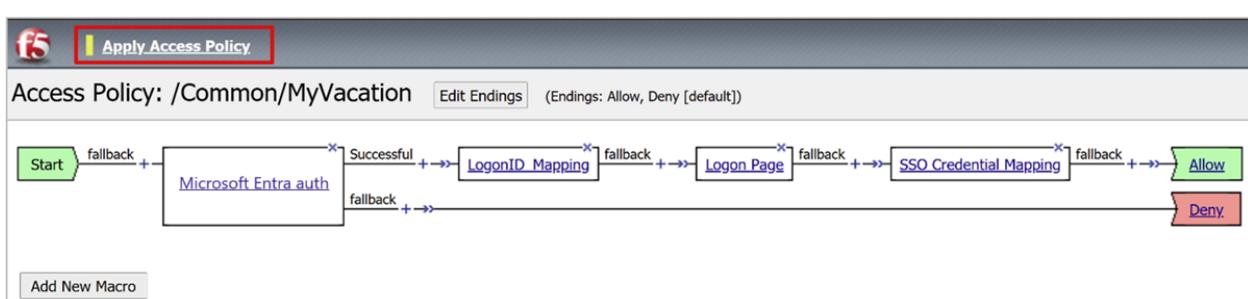
9. For Session Variable, user `session.saml.last.identity`.

10. Select Finished.

11. Select Save.

12. Select Apply Access Policy.

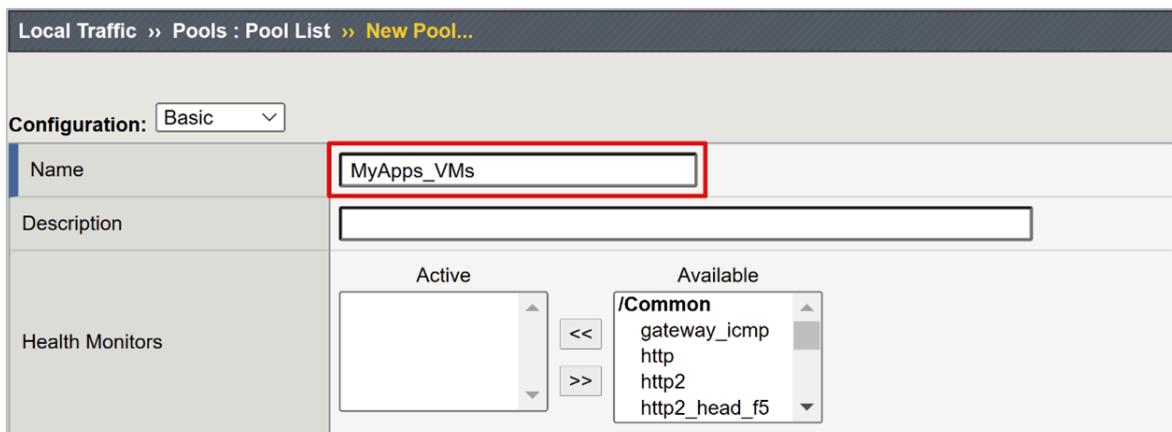
13. Close the Visual Policy Editor.



Configure a back-end pool

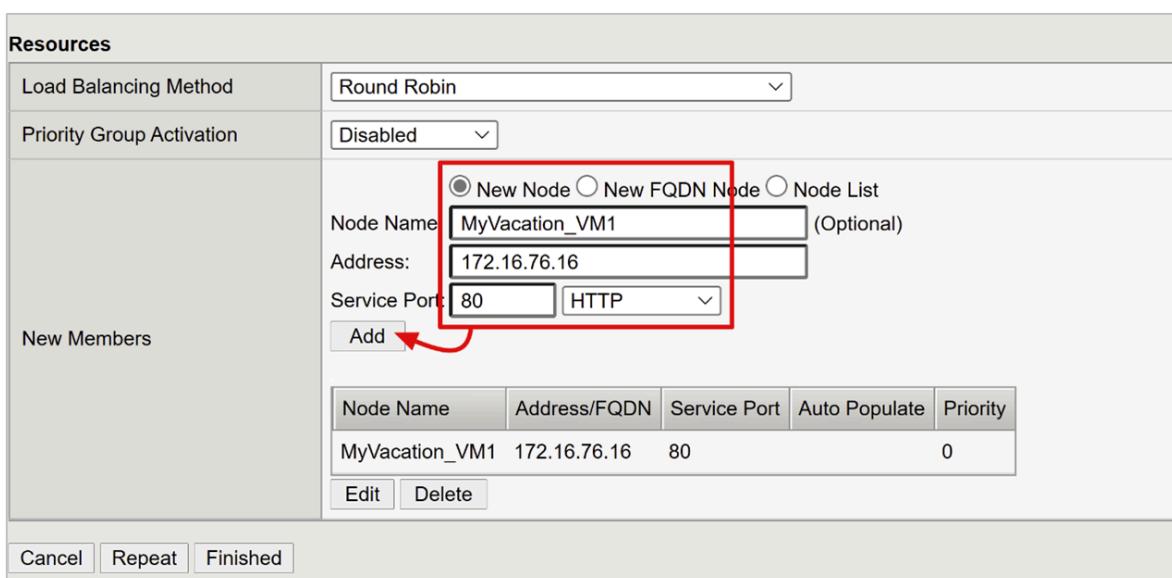
To enable BIG-IP to forward client traffic correctly, create a BIG-IP node object that represents the back-end server that hosts your application. Then, place that node in a BIG-IP server pool.

1. Select Local Traffic > Pools.
2. Select Pool List.
3. Select Create.
4. Enter a Name for a server pool object. For example, MyApp_VMs.



The screenshot shows the 'Local Traffic > Pools : Pool List' screen. A new pool is being created with the name 'MyApp_VMs'. The 'Name' field is highlighted with a red box. Below it, there's a 'Description' field and a section for 'Health Monitors' which includes a list of available monitors: /Common/gateway_icmp, http, http2, and http2_head_f5. There are also 'Active' and 'Available' lists with arrows for moving items between them.

5. For **Node Name**, enter a server display name. This server hosts the back-end web application.
6. For **Address**, enter the application server host IP address.
7. For **Service Port** enter the HTTP/S port the application is listening on.



The screenshot shows the 'Resources' tab of the 'New Members' configuration dialog. It includes fields for 'Load Balancing Method' (set to 'Round Robin'), 'Priority Group Activation' (set to 'Disabled'), and 'New Members'. Under 'New Members', there are fields for 'Node Name' (set to 'MyVacation_VM1'), 'Address' (set to '172.16.76.16'), and 'Service Port' (set to '80 HTTP'). An 'Add' button is highlighted with a red arrow. Below this, a table shows the member details: Node Name: MyVacation_VM1, Address/FQDN: 172.16.76.16, Service Port: 80, Auto Populate: Yes, Priority: 0. There are 'Edit' and 'Delete' buttons at the bottom of the table.

(!) Note

Health monitors require configuration this article doesn't cover. Go to support.f5.com for [K13397: Overview of HTTP health monitor request formatting for the BIG-IP DNS system](#).

Configure a virtual server

A virtual server is a BIG-IP data-plane object represented by a virtual IP address. The server listens for client requests to the application. Any received traffic is processed and evaluated against the APM access profile associated with the virtual server. The traffic is directed according to policy.

To configure a virtual server:

1. Select Local Traffic > Virtual Servers.
2. Select Virtual Server List.
3. Select Create.
4. Enter a Name.
5. For Destination Address/Mask, select Host and enter an IPv4 or IPv6 address. The address receives client traffic for the published back-end application.
6. For Service Port, select Port, enter 443, and select HTTPS.

The screenshot shows the 'Local Traffic > Virtual Servers : Virtual Server List' screen. A yellow link 'New Virtual Server...' is visible. The 'General Properties' table has the following entries:

General Properties	
Name	<input type="text" value="MyVacation"/>
Description	APM Listener for MyVacation web application
Type	Standard
Source Address	<input checked="" type="radio"/> Host <input type="radio"/> Address List <input type="text"/>
Destination Address/Mask	<input checked="" type="radio"/> Host <input type="radio"/> Address List <input type="text" value="172.16.76.28"/>
Service Port	<input checked="" type="radio"/> Port <input type="radio"/> Port List <input type="text" value="443"/> <input type="radio"/> HTTPS
Notify Status to Virtual Address	<input checked="" type="checkbox"/>
State	Enabled

7. For HTTP Profile (Client), select http.

8. For **SSL Profile (Client)**, select the profile you created, or leave the default for testing. This option enables a virtual server for Transport Layer Security (TLS) to publish services over HTTPS.

Configuration: Basic	
DoH Profile Type	None
Protocol	TCP
Protocol Profile (Client)	tcp
Protocol Profile (Server)	(Use Client Profile)
HTTP Profile (Client)	http
HTTP Profile (Server)	(Use Client Profile)
HTTP Proxy Connect Profile	None
FTP Profile	None
RTSP Profile	None
PPTP Profile	None
Selected	
SSL Profile (Client)	/Common Contoso_clientssl
Available	
SSL Profile (Client)	/Common F5Demo F5DemoClient clientssl clientssl-insecure-compatible clientssl-quic clientssl-secure
Selected	
SSL Profile (Server)	/Common
Available	
SSL Profile (Server)	/Common apm-default-serverssl cloud-service-default-ssl

9. For **Source Address Translation**, select **Auto Map**.

Source Address Translation	Auto Map
----------------------------	----------

10. Under **Access Policy**, in the **Access Profile** box, enter the name you created. This action binds the Microsoft Entra SAML preauthentication profile and FBA SSO policy to the virtual server.

Access Policy	
Access Profile	<input type="text" value="MyVacation"/>
Connectivity Profile	<input type="text" value="None"/>
Per-Request Policy	<input type="text" value="None"/>
VDI Profile	<input type="text" value="None"/>
Application Tunnels (Java & Per-App VPN)	<input type="checkbox"/> Enabled
OAM Support	<input type="checkbox"/> Enabled
ADFS Proxy	<input type="checkbox"/> Enabled
PingAccess Profile	<input type="text" value="None"/>

11. Under Resources, for Default Pool, select the back-end pool objects you created.
12. Select Finished.

Resources		
iRules	<div style="display: flex; align-items: center;"> <div style="flex: 1;"> <p>Enabled</p> <div style="border: 1px solid #ccc; padding: 5px; height: 100px;"></div> </div> <div style="margin: 0 10px;"> << >> </div> <div style="flex: 1;"> <p>Available</p> <div style="border: 1px solid #ccc; padding: 5px; height: 100px;"> /Common <code>_sys_APM_ExchangeSupport_OA</code> <code>_sys_APM_ExchangeSupport_OA</code> <code>_sys_APM_ExchangeSupport_he</code> <code>_sys_APM_ExchangeSupport_ma</code> </div> </div> </div>	
Policies	<div style="display: flex; align-items: center;"> <div style="flex: 1;"> <p>Enabled</p> <div style="border: 1px solid #ccc; padding: 5px; height: 100px;"></div> </div> <div style="margin: 0 10px;"> << >> </div> <div style="flex: 1;"> <p>Available</p> <div style="border: 1px solid #ccc; padding: 5px; height: 100px;"></div> </div> </div>	
Default Pool	<div style="display: flex; align-items: center;"> + <input type="text" value="MyApps_VMs"/> </div>	
Default Persistence Profile	<input type="text" value="None"/>	
Fallback Persistence Profile	<input type="text" value="None"/>	

Configure session management settings

BIG-IP session management settings define conditions for sessions termination and continuation. Create policy in this area.

1. Go to Access Policy.
2. Select Access Profiles.
3. Select Access Profile.
4. From the list, select your application.

If you defined a single logout URI value in Microsoft Entra ID, IdP-initiated sign out from MyApps ends the client and the BIG-IP APM session. The imported application federation metadata XML file provides the APM with the Microsoft Entra SAML endpoint for SP-initiated sign out. Ensure the APM responds correctly to a user sign out.

If there's no BIG-IP web portal, users can't instruct the APM to sign out. If the user signs out of the application, BIG-IP is oblivious. The application session can be reinstated through SSO. For SP-initiated sign out, ensure sessions terminate securely.

You can add an SLO function to your application **sign out** button. This function redirects the client to the Microsoft Entra SAML sign out endpoint. To locate SAML sign out endpoint, go to **App Registrations > Endpoints**.

If you can't change the app, have the BIG-IP listen for the app sign out call and trigger SLO.

Learn more:

- [K42052145: Configuring automatic session termination \(logout\) based on a URI-referenced file name ↗](#)
- [K12056: Overview of the Logout URI Include option ↗](#)

Published application

Your application is published and accessible with SHA with the app URL or Microsoft portals.

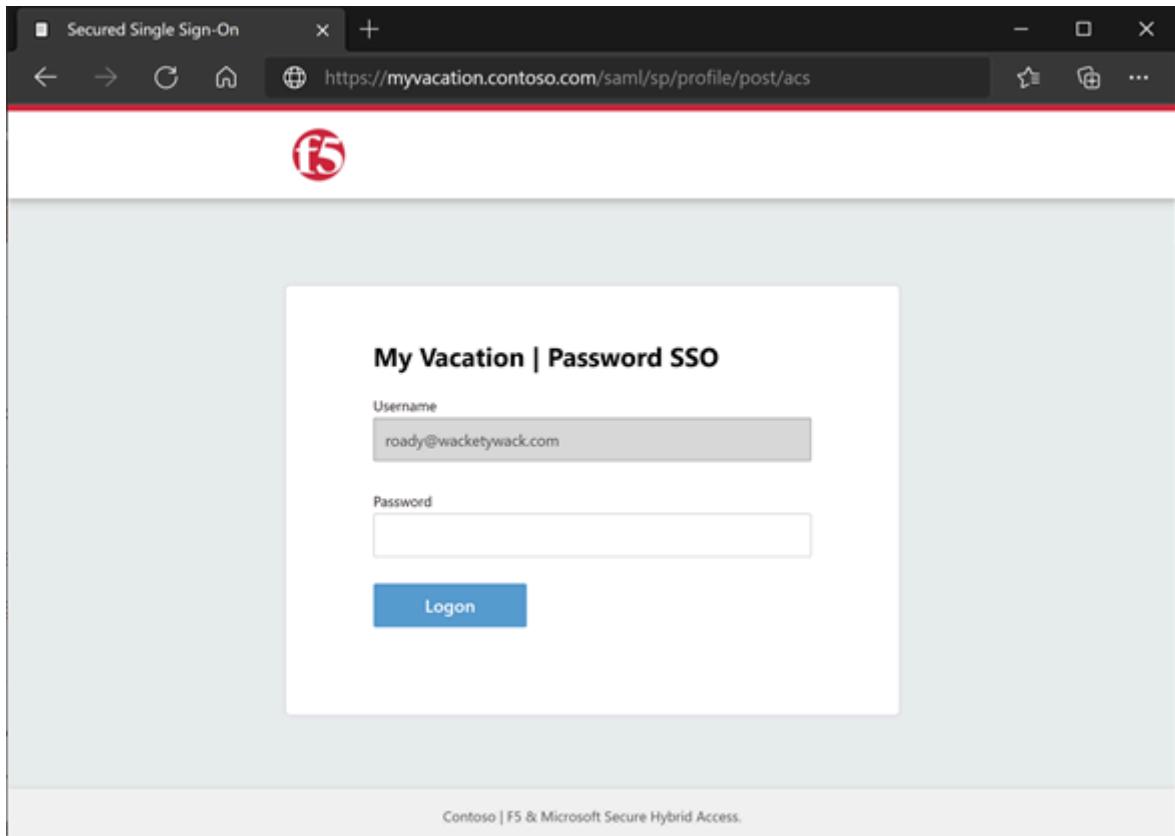
The application appears as a target resource in Conditional Access. Learn more: [Building a Conditional Access policy](#).

For increased security, block direct access to the application, enforcing a path through the BIG-IP.

Test

1. The user connects to the application external URL, or in My Apps, and selects the application icon.
2. The user authenticates to Microsoft Entra ID.
3. The user is redirected to the BIG-IP endpoint for the application.
4. The password prompt appears.

5. The APM fills the username with the UPN from Microsoft Entra ID. The username is read-only for session consistency. Hide this field, if needed.



6. The information is submitted.

7. The user is signed in to the application.

The screenshot shows a web browser window with the title "My Vacation". The address bar displays the URL "https://myvacation.contoso.com/secured/default.html". The page header includes a navigation bar with links: "MyTravel", "Before & during your trip", "Sustainable travel", "Planning resources", and "MyTravel News".

The main content area features a yellow banner with the text "Welcome Roady". Below it are two main sections:

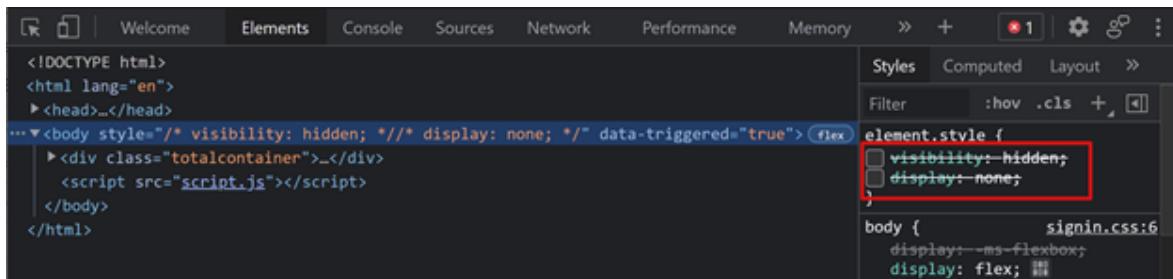
- Your online booking tool**: Includes a circular icon with a plane symbol and the text "Concur Travel tool". It notes: "Note: You will leave the MyTravel site and enter a third party site for travel services." It lists "Upcoming trainings": "Concur booking tool training" on Sep 8, 2021, and "MyTravel Program Intro and Q&A" on Oct 13, 2021.
 - Concur booking tool training**: Sep 8, 2021 10:00 AM – 11:00 AM (UTC±00) Greenwich Mean Time [\[link\]](#)
 - MyTravel Program Intro and Q&A**: Oct 13, 2021 10:00 AM – 11:00 AM (UTC±00) Greenwich Mean Time [\[link\]](#)
- Your travel agency contact**: Includes a circular icon with a phone symbol and the text "Country of work: United Kingdom". It provides contact details: "Phone: +44 203788373", "Email: travelbureau@contoso.com", "Hours: Monday-Friday Offline 08:00-18:00; Online Navigation Support 24 hours - 7 days a week", "Traveler Profile: Your travel profile is stored within the [Online Booking tool](#). Please access the tool to fully complete your profile before you need to book your first trip.", and "Visa Support: Visa support services are available <http://contoso.visas.com> account number: 102456 or contact your country destination embassy directly."

In the bottom left corner, there is a blue circular button with a white "X" and the text "Ask MyTravel".

Troubleshoot

When troubleshooting, consider the following information:

- BIG-IP performs FBA SSO as it parses the sign in form at the URI
 - BIG-IP seeks the username and password element tags from your configuration
- Confirm element tags are consistent, or SSO fails
- Complex forms generated dynamically might require dev tool analysis to understand the sign in form
- Client initiation is better for sign in pages with multiple forms
 - You can select the form name and customize the JavaScript form handler logic
- FBA SSO methods hide form interactions to optimize user experience and security:
 - You can validate if the credentials are injected
 - In client-initiated mode, disable form auto submission in your SSO profile
 - Use dev tools to disable the two style properties that prevent the sign in page from appearing



Increase log verbosity

BIG-IP logs contain information to isolating authentication and SSO issues. Increase the log verbosity level:

1. Go to Access Policy > Overview.
2. Select Event Logs.
3. Select Settings.
4. Select the row of your published application.
5. Select Edit.
6. Select Access System Logs.
7. In the SSO list, select Debug.
8. Select OK.
9. Reproduce the issue.
10. Review the logs.

Revert the settings otherwise there's excessive data.

BIG-IP error message

If a BIG-IP error appears after Microsoft Entra preauthentication, the issue might relate to Microsoft Entra ID and BIG-IP SSO.

1. Go to **Access > Overview**.
2. Select **Access reports**.
3. Run the report for the last hour.
4. Review the logs for clues.

Use the **View session variables** link for your session to determine if the APM receives expected Microsoft Entra claims.

No BIG-IP error message

If no BIG-IP error message appears, the issue might relate to the back-end request, or BIG-IP-to-application SSO.

1. Select **Access Policy > Overview**.
2. Select **Active Sessions**.
3. Select the active session link.

Use the **View Variables** link in this location to help determine root cause, particularly if the APM fails to obtain correct user identifier and password.

To learn more, go to [techdocs.f5.com](#) for [Manual Chapter: Session Variables](#).

Resources

- Go to [techdocs.f5.com](#) for [Manual Chapter: Authentication](#)
- [Passwordless authentication](#)
- [What is Conditional Access?](#)
- [Zero Trust framework to enable remote work](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Tutorial: Configure F5 BIG-IP SSL-VPN for Microsoft Entra SSO

Article • 04/19/2024

In this tutorial, learn how to integrate F5 BIG-IP based secure socket layer virtual private network (SSL-VPN) with Microsoft Entra ID for secure hybrid access (SHA).

Enabling a BIG-IP SSL-VPN for Microsoft Entra single sign-on (SSO) provides many benefits, including:

- Zero Trust governance through Microsoft Entra preauthentication and Conditional Access.
 - [Conditional Access](#)
- [Passwordless authentication ↗](#) to the VPN service
- Identity and access management from a single control plane, the [Microsoft Entra admin center ↗](#)

To learn about more benefits, see

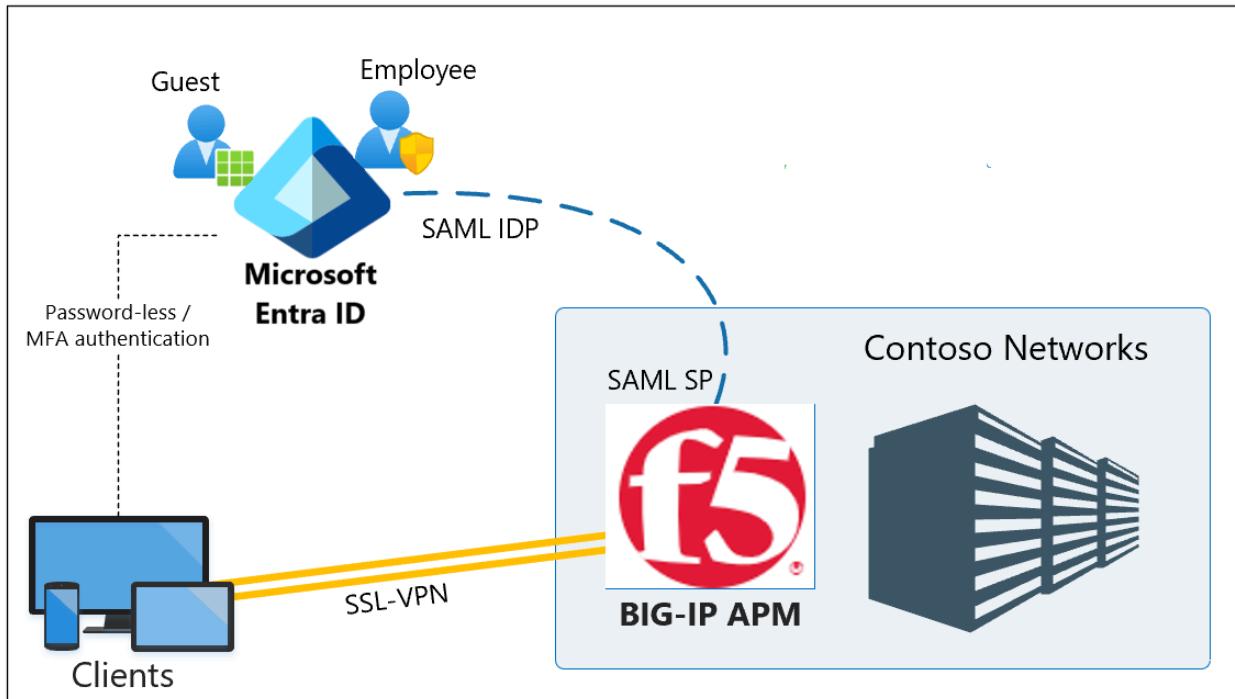
- [F5 BIG-IP integration with Microsoft Entra ID](#)
- [SSO in Microsoft Entra ID](#)

ⓘ Note

Classic VPNs remain network orientated, often providing little to no fine-grained access to corporate applications. We encourage a more identity-centric approach to achieve Zero Trust. Learn more: [Five steps for integrating all your apps with Microsoft Entra ID](#).

Scenario description

In this scenario, the BIG-IP Access Policy Manager (APM) instance of the SSL-VPN service is configured as a Security Assertion Markup Language (SAML) service provider (SP) and Microsoft Entra ID is the trusted SAML identity provider (IdP). Single sign-on (SSO) from Microsoft Entra ID is through claims-based authentication to the BIG-IP APM, a seamless virtual private network (VPN) access experience.



! Note

Replace example strings or values in this guide with those in your environment.

Prerequisites

Prior experience or knowledge of F5 BIG-IP isn't necessary, however, you need:

- A Microsoft Entra subscription
 - If you don't have one, you can get an [Azure free account](#)
- User identities [synchronized from their on-premises directory](#) to Microsoft Entra ID
- One of the following roles: Global Administrator, Cloud Application Administrator, or Application Administrator
- BIG-IP infrastructure with client traffic routing to and from the BIG-IP
 - Or [deploy a BIG-IP Virtual Edition into Azure](#)
- A record for the BIG-IP published VPN service in a public domain name server (DNS)
 - Or a test client localhost file while testing
- The BIG-IP provisioned with the needed SSL certificates for publishing services over HTTPS

To improve the tutorial experience, you can learn industry-standard terminology on the F5 BIG-IP [Glossary](#).

Add F5 BIG-IP from the Microsoft Entra gallery

💡 Tip

Steps in this article might vary slightly based on the portal you start from.

Set up a SAML federation trust between the BIG-IP to allow the Microsoft Entra BIG-IP to hand off the preauthentication and [Conditional Access](#) to Microsoft Entra ID, before it grants access to the published VPN service.

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**, then select **New application**.
3. In the gallery, search for *F5* and select **F5 BIG-IP APM Microsoft Entra ID integration**.
4. Enter a name for the application.
5. Select **Add** then **Create**.
6. The name, as an icon, appears in the Microsoft Entra admin center and Office 365 portal.

Configure Microsoft Entra SSO

1. With F5 application properties, go to **Manage > Single sign-on**.
2. On the **Select a single sign-on method** page, select **SAML**.
3. Select **No, I'll save later**.
4. On the **Setup single sign-on with SAML** menu, select the pen icon for **Basic SAML Configuration**.
5. Replace the **Identifier URL** with your BIG-IP published service URL. For example, `https://ssl-vpn.contoso.com`.
6. Replace the **Reply URL**, and the SAML endpoint path. For example, `https://ssl-vpn.contoso.com/saml/sp/profile/post.acs`.

❗ Note

In this configuration, the application operates in an IdP-initiated mode: Microsoft Entra ID issues a SAML assertion before redirecting to the BIG-IP SAML service.

7. For apps that don't support IdP-initiated mode, for the BIG-IP SAML service, specify the **Sign-on URL**, for example, `https://ssl-vpn.contoso.com`.
8. For the Logout URL, enter the BIG-IP APM Single logout (SLO) endpoint prepended by the host header of the service being published. For example, `https://ssl-vpn.contoso.com/saml/sp/profile/redirect/slo`

① Note

An SLO URL ensures a user session terminates at BIG-IP and Microsoft Entra ID, after the user signs out. BIG-IP APM has an option to terminate all sessions when calling an application URL. Learn more on the F5 article, [K12056: Overview of the Logout URI Include option](#).

1

Basic SAML Configuration

 Edit

Identifier (Entity ID)
`https://ssl-vpn.contoso.com`

Reply URL (Assertion Consumer Service URL)
`https://ssl-vpn.contoso.com/saml/sp/profile/post/acs`
Optional

Sign on URL
Optional
`https://ssl-vpn.contoso.com/saml/sp/profile/redirect/slo`

② Note

From TMOS v16, the SAML SLO endpoint has changed to /saml/sp/profile/redirect/slo.

9. Select **Save**

10. Skip the SSO test prompt.

11. In **User Attributes & Claims** properties, observe the details.

2

User Attributes & Claims

 Edit

givenname
surname
emailaddress
name
identity
Unique User Identifier

user.givenname
user.surname
user.mail
user.userprincipalname
user.onpremisessamaccountname
user.userprincipalname

You can add other claims to your BIG-IP published service. Claims defined in addition to the default set are issued if they're in Microsoft Entra ID. Define directory [roles or group](#) memberships against a user object in Microsoft Entra ID, before they can be issued as a claim.

3 SAML Signing Certificate

Status	Active
Thumbprint	9285F77DC7A9582B1C4FFC690EA8660844E02A56
Expiration	19/03/2023, 2:12:02 PM
Notification Email	rodney@contoso.com
App Federation Metadata Url	https://login.microsoftonline.com/536279f6-15cc-41d0-8a22-ebc5a1a3a3a3/.well-known/openid-configuration
Certificate (Base64)	Download
Certificate (Raw)	Download
Federation Metadata XML	Download

SAML signing certificates created by Microsoft Entra ID have a lifespan of three years.

Microsoft Entra authorization

By default, Microsoft Entra ID issues tokens to users with granted access to a service.

1. In the application configuration view, select **Users and groups**.
2. Select **+ Add user**.
3. In the **Add Assignment** menu, select **Users and groups**.
4. In the **Users and groups** dialog, add the user groups authorized to access the VPN
5. Select **Select > Assign**.

Add user		Edit	Remove	Update Credentials	Columns	...	
<p>The application will appear on the Access Panel for assigned users. Set 'visible to users?' to → no in properties to prevent this.</p>							
First 100 shown, to search all users & groups, enter a display name.							
Display Name	Object Type	Role assigned					
<input type="checkbox"/> AW	All Contoso users	Group	Default Access				

You can set up BIG-IP APM to publish the SSL-VPN service. Configure it with corresponding properties to complete the trust for SAML preauthentication.

BIG-IP APM configuration

SAML federation

To complete federating the VPN service with Microsoft Entra ID, create the BIG-IP SAML service provider and corresponding SAML IDP objects.

1. Go to Access > Federation > SAML Service Provider > Local SP Services.

2. Select Create.



3. Enter a Name and the Entity ID defined in Microsoft Entra ID.

4. Enter the Host fully qualified domain name (FQDN) to connect to the application.

A screenshot of the 'Create New SAML SP Service' dialog box. On the left is a sidebar with icons for General Settings, Endpoint Settings, Security Settings, Authentication Cont..., Requested Attributes, and Advanced Settings. The main area has two sections: 'Name*' (containing 'VPN') and 'Entity ID*' (containing 'https://ssl-vpn.contoso.com'). Both of these sections are enclosed in a red rectangular box. Below them is a section titled 'SP Name Settings' with fields for 'Scheme' (set to 'https') and 'Host'. Further down are fields for 'Description' and 'Relay State'. At the bottom right are 'OK' and 'Cancel' buttons.

Note

If the entity ID isn't an exact match of the hostname of the published URL, configure SP Name settings, or perform this action if it isn't in hostname URL format. If entity ID is `urn:ssl-vpn:contosoonline`, provide the external scheme and hostname of the application being published.

5. Scroll down to select the new SAML SP object.

6. Select Bind/UnBind IDP Connectors.

The screenshot shows the 'Access' interface with the 'Federation : SAML Service Provider : Local SP Services' path selected. The 'SAML Service Provider' tab is active. A table displays a single row for 'VPN', which is described as 'SP Object for BIG-IP VPN' and belongs to the 'Common' partition. Below the table are buttons for 'Edit', 'Delete', and 'Bind/Unbind IdP Connectors', with the latter being highlighted by a red box.

7. Select Create New IDP Connector.

8. From the drop-down menu, select From Metadata

The screenshot shows the 'Edit SAML IdPs that use this SP' dialog. It lists a single row for 'SAML IdP Connectors' under 'Matching Source'. To the right, a dropdown menu is open, showing three options: 'Custom', 'From Metadata' (which is highlighted with a red arrow), and 'From Template'. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

9. Browse to the federation metadata XML file you downloaded.

10. For the APM object, provide an **Identity Provider Name** that represents the external SAML IdP.

11. To select the new Microsoft Entra external IdP connector, select **Add New Row**.

Edit SAML IdPs that use this SP

IdP Connectors associated with this SP Service		
<input type="button" value="Add New Row"/> <input type="button" value="Create New IdP Connector"/>		
SAML IdP Connectors	Matching Source	Matching Value
<input checked="" type="checkbox"/> SAML IdP Connectors	/Common/VPN	
<input type="button" value="Update"/> <input type="button" value="Cancel"/>		
<input type="button" value="Edit"/> <input type="button" value="Delete"/>		
<input type="button" value="OK"/> <input type="button" value="Cancel"/>		

12. Select **Update**.

13. Select **OK**.

Edit SAML IdPs that use this SP

IdP Connectors associated with this SP Service		
<input type="button" value="Add New Row"/> <input type="button" value="Create New IdP Connector"/>		
SAML IdP Connectors	Matching Source	Matching Value
<input type="checkbox"/>	<u>/Common/VPN</u>	
<input type="button" value="Edit"/> <input type="button" value="Delete"/>		
<input type="button" value="OK"/> <input type="button" value="Cancel"/>		

Webtop configuration

Enable the SSL-VPN to be offered to users via the BIG-IP web portal.

1. Go to Access > Webtops > Webtop Lists.

2. Select **Create**.

3. Enter a portal name.

4. Set the type to **Full**, for example, `Contoso_webtop`.

5. Complete the remaining preferences.

6. Select **Finished**.

The screenshot shows a configuration dialog for a new webtop. At the top, the path is shown as 'Access » Webtops : Webtop Lists » New Webtop...'. The 'General Properties' section contains three fields: 'Name' (Contoso_Webtop), 'Type' (Network Access, which is highlighted with a red box), and 'Customization Type' (Modern). Below this is a 'Configuration' section with a single field 'Minimize To Tray' (Enabled, checked). At the bottom are three buttons: 'Cancel', 'Repeat', and 'Finished'.

VPN configuration

VPN elements control aspects of the overall service.

1. Go to **Access > Connectivity/VPN > Network Access (VPN) > IPV4 Lease Pools**
2. Select **Create**.
3. Enter a name for the IP address pool allocated to VPN clients. For example, **Contoso_vpn_pool**.
4. Set type to **IP Address Range**.
5. Enter a start and end IP.
6. Select **Add**.
7. Select **Finished**.

General Properties

Name

Contoso_VPN_Pool

ConfigurationType: IP Address IP Address Range

Start IP Address 172.16.76.200

End IP Address 172.16.76.220

Add

172.16.76.200 - 172.16.76.220

Member List

Edit **Delete****Cancel** **Repeat** **Finished**

A Network access list provisions the service with IP and DNS settings from the VPN pool, user routing permissions, and can launch applications.

1. Go to Access > Connectivity/VPN: Network Access (VPN) > Network Access Lists.
2. Select **Create**.
3. Provide a name for the VPN access list and caption, for example, Contoso-VPN.
4. Select **Finished**.

General Properties

Name

Contoso-VPN

Description

Auto launch

 Enable**Customization Settings for English**

Language

English

Caption

Contoso-VPN

Detailed Description

Choose File No file chosen **View/Hide** **Restore Default**

Image

**Cancel** **Finished**

5. From the top ribbon, select **Network Settings**.

6. For **Supported IP version**: IPV4.

7. For **IPV4 Lease Pool**, select the VPN pool created, for example, Contoso_vpn_pool

The screenshot shows the 'Network Settings' tab selected in the top navigation bar. Under 'General Settings', the 'Supported IP Version' is set to 'IPV4'. The 'IPV4 Lease Pool' dropdown is set to 'Contoso_VPN_Pool', which is highlighted with a red box. In the 'Client Settings' section, the 'Force all traffic through tunnel' radio button is selected. Other options like 'Use split tunneling for traffic', 'Allow Local Subnet', and 'Client Options' are also visible.

! Note

Use the Client Settings options to enforce restrictions for how client traffic is routed in an established VPN.

8. Select **Finished**.

9. Go to the **DNS/Hosts** tab.

10. For **IPV4 Primary Name Server**: Your environment DNS IP

11. For **DNS Default Domain Suffix**: The domain suffix for this VPN connection. For example, contoso.com

Access » Connectivity / VPN : Network Access (VPN) : Network Access Lists » Contoso-VPN

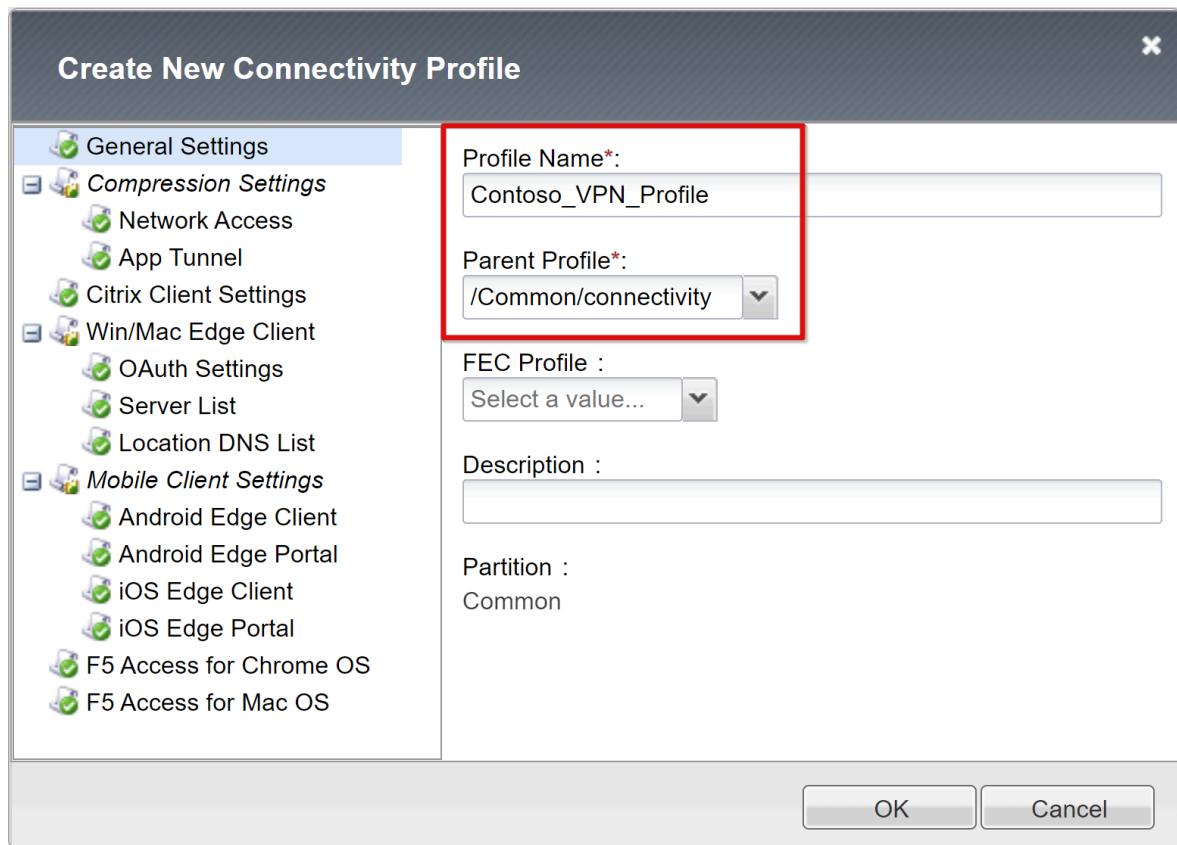
Properties	Network Settings	Optimization	DNS/Hosts	Drive Mappings	Launch Applications
DNS Configuration					
IPV4 Primary Name Server	<input type="text" value="172.16.76.14"/>				
IPV4 Secondary Name Server	<input type="text"/>				
Primary WINS Server	<input type="text"/>				
Secondary WINS Server	<input type="text"/>				
DNS Default Domain Suffix	<input type="text" value="contoso.com"/>				
Register this connection's addresses in DNS	<input type="checkbox"/> Enable				
Use this connection's DNS suffix in DNS registration	<input type="checkbox"/> Enable				
Enforce DNS search order	<input checked="" type="checkbox"/> Enable				
	Host Name	<input type="text"/>			
	IP Address	<input type="text"/>			
	Add	<input type="button" value="Add"/>			
Static Hosts	<input type="button" value="Edit"/> <input type="button" value="Delete"/>				

(!) Note

See the F5 article, [Configuring Network Access Resources](#) for other settings.

A BIG-IP connection profile is required to configure VPN client-type settings the VPN service needs to support. For example, Windows, OSX, and Android.

1. Go to Access > Connectivity/VPN > Connectivity > Profiles
2. Select Add.
3. Enter a profile name.
4. Set the parent profile to **/Common/connectivity**, for example, Contoso_VPN_Profile.



For more information on client support, see the F5 article, [F5 Access and BIG-IP Edge Client](#).

Access profile configuration

An access policy enables the service for SAML authentication.

1. Go to Access > Profiles/Policies > Access Profiles (Per-Session Policies).
2. Select **Create**.
3. Enter a profile name and for the profile type.
4. Select **All**, for example, Contoso_network_access.
5. Scroll down and add at least one language to the Accepted Languages list
6. Select **Finished**.

Access » Profiles / Policies : Access Profiles (Per-Session Policies) » New Profile...

General Properties

Name	Contoso_network_access
Parent Profile	access
Profile Type	All
Profile Scope	Profile
Customization Type	Modern

Settings Custom

Configurations

SSO Across Authentication Domains (Single Domain mode)

Language Settings

Additional Languages	Afar (aa) <input type="button" value="Add"/>
Languages	<p>Accepted Languages</p> <div style="border: 1px solid red; padding: 2px;"> <input type="button" value="English (en)"/> </div> <p>Factory BuiltIn Languages</p> <ul style="list-style-type: none"> Japanese (ja) Chinese (Simplified) (zh-cn) Chinese (Traditional) (zh-tw) Korean (ko) Spanish (es) French (fr) German (de)
Default Language	English (en) <input type="button" value=""/>

7. In the new access profile, on the Per-Session Policy field, select **Edit**.

8. The visual policy editor opens in a new tab.

Access » Profiles / Policies : Access Profiles (Per-Session Policies)									
		Status		Access Profile Name		Application		Per-Session Policy	
<input checked="" type="checkbox"/>		<input type="checkbox"/>		Contoso_network_access		All		<input type="button" value="Edit..."/>	
<input type="checkbox"/>		access		All	(none)	(none)	(none)	default-log-setting	VPN_Listener
<input type="button" value="Delete..."/>	<input type="button" value="Apply"/>							Common	Common

9. Select the + sign.

10. In the menu, select **Authentication > SAML Auth**.

11. Select **Add Item**.

12. In the SAML authentication SP configuration, select the VPN SAML SP object you created

13. Select **Save**.

Properties* Branch Rules

Name: SAML Auth

SAML Authentication SP

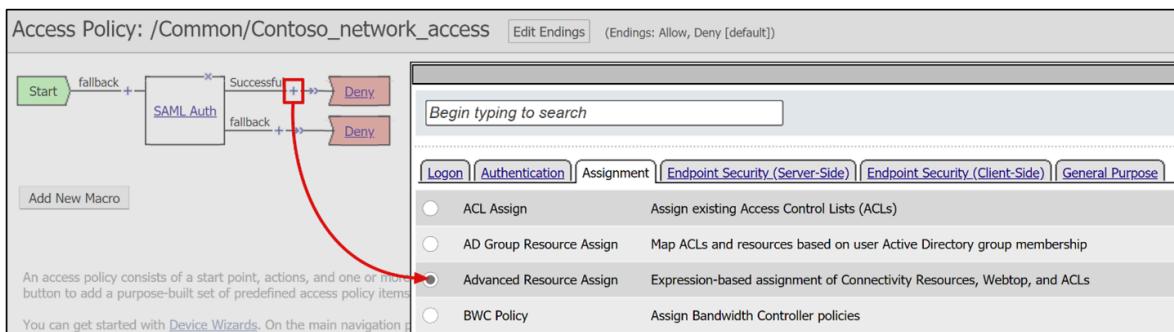
AAA Server	/Common/VPN
Attribute Consuming Service	None
Force Authentication	Use AAA server setting

Cancel Save (*Data in tab has been changed, please don't forget to save) Help

14. For the Successful branch of SAML auth, select + .

15. From the Assignment tab, select **Advanced Resource Assign.**

16. Select **Add Item**.



17. In the pop-up, select **New Entry**

18. Select **Add/Delete**.

19. In the window, select **Network Access**.

20. Select the Network Access profile you created.

Properties* Branch Rules

Name: Advanced Resource Assign

Resource Assignment

Add new entry

Expression: Empty change

1 Add/Delete

Begin typing to search in Current Tab

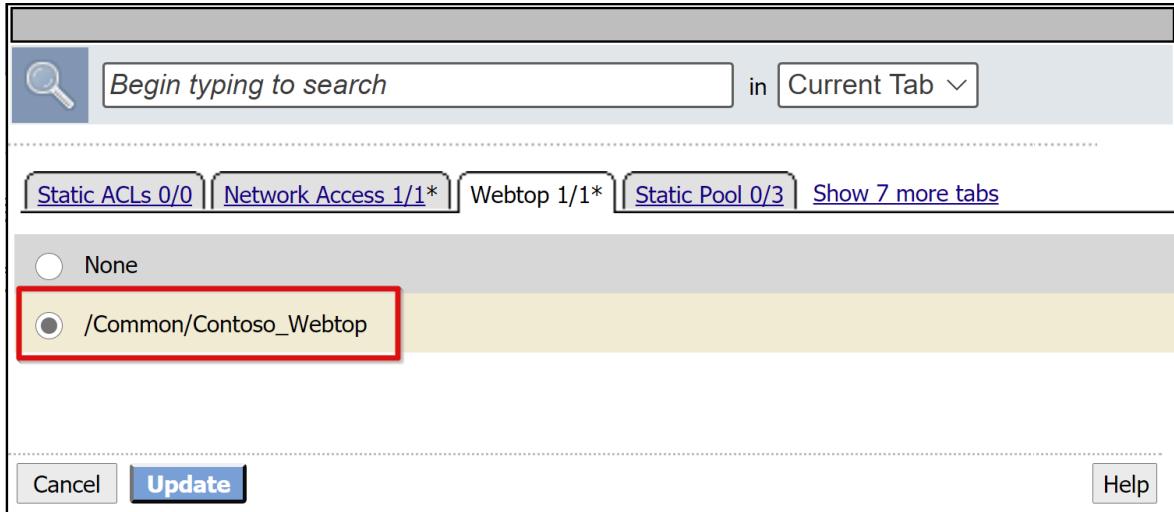
Static ACLs 0/0 Network Access 1/1* Webtop 0/1 Static Pool 0/3 Show 7 more tabs

/Common/Contoso-VPN

Cancel Save (*Data in tab has been changed, please don't forget to save) Update Help

21. Go to the **Webtop** tab.

22. Add the Webtop object you created.



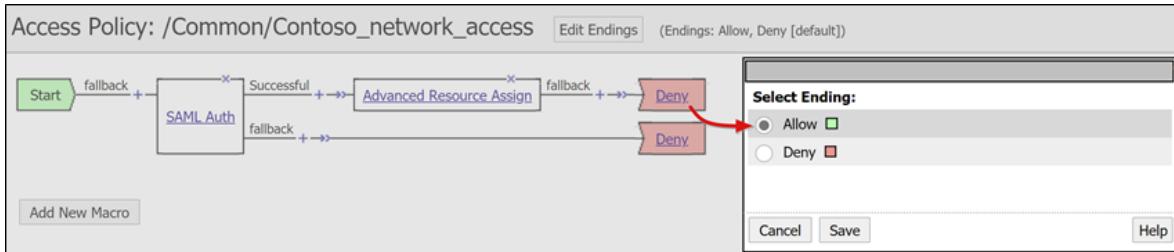
23. Select Update.

24. Select Save.

25. To change the Successful branch, select the link in the upper Deny box.

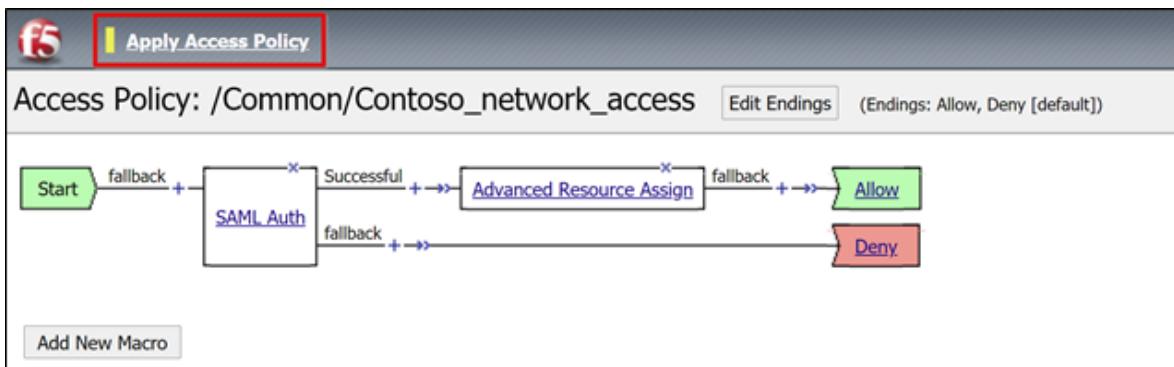
26. The Allow label appears.

27. Save.



28. Select Apply Access Policy

29. Close the visual policy editor tab.



Publish the VPN service

The APM requires a front-end virtual server to listen for clients connecting to the VPN.

1. Select Local Traffic > Virtual Servers > Virtual Server List.
2. Select Create.
3. For the VPN virtual server, enter a **Name**, for example, VPN_Listener.
4. Select an unused **IP Destination Address** with routing to receive client traffic.
5. Set the Service Port to **443 HTTPS**.
6. For **State**, ensure **Enabled** is selected.

Local Traffic » Virtual Servers : Virtual Server List » New Virtual Server...

General Properties	
Name	<input type="text" value="VPN_Listener"/>
Description	<input type="text"/>
Type	Standard
Source Address	<input checked="" type="radio"/> Host <input type="radio"/> Address List <input type="text"/>
Destination Address/Mask	<input checked="" type="radio"/> Host <input type="radio"/> Address List <input type="text" value="172.16.76.29"/>
Service Port	<input checked="" type="radio"/> Port <input type="radio"/> Port List 443 <input type="text" value="HTTPS"/>
Notify Status to Virtual Address	<input checked="" type="checkbox"/>
State	Enabled

7. Set the **HTTP Profile** to **http**.
8. Add the **SSL Profile (Client)** for the public SSL certificate you created.

Configuration: Basic

Protocol	TCP
Protocol Profile (Client)	tcp
Protocol Profile (Server)	(Use Client Profile)
HTTP Profile (Client)	http
HTTP Profile (Server)	(Use Client Profile)
HTTP Proxy Connect Profile	None
FTP Profile	None
RTSP Profile	None
	Selected <div style="border: 1px solid #ccc; padding: 5px; display: inline-block;"> /Common Contoso_clientssl </div> <div style="margin-top: 10px;"> << >> </div>
SSL Profile (Client)	Available <div style="border: 1px solid #ccc; padding: 5px; display: inline-block;"> /Common clientssl clientssl-insecure-compatible clientssl-quic clientssl-secure crypto-server-default-clientssl splitsession-default-clientssl </div>

9. To use the created VPN objects, under Access Policy, set the **Access Profile** and **Connectivity Profile**.

Access Policy

Access Profile	Contoso_network_access
Connectivity Profile	+ Contoso_VPN_Profile
Per-Request Policy	None
VDI Profile	None
Application Tunnels (Java & Per-App VPN)	<input type="checkbox"/> Enabled
OAM Support	<input type="checkbox"/> Enabled
ADFS Proxy	<input type="checkbox"/> Enabled
PingAccess Profile	None

Ephemeral Authentication

API Protection

Acceleration: Basic

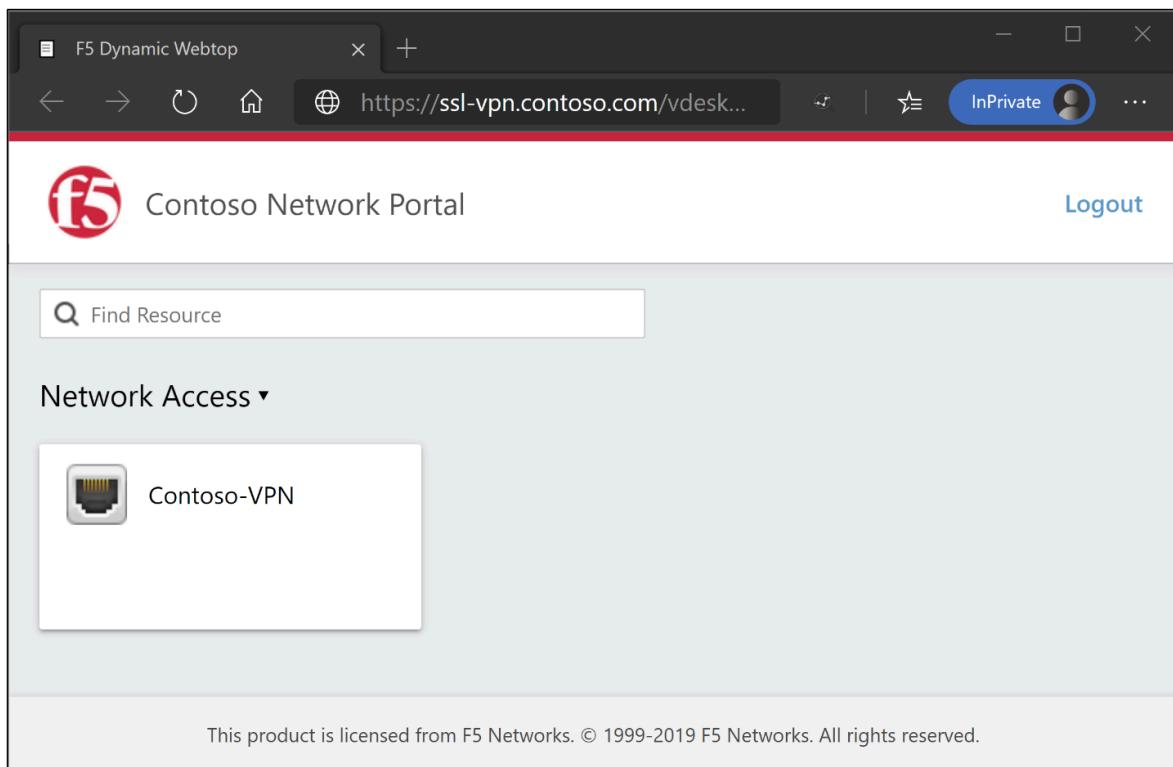
Buttons: Update | Delete

10. Select **Finished**.

Your SSL-VPN service is published and accessible via SHA, either with its URL or through Microsoft application portals.

Next steps

1. Open a browser on a remote Windows client.
2. Browse to the **BIG-IP VPN service URL**.
3. The BIG-IP webtop portal and VPN launcher appear.



ⓘ Note

Select the VPN tile to install the BIG-IP Edge client and establish a VPN connection configured for SHA. The F5 VPN application is visible as a target resource in Microsoft Entra Conditional Access. See [Conditional Access policies](#) to enable users for Microsoft Entra ID [password-less authentication](#) ↗ .

Resources

- [The end of passwords, go passwordless ↗](#)
- [Five steps to full application integration with Microsoft Entra ID](#)
- [Microsoft Zero Trust framework to enable remote work ↗](#)

Tutorial: Enable secure hybrid access for applications with Azure Active Directory B2C and F5 BIG-IP

Article • 06/21/2024

Learn to integrate Azure Active Directory B2C (Azure AD B2C) with F5 BIG-IP Access Policy Manager (APM). You can expose legacy applications securely to the internet through BIG-IP security, with Azure AD B2C preauthentication, Conditional Access (CA), and single sign-on (SSO). F5 Inc. focuses on delivery, security, performance, and availability of connected services, including computing, storage, and network resources. It provides hardware, modularized software, and cloud-ready virtual appliance solutions.

Deploy F5 BIG-IP Application Delivery Controller (ADC) as a secure gateway between private networks and the internet. There are features for application-level inspection and customizable access controls. If deployed as a reverse proxy, use the BIG-IP to enable secure hybrid access to business applications, with a federated identity access layer managed by APM.

Go to f5.com resources and white papers for: [Easily Configure Secure Access to All Your Applications via Microsoft Entra ID ↗](#)

Prerequisites

To get started, you need:

- An Azure subscription
 - If you don't have one, get an [Azure free account ↗](#)
- An Azure AD B2C tenant linked to the Azure subscription
 - See, [Tutorial: Create an Azure Active Directory B2C tenant](#)
- A BIG-IP or a deployed trial BIG-IP Virtual Environment (VE) on Azure
 - See, [Deploy F5 BIG-IP Virtual Edition VM in Azure](#)
- Any of the following F5 BIG-IP licenses:
 - F5 BIG-IP® Best bundle
 - F5 BIG-IP Access Policy Manager™ standalone license
 - F5 BIG-IP Access Policy Manager™ add-on license on a BIG-IP F5 BIG-IP® Local Traffic Manager™ (LTM)
 - 90-day BIG-IP full feature [trial license ↗](#)
- A header-based web application or an IIS app for testing
 - See, [Set up an IIS app](#)

- SSL certificate to publish services over HTTPS, or use default while testing
 - See, [SSL profile](#)

Scenario description

The following scenario is header-based, but you can use these methods to achieve Kerberos SSO.

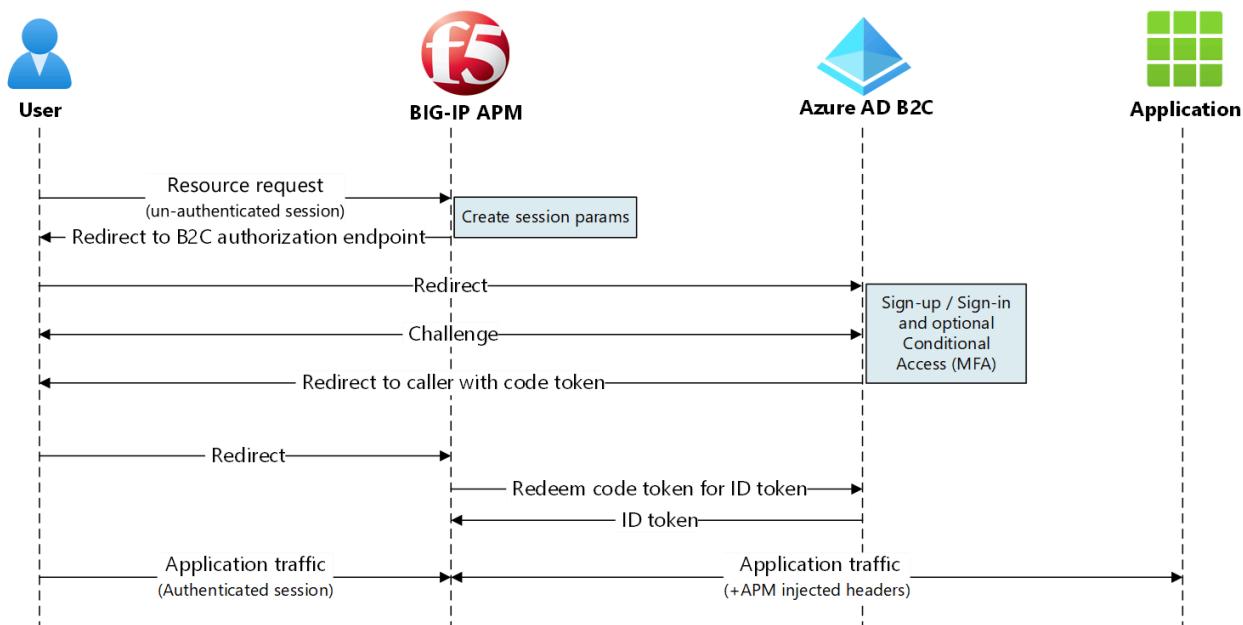
For this scenario, access for an internal application relies on receiving HTTP authorization headers from a legacy broker system. Sales agents can be directed to respective areas of content. The service needs to be expanded to a broader consumer base. The application gets upgraded for consumer authentication options, or gets replaced.

Ideally, an application upgrade supports direct management and governance with a modern control plane. However, time and effort to modernize introduces costs and potential downtime. Instead, deploy a BIG-IP Virtual Edition (VE) between the public internet and the internal Azure virtual network (VNet) to gate access with Azure AD B2C. BIG-IP in front of the application enables overlay of the service with Azure AD B2C preauthentication and header-based SSO, improving the app security posture.

The secure hybrid access solution has of the following components:

- **Application** - back-end service protected by Azure AD B2C and BIG-IP secure hybrid access
- **Azure AD B2C** - identity provider (IdP) and OpenID Connect (OIDC) authorization server that verifies user credentials, multifactor authentication, and SSO to the BIG-IP APM
- **BIG-IP** - reverse proxy for the application. The BIG-IP APM is the OIDC client, delegating authentication to the OIDC authorization server, before header-based SSO to the back-end service.

The following diagram illustrates the service provider (SP) initiated flow for this scenario.



1. User connects to the application endpoint. BIG-IP is service provider.
2. BIG-IP APM OIDC client redirects user to Azure AD B2C tenant endpoint, the OIDC authorization server
3. Azure AD B2C tenant pre-authenticates user and applies Conditional Access policies
4. Azure AD B2C redirects user back to the SP with authorization code
5. OIDC client asks the authorization server to exchange authorization code for an ID token
6. BIG-IP APM grants user access and injects the HTTP headers in the client request forwarded on to the application

Azure AD B2C configuration

To enable a BIG-IP with Azure AD B2C authentication, use an Azure AD B2C tenant with a user flow or custom policy.

See, [Tutorial: Create user flows and custom policies in Azure AD B2C](#)

Create custom attributes

Obtain custom attributes from Azure AD B2C user objects, federated IdPs, API connectors, or user sign-up. Include attributes in the token that goes to the application.

Legacy applications expect specific attributes, so include them in your user flow. You can replace them with attributes your application requires. Or if you're setting up a test app using the instructions, then user any headers.

1. Sign in to the [Azure portal](#) as at least B2C IEF Policy Administrator.

2. In the left-hand pane, select **User attributes**.
3. Select **Add** to create two custom attributes.
4. For Agent ID, select String **Data Type**.
5. For Agent Geo, select String **Data Type**.

Add attributes to user flow

1. In the left-hand menu, navigate go to **Policies > User flows**.
2. Select your policy, for example, **B2C_1_SignupSignin**.
3. Select **User attributes**.
4. Add both custom attributes.
5. Add the **Display Name** attribute. These attributes are collected during user sign-up.
6. Select **Application claims**.
7. Add both custom attributes.
8. Add the **Display Name**. These attributes go to the BIG-IP.
9. Select **Run user flow**.
10. In the user flow menu, on the left navigation bar, verify the prompts for defined attributes.

Learn more: [Tutorial: Create user flows and custom policies in Azure AD B2C](#)

Azure AD B2C federation

Federate BIG-IP and Azure AD B2C for mutual trust. Register the BIG-IP in the Azure AD B2C tenant as an OIDC application.

1. In the portal, select **App registrations > New registration**.
2. Enter an app **Name**, for example, **HeaderApp1**.
3. Under **Supported account types**, select **Accounts in any identity provider or organizational directory (for authenticating users with user flows)**.
4. Under **Redirect URI**, select **Web**.
5. Enter protected service public FQDN.
6. Enter the path.
7. Leave the remaining selections.
8. Select **Register**.
9. Navigate to **Certificates & secrets > + New client secret**.
10. Enter a descriptive name
11. Enter a TTL for the secret used by the BIG-IP.
12. Note the Client Secret for BIG-IP configuration.

The redirect URI is the BIG-IP endpoint. After authentication, the authorization server (Azure AD B2C) sends users to the endpoint.

Learn more: [Tutorial: Register a web application in Azure AD B2C](#) for Azure AD B2C.

BIG-IP configuration

For BIG-IP configuration use Guided Configuration v.7/8. The workflow framework is tailored to access topologies and it accomplishes rapid web service publishing.

Guided Configuration version

1. To confirm version, sign in to the BIG-IP web config with an administrator account.
2. Go to **Access > Guided Configuration**.
3. The version appears in the top right-hand corner.

To upgrade the Guided Configuration, go to [my.f5.com for K85454683: Upgrade F5 BIG-IP Guided Configuration on the BIG-IP system ↗](#).

SSL profiles

Use BIG-IP configured with a client SSL profile to secure client-side traffic over TLS. Import a certificate that matches the domain name, used by the public-facing URL for your app. We recommend you use a public certificate authority, but you can use BIG-IP self-signed certificates for testing.

To add and manage certificates in the BIG-IP VE, go to [techdocs.f5.com for BIG-IP System: SSL Administration ↗](#).

Guided Configuration

1. To launch the deployment wizard, in the web config, go to **Access > Guided Configuration**.
2. Select **Federation > F5 as OAuth Client and Resource Server**.
3. Observe the flow summary for this scenario.
4. Select **Next**.
5. The wizard starts.

OAuth properties

In the following sections, define properties to enable federation between the BIG-IP APM and the OAuth authorization server, the Azure AD B2C tenant. OAuth is referred to throughout BIG-IP configuration. The solution uses OIDC, an identity layer on the OAuth 2.0 protocol. OIDC clients verify user identity and obtain other profile information.

Configuration name

A configuration display name helps distinguish between deployment configurations in the Guided Configuration. You can't change the name, and it appears only in the Guided Configuration view.

Mode

The BIG-IP APM is an OIDC client, therefore select the Client option.

DNS resolver

The specified target must resolve the public IP addresses of the Azure AD B2C endpoints. Select a public DNS resolver, or create a new one.

Provider settings

Configure Azure AD B2C as the OAuth2 IdP. The Guided Configuration has Azure AD B2C templates, but not certain scopes.

Add a new provider and configure it:

OAuth general properties

 Expand table

Properties	Description
OAuth provider type	Custom
Choose OAuth provider	Create new, or use an OAuth provider
Name	A display name for the B2C IdP. This name appears to users as a provider option at sign-in
Token type	JSON web token

OAuth policy settings

[Expand table](#)

Properties	Description
Scope	Leave blank. The OpenID scope for user sign-in is added automatically
Grant type	Authorization code
Enable OpenID Connect	Select the option to put the APM OAuth client in OIDC mode
Flow type	Authorization code

OAuth provider settings

The following OpenID URI refers to the metadata endpoint used by OIDC clients to discover IdP information such as signing certificate rollover.

1. Locate the metadata endpoint for your Azure AD B2C tenant. Navigating to [App registrations > Endpoints](#).
2. Copy the Azure AD B2C OpenID Connect metadata document URI. For example,

```
https://wacketywackb2c  
.b2clogin.com/<tenantname>.onmicrosoft.com/<policyname>/v2.0/.well-  
known/openid-configuration.
```

3. Update the URI with your properties,

```
https://<tenantname>.b2clogin.com/WacketywackB2C.onmicrosoft.com/B2C_1_SignUpI  
n/v2.0/.well-known/openid-configuration.
```

4. Paste the URI into the browser.
5. View the OIDC metadata for your Azure AD B2C tenant.

[Expand table](#)

Property	Description
Audience	The application client ID representing the BIG-IP in the Azure AD B2C tenant
Authentication URI	The authorization endpoint in your B2C OIDC metadata
Token URI	The token endpoint in your Azure AD B2C metadata
Userinfo request URI	Leave empty. Azure AD B2C doesn't support this feature
OpenID URI	The OpenID URI metadata endpoint you created
Ignore expired certificate validation	Leave unchecked

Property	Description
Allow self-signed JWK config certificate	Check
Trusted CA bundle	Select ca-bundle.crt to use the default F5 trusted authorities
Discovery interval	Provide an interval for the BIG-IP to query your Azure AD B2C tenant for updates. The minimum interval in AGC version 16.1 0.0.19, is 5 minutes.

OAuth server settings

For the OIDC authorization server, being your Azure AD B2C tenant.

[\[+\] Expand table](#)

Property	Descriptions
Client ID	The application Client ID representing the BIG-IP in the Azure AD B2C tenant
Client Secret	The application Client Secret
Client-server SSL profile	Set an SSL profile to ensure APM communicates with the Azure AD B2C IdP over TLS. Select the default serverssl.

OAuth request settings

The BIG-IP has required Azure AD B2C requests in its preconfigured request set. However, the requests were malformed, and missing important parameters. So, we created them manually.

Token request: Enabled

[\[+\] Expand table](#)

Property	Description
Choose OAuth request	Create new
HTTP method	POST
Enable headers	Unchecked
Enable parameters	Checked

[\[+\] Expand table](#)

Parameter	Parameter name	Parameter value
client_id	client_id	N/A
nonce	nonce	N/A
redirect_uri	redirect_uri	N/A
scope	scope	N/A
response_type	response_type	N/A
client_secret	client_secret	N/A
custom	grant_type	authorization_code

Auth redirect request: Enabled

[\[+\] Expand table](#)

Property	Description
Choose OAuth request	Create new
HTTP method	GET
Prompt type	None
Enable headers	Unchecked
Enable parameters	Checked

[\[+\] Expand table](#)

Parameter	Parameter name	Parameter value
client_id	client_id	N/A
redirect_uri	redirect_uri	N/A
response_type	response_type	N/A
scope	scope	N/A
nonce	nonce	N/A

Token refresh request: Disabled You can enable and configure as needed.

OpenID UserInfo request: Disabled Not supported in global Azure AD B2C tenants.

Virtual server properties

Create a BIG-IP virtual server to intercept external client requests for the back-end service protected by secure hybrid access. Assign the virtual server an IP mapped to the public DNS record for the BIG-IP service endpoint representing the application. Use a virtual server if available, otherwise provide the following properties.

[\[+\] Expand table](#)

Property	Description
Destination address	Private or public IP that becomes the BIG-IP service endpoint for the back-end application
Service port	HTTPS
Enable redirect port	Select so users are auto redirected from http to https
Redirect port	HTTP
Client SSL profile	Swap the predefined <code>clientssl</code> profile with the one that has your SSL certificate. You can test with the default profile, but it likely causes a browser alert.

Pool properties

Back-end services appear in the BIG-IP as a pool, with one or more application servers to which virtual servers direct inbound traffic. Select a pool, otherwise create a new one.

[\[+\] Expand table](#)

Property	Description
Load-balancing method	Select Round Robin
Pool server	Internal IP of the back-end application
Port	Service port of the back-end application

ⓘ Note

Ensure the BIG-IP has line of sight to the pool server address.

SSO settings

A BIG-IP supports SSO options, but in OAuth client mode the Guided Configuration is limited to Kerberos or HTTP Headers. Enable SSO and use the following information for the APM to map defined inbound attributes to outbound headers.

[Expand table](#)

Property	Description
Header Operation	Insert
Header Name	name
Header Value	<code>%{session.oauth.client.last.id_token.name}</code>
Header Operation	Insert
Header Name	agentid
Header Value	<code>%{session.oauth.client.last.id_token.extension_AgentGeo}</code>

Note

APM session variables in curly brackets are case-sensitive. Entering agentid, when the Azure AD B2C attribute name is sent as AgentID, causes an attribute mapping failure. Define attributes in lowercase. In Azure AD B2C, the user flow prompts the user for more attributes, using the attribute name in the portal. Therefore, use sentence case instead of lowercase.

Single Sign-On Settings

Enable Single Sign-On (Optional)

Selected Single Sign-On Type 

HTTP header-based 

SSO Headers 

Header Operation	Header Name	Header Value	Delimiter	Action
insert	name	<code>session.oauth.client.last.id_token.name}</code>		 
insert	agentid	<code>.client.last.id_token.extension_AgentID]</code>		 
insert	agentgeo	<code>lient.last.id_token.extension_AgentGeo</code>		 

Customization properties

Customize the language and appearance of screens users see in the APM access policy flow. Edit screen messages and prompts, change screen layouts, colors, images, and localize captions, descriptions, and messages.

In the **Form Header** text field, replace the `F5 Networks` string with a name that you want.

Session management properties

Use the BIG-IP session management settings to define conditions that terminate sessions or allow them to continue. Set limits for users and IP addresses, and error pages. We recommend implementing single log out (SLO), which terminates sessions securely, reducing risks of unauthorized access.

Deploy settings

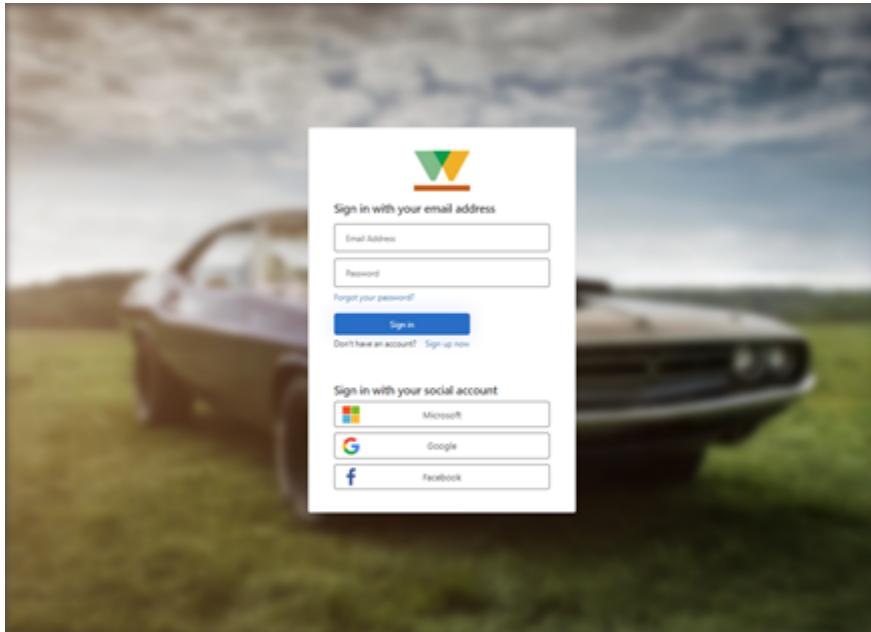
Select **Deploy** to commit settings and create BIG-IP and APM objects for secure hybrid access to the application. The application appears as a target resource in Conditional Access. For increased security, block direct access to the application, thereby enforcing a path through the BIG-IP.

Learn more: [Identity Protection and Conditional Access for Azure AD B2C](#)

Test the sign-in sign-up flow

1. As a user, go to the application external URL.
2. The BIG-IP's OAuth client sign-in page appears.
3. Sign in using the authorization code grant. To remove this step, see the **Supplemental configurations** section.
4. Sign up and authenticate against your Azure AD B2C tenant.

The following images are the user sign in dialog and the sign-in welcome page.



AGENT: TD219 Geneva

Welcome to The Drive Owner Portal

Get tips and tricks on how to make the most of your vehicle, schedule a service pickup, get diagnostic information, read FAQs, or manage all options personalized to your vehicle

Body Wraps Warm Engine Power Mod Crazy Cup Rally

COLOURS START VEHICLE REMAP ECU REGISTER

ABOUT US

The Drive is the world's only independent luxury car group with more than 80 years of design and automotive excellence across multiple brands. We strive to continue being the most renowned car company for creating the most beautiful and accomplished automotive art on the planet.

LOCATIONS

Monaco
London
Fremont

MEDIA

Gallery
News
Blog

CONNECT

FAQs
Help
Contact Us

© 2021 Copyright Wacketywack Inc.

For increased security, block direct access to the application, thereby enforcing a path through the BIG-IP.

Supplemental configurations

Single log-out (SLO)

Azure AD B2C supports identity provider (IdP) and application sign-out. See, [Single sign out](#).

To achieve SLO, enable your application sign out function to call the Azure AD B2C sign-out endpoint. Then, Azure AD B2C issues a final redirect to the BIG-IP. This action ensures the user-application APM session terminates.

An alternative SLO process is to enable the BIG-IP to listen for the request, when selecting the applications **Sign out** button. Upon detecting the request, it calls to the Azure AD B2C sign out endpoint. This approach precludes making changes to the application.

To learn more BIG-IP iRules, go to support.f5.com for [K42052145: Configuring automatic session termination \(logout\) based on a URI-referenced file name](#).

Note

Regardless of approach, ensure the Azure AD B2C tenant knows the APM sign-out endpoint.

1. In the portal, navigate to **Manage > Manifest**.
2. Locate the `logoutUrl` property. It reads null.
3. Add the APM post log-out URI: `https://<mysite.com>/my.logout.php3`

Note

`<mysite.com>` is the BIG-IP FQDN for your header-based application.

Optimized login flow

To improve the user sign-in experience, suppress the OAuth user sign-in prompt that appears before Microsoft Entra preauthentication.

1. Navigate to **Access > Guided Configuration**.
2. On the far right of the row, select the **padlock** icon.
3. The header-based application unlocks the strict configuration.

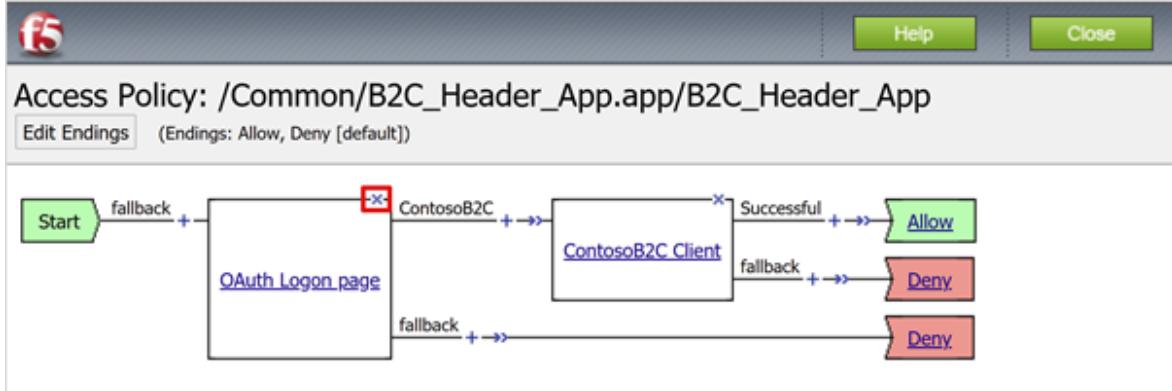
Import	Filter Configurations by Name...	
Status	Name	Type
DEPLOYED	B2C_Header_App	F5 as OAuth Client and Resource Server

Unlocking the strict configuration prevents changes with the wizard UI. BIG-IP objects are associated with the published instance of the application, and are open for direct management.

4. Navigate to Access > Profiles/ Policies > Access Profiles (Per-session Policies).
5. For the application policy object, in the Per-Session Policy column, select Edit.

Access » Profiles / Policies : Access Profiles (Per-Session Policies)						
Access Profiles	Per-Request Policies	Policy Sync	Customization			
<input type="checkbox"/> <input type="button" value="Search"/>						
Status	▲ Access Profile Name	Application	Profile Type	Per-Session Policy	Export	
<input type="checkbox"/>	B2C_Header_App	B2C_HeaderApp	All	<input type="button" value="Edit..."/>	Export...	
<input type="checkbox"/>	access		All	(none)	(none)	

6. To delete the OAuth Logon Page policy object, select X.
7. At the prompt, connect to the previous node.



8. In the top left corner, select Apply Access Policy.
9. Close the visual editor tab.

When you attempt to connect to the application, the Azure AD B2C sign-in page appears.

ⓘ Note

If you re-enable strict mode and deploy a configuration, settings performed outside the Guided Configuration UI are overwritten. Implement this scenario by manually creating configuration objects for production services.

Troubleshooting

Use the following troubleshooting guidance if access to the protected application is prevented.

Log verbosity

BIG-IP logs have information to isolate authentication and SSO issues. Increase the log verbosity level.

1. Go to **Access Policy > Overview > Event Logs > Settings**.
2. Select the row for your published application then **Edit > Access System Logs**.
3. From the SSO list, select **Debug**.
4. Select **OK**.
5. Before reviewing logs, reproduce your issue.

When complete, revert the previous settings.

BIG-IP error message

If you see a BIG-IP error message after Azure AD B2C authentication, the issue might relate to SSO from Microsoft Entra ID to the BIG-IP.

1. Navigate to **Access > Overview > Access reports**.
2. Run the report for the last hour
3. Review logs for clues.
4. Select the **View session variables** link.
5. Determine if the APM receives the expected Microsoft Entra claims.

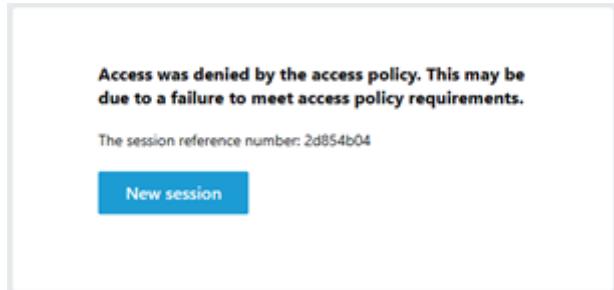
No BIG-IP error message

If no BIG-IP error message appears, the issue might be related to the back-end request, or SSO from the BIG-IP to the application.

1. Go to **Access Policy > Overview > Active Sessions**.
2. Select the link for your active session.
3. Select the **View Variables** link.
4. Review to determine root cause, particularly if the BIG-IP APM obtains inaccurate session attributes.
5. Use the application logs to help understand if it received the attributes as headers.

Guided Configuration v8 known issue

If using Guided Configuration v8, a known issue generates the following error after successful Azure AD B2C authentication. The issue might be the AGC not enabling the Auto JWT setting during deployment. The APM can't obtain the current token signing keys. F5 engineering is investigating root cause.



The same access log provides detail.

A screenshot of a log details interface. The top bar is labeled "Details".

Local Time:
2021-07-29 11:18:49

Log Message:
/Common/B2C_header_App.app/B2C_header_App.Common.2d854b04.
Session variable 'session.oauth.client.lastErrMsg' set to 'None of the
configured JWK keys match the received JWT token, JWT Header:
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiJ3b2pCRNvtMWtsI
Partitions:
Common

Below the log message, there is a navigation bar with buttons for First, Previous, Next, and Last.

Manually enable the setting

1. Navigate to Access > Guided Configuration.
2. On the far-right of the row for your header-based application, select the padlock.
3. Navigate to Access > Federation > OAuth Client/Resource Server > Providers.
4. Select the provider for your Azure AD B2C configuration.
5. Check the **Use Auto JWT** box.
6. Select Discover.
7. Select Save.
8. The **Key (JWT)** field has the token signing certificate key ID (KID) from OpenID URI metadata.
9. In the top-left corner, select **Apply Access Policy**.
10. Select **Apply**.

For more information, go to [techdocs.f5.com](#) for [OAuth client and resource server troubleshooting tips](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Tutorial: Configure Secure Hybrid Access with Microsoft Entra ID and Silverfort

Article • 04/18/2024

[Silverfort](#) uses agent-less and proxy-less technology to connect your assets on-premises and in the cloud to Microsoft Entra ID. This solution enables organizations to apply identity protection, visibility, and user experience across environments in Microsoft Entra ID. It enables universal risk-based monitoring and assessment of authentication activity for on-premises and cloud environments, and helps to prevent threats.

In this tutorial, learn how to integrate your on-premises Silverfort implementation with Microsoft Entra ID.

Learn more:

- [Microsoft Entra hybrid joined devices](#)
- [Silverfort bridging to Microsoft Entra ID](#)

Silverfort connects assets with Microsoft Entra ID. These bridged assets appear as regular applications in Microsoft Entra ID and can be protected with [Conditional Access](#), single-sign-on (SSO), multifactor authentication, auditing and more. Use Silverfort to connect assets including:

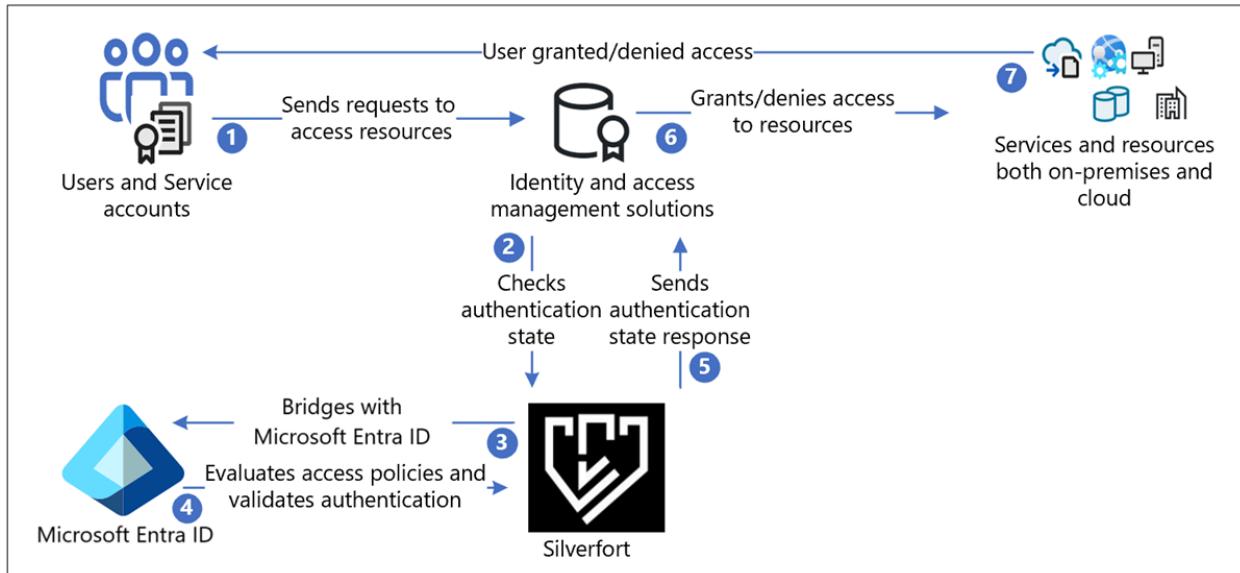
- Legacy and homegrown applications
- Remote desktop and Secure Shell (SSH)
- Command-line tools and other admin access
- File shares and databases
- Infrastructure and industrial systems

Silverfort integrates corporate assets and third-party Identity and Access Management (IAM) platforms, which includes Active Directory Federation Services (AD FS), and Remote Authentication Dial-In User Service (RADIUS) in Microsoft Entra ID. The scenario includes hybrid and multicloud environments.

Use this tutorial to configure and test the Silverfort Microsoft Entra ID bridge in your Microsoft Entra tenant to communicate with your Silverfort implementation. After configuration, you can create Silverfort authentication policies that bridge authentication requests from identity sources to Microsoft Entra ID for SSO. After an application is bridged, you can manage it in Microsoft Entra ID.

Silverfort with Microsoft Entra authentication architecture

The following diagram shows the authentication architecture orchestrated by Silverfort, in a hybrid environment.



User flow

1. Users send authentication request to the original identity provider (IdP) through protocols such as Kerberos, SAML, NTLM, OIDC, and LDAPs
2. Responses are routed as-is to Silverfort for validation to check authentication state
3. Silverfort provides visibility, discovery, and a bridge to Microsoft Entra ID
4. If the application is bridged, the authentication decision passes to Microsoft Entra ID. Microsoft Entra ID evaluates Conditional Access policies and validates authentication.
5. The authentication state response goes as-is from Silverfort to the IdP
6. IdP grants or denies access to the resource
7. Users are notified if access request is granted or denied

Prerequisites

You need Silverfort deployed in your tenant or infrastructure to perform this tutorial. To deploy Silverfort in your tenant or infrastructure, go to silverfort.com [Silverfort](#) to install the Silverfort desktop app on your workstations.

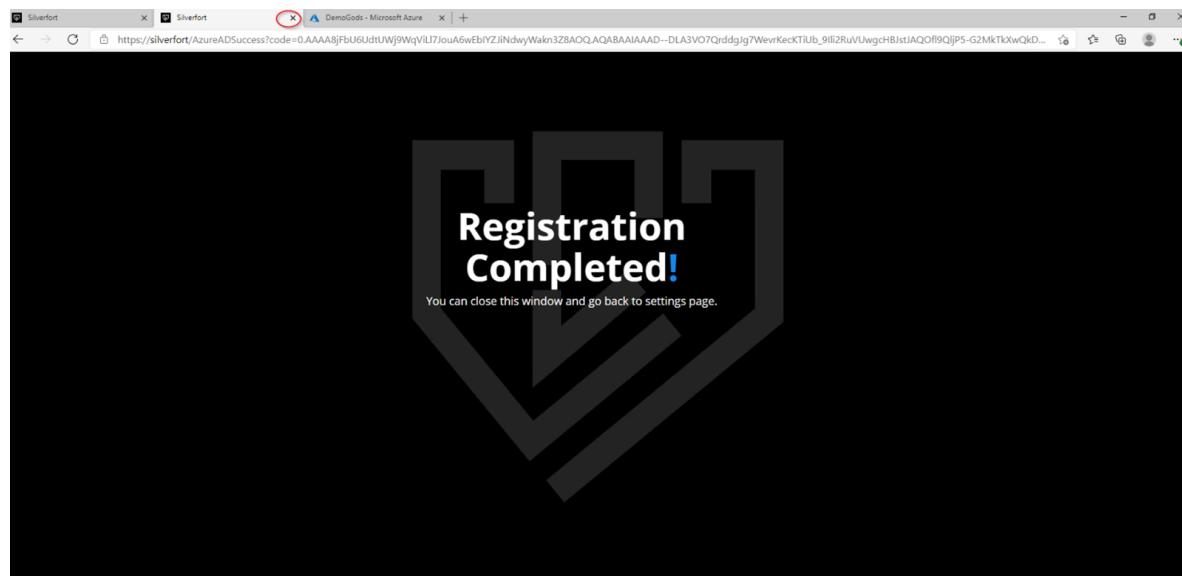
Set up Silverfort Microsoft Entra Adapter in your Microsoft Entra tenant:

- An Azure account with an active subscription

- You can create an [Azure free account](#)
- One of the following roles in your Azure account:
 - Global Administrator
 - Cloud Application Administrator
 - Application Administrator
 - Service Principal Owner
- The Silverfort Microsoft Entra Adapter application in the Microsoft Entra application gallery is preconfigured to support SSO. From the gallery, add the Silverfort Microsoft Entra Adapter to your tenant as an Enterprise application.

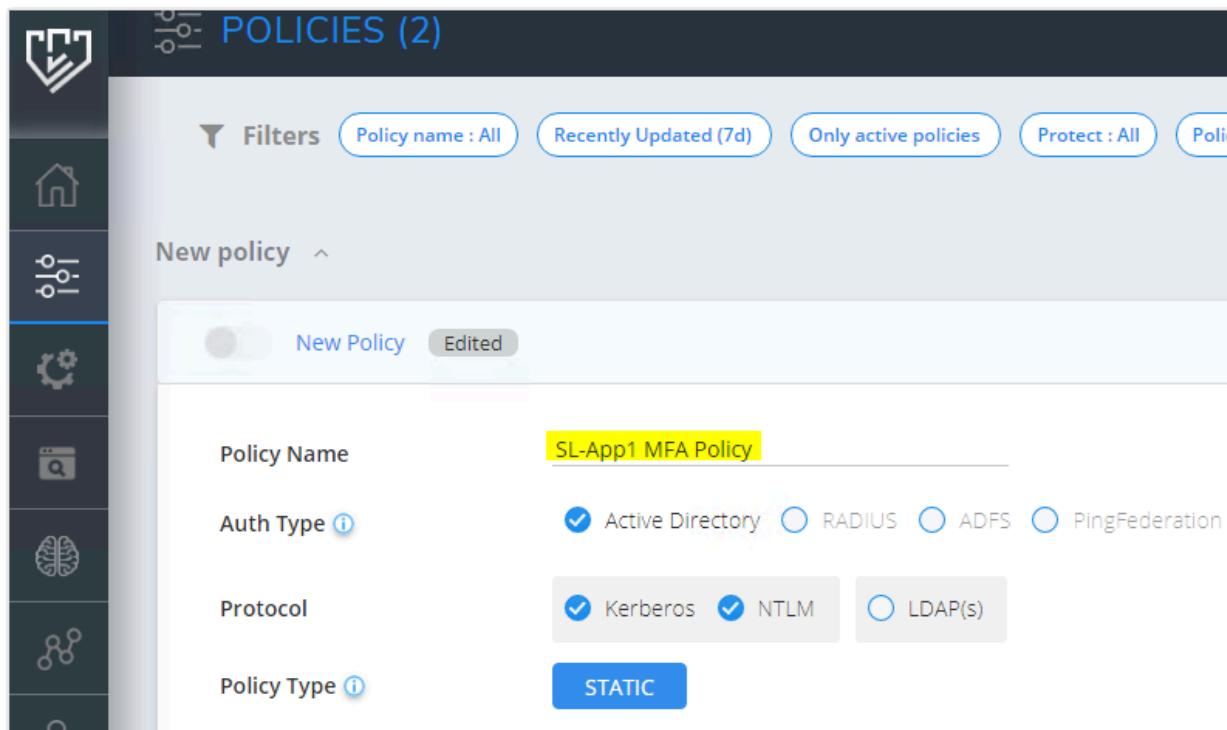
Configure Silverfort and create a policy

1. From a browser, sign in to the Silverfort admin console.
2. In the main menu, navigate to **Settings** and then scroll to **Microsoft Entra ID Bridge Connector** in the General section.
3. Confirm your tenant ID, and then select **Authorize**.
4. Select **Save Changes**.
5. On the **Permissions requested** dialog, select **Accept**.
6. A Registration Completed message appears in a new tab. Close this tab.

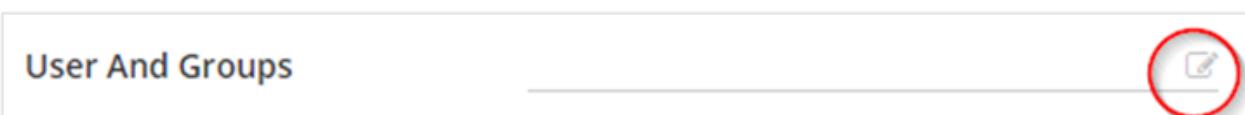


7. On the **Settings** page, select **Save Changes**.
8. Sign in to your Microsoft Entra account. In the left pane, select **Enterprise applications**. The **Silverfort Microsoft Entra Adapter** application appears as registered.

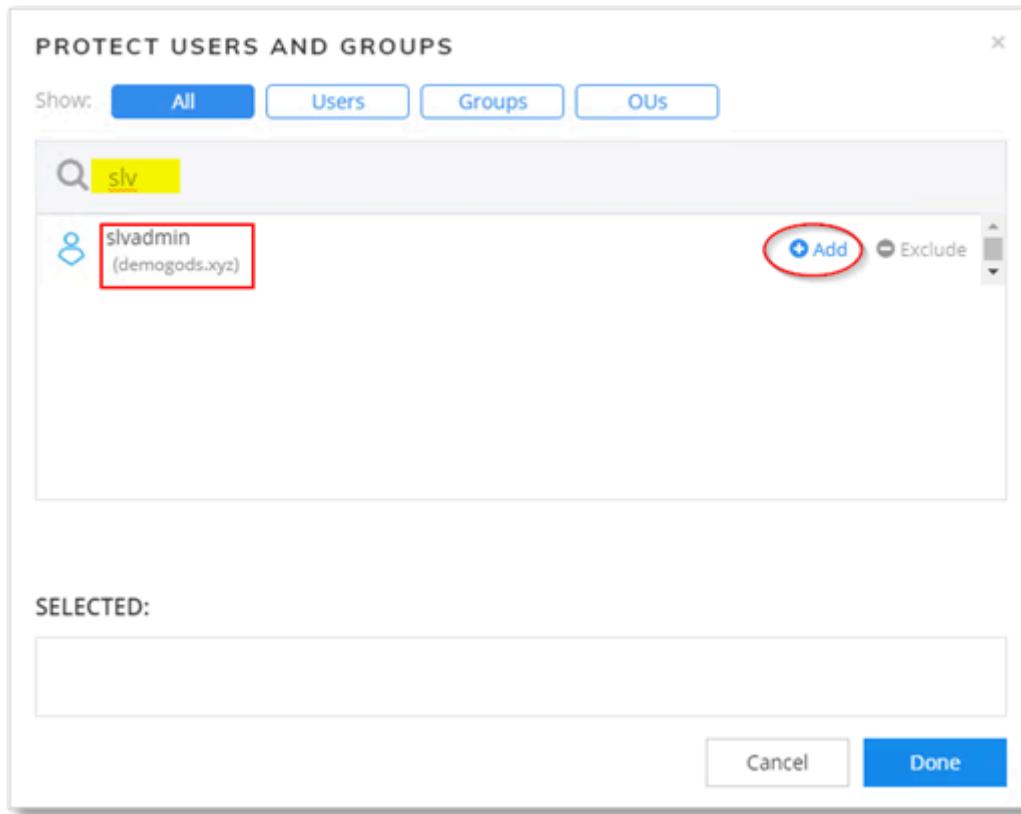
9. In the Silverfort admin console, navigate to the **Policies** page and select **Create Policy**. The **New Policy** dialog appears.
10. Enter a **Policy Name**, the application name to be created in Azure. For example, if you're adding multiple servers or applications for this policy, name it to reflect the resources covered by the policy. In the example, we create a policy for the SL-APP1 server.



11. Select the **Auth Type**, and **Protocol**.
12. In the **Users and Groups** field, select the **Edit** icon to configure users affected by the policy. These users' authentication bridges to Microsoft Entra ID.



13. Search and select users, groups, or Organization Units (OUs).



14. Selected users appear in the **SELECTED** box.



15. Select the **Source** for which the policy applies. In this example, **All Devices** is selected.

User And Groups	slvadmin	
Source	All Devices	

16. Set the **Destination** to SL-App1. Optional: You can select the **edit** button to change or add more resources, or groups of resources.

Destination	SL-App1	
-------------	---------	--

17. For Action, select **Entra ID BRIDGE**.

18. Select **Save**. You're prompted to turn on the policy.
19. In the Entra ID Bridge section, the policy appears on the Policies page.
20. Return to the Microsoft Entra account, and navigate to **Enterprise applications**.
The new Silverfort application appears. You can include this application in Conditional Access policies.

Learn more: [Tutorial: Secure user sign-in events with Microsoft Entra multifactor authentication](#).

Next steps

- [Silverfort Microsoft Entra Adapter](#) ↗
- [Silverfort resources](#) ↗
- [Silverfort, company contact](#) ↗

Troubleshoot password-based single sign-on

Article • 10/29/2024

To use password-based single sign-on (SSO) in My Apps, the browser extension must be installed. The extension downloads automatically when you select an app that's configured for password-based SSO. To learn about using My Apps from an end-user perspective, see [My Apps portal help](#).

My Apps browser extension not installed

Make sure the browser extension is installed. To learn more, see [Plan a Microsoft Entra My Apps deployment](#).

Single sign-on not configured

Make sure password-based single sign-on is configured. To learn more, see [Configure password-based single sign-on](#).

Users not assigned

Make sure the user is assigned to the app. To learn more, see [Assign a user or group to an app](#).

Credentials are filled in, but the extension does not submit them

This problem typically happens if the application vendor has changed their sign-in page recently to add a field, changed an identifier used for detecting the username and password fields, or modified how the sign-in experience works for their application. Fortunately, in many instances, Microsoft can work with application vendors to rapidly resolve these issues.

While Microsoft has technologies to automatically detect when integrations break, it might not be possible to find the issues right away, or the issues take some time to fix. In the case when one of these integrations does not work correctly, open a support case so it can be fixed as quickly as possible.

If you are in contact with this application's vendor, send them our way so Microsoft can work with them to natively integrate their application with Microsoft Entra ID. You can send the vendor to the [Listing your application in the Microsoft Entra application gallery](#) to get them started.

Credentials are filled in and submitted, but the page indicates the credentials are incorrect

To resolve this issue, first try these things:

- Have the user first try to **sign in to the application website directly** with the credentials stored for them.
 - If sign-in works, then have the user click the **Update credentials** button on the **Application Tile** in the **Apps** section of [My Apps](#) to update them to the latest known working username and password.
 - If you, or another administrator assigned the credentials for this user, find the user or group's application assignment by navigating to the **Users & Groups** tab of the application, selecting the assignment and clicking the **Update Credentials** button.
- If the user assigned their own credentials, have the user **check to be sure that their password has not expired in the application** and if so, **update their expired password** by signing in to the application directly.
 - After the password has been updated in the application, request the user to click the **Update credentials** button on the **Application Tile** in the **Apps** section of [My Apps](#) to update them to the latest known working username and password.
 - If you, or another administrator assigned the credentials for this user, find the user or group's application assignment by navigating to the **Users & Groups** tab of the application, selecting the assignment and clicking the **Update Credentials** button.
- Ensure that the My Apps browser extension is running and enabled in your user's browser.
- Ensure that your users are not trying to sign in to the application from My Apps while in **incognito, inPrivate, or Private mode**. The My Apps extension is not supported in these modes.

In case the previous suggestions do not work, it could be the case that a change has occurred on the application side that has temporarily broken the application's integration with Microsoft Entra ID. For example, this can occur when the application vendor introduces a script on their page which behaves differently for manual vs automated input, which causes automated integration, like our own, to break. Fortunately, in many instances, Microsoft can work with application vendors to rapidly resolve these issues.

While Microsoft has technologies to automatically detect when application integrations break, it might not be possible to find the issues right away, or the issues might take some time to fix. When an integration does not work correctly, you can open a support case to get it fixed as quickly as possible.

In addition to this, **if you are in contact with this application's vendor, send them our way** so we can work with them to natively integrate their application with Microsoft Entra ID. You can send the vendor to the [Listing your application in the Microsoft Entra application gallery](#) to get them started.

Check if the application's login page has changed recently or requires an additional field

If the application's login page has changed drastically, sometimes this causes our integrations to break. An example of this is when an application vendor adds a sign-in field, a captcha, or multi-factor authentication to their experiences. Fortunately, in many instances, Microsoft can work with application vendors to rapidly resolve these issues.

While Microsoft has technologies to automatically detect when application integrations break, it might not be possible to find the issues right away, or the issues might take some time to fix. When an integration does not work correctly, you can open a support case to get it fixed as quickly as possible.

In addition to this, **if you are in contact with this application's vendor, send them our way** so we can work with them to natively integrate their application with Microsoft Entra ID. You can send the vendor to the [Listing your application in the Microsoft Entra application gallery](#) to get them started.

Capture sign-in fields for an app

Sign-in field capture is supported only for HTML-enabled sign-in pages. It's not supported for non-standard sign-in pages, like those that use Adobe Flash or other non-HTML-enabled technologies.

There are two ways to capture sign-in fields for your custom apps:

- **Automatic sign-in field capture** works well with most HTML-enabled sign-in pages, *if they use well-known DIV IDs* for the user name and password fields. The HTML on the page is scraped to find DIV IDs that match certain criteria. That metadata is saved so that it can be replayed to the app later.
- **Manual sign-in field capture** is used if the app vendor *doesn't label the sign-in input fields*. Manual capture is also used if the vendor *renders multiple fields that can't be auto-detected*. Microsoft Entra ID can store data for as many fields as there are on the sign-in page, if you tell it where those fields are on the page.

In general, if automatic sign-in field capture doesn't work, try the manual option.

Tip

Steps in this article might vary slightly based on the portal you start from.

Automatically capture sign-in fields for an app

To configure password-based SSO by using automatic sign-in field capture, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. Select the app that you want to configure for SSO.
4. After the app loads, select **Single sign-on** in the navigation pane on the left side.
5. Select **Password-based Sign-on mode**.
6. Enter the **Sign-on URL**, which is the URL of the page where users enter their user name and password to sign in. *Make sure that the sign-in fields are visible on the page for the URL that you provide*.
7. Select **Save**. The page is automatically scraped for the user name and password input boxes. You can now use Microsoft Entra ID to securely transmit passwords to that app by using the My Apps browser extension.

Manually capture sign-in fields for an app

To manually capture sign-in fields, you must have the My Apps browser extension installed. Also, your browser can't be running in *inPrivate*, *incognito*, or *private* mode.

To configure password-based SSO for an app by using manual sign-in field capture, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. Select the app that you want to configure for SSO.
4. After the app loads, select **Single sign-on** in the navigation pane on the left side.
5. Select **Password-based Sign-on mode**.
6. Enter the **Sign-on URL**, which is the page where users enter their user name and password to sign in. *Make sure that the sign-in fields are visible on the page for the URL that you provide.*
7. Select **Configure <appname> Password Single Sign-on Settings**.
8. Select **Manually detect sign-in fields**.
9. Select **Ok**.
10. Select **Save**.
11. Follow the instructions to use My Apps.

Troubleshoot problems

I get a “We couldn’t find any sign-in fields at that URL” error

You get this error message when automatic detection of sign-in fields fails. To resolve the issue, try manual sign-in field detection. See the [Manually capture sign-in fields for an application](#) section of this article.

I get an “Unable to save single sign-on configuration” error

Rarely, updating the SSO configuration fails. To resolve this problem, try saving the configuration again.

If you keep getting the error, open a support case. Include the information that's described in the [View portal notification details](#) and [Send notification details to a support engineer to get help](#) sections of this article.

I can't manually detect sign-in fields for my app

You might observe the following behaviors when manual detection isn't working:

- The manual capture process appeared to work, but the captured fields aren't correct.
- The correct fields don't get highlighted when the capture process runs.
- The capture process takes you to the app's sign-in page as expected, but nothing happens.
- Manual capture appears to work, but SSO doesn't happen when users navigate to the app from My Apps.

If you experience any of these problems, do the following things:

- Make sure that you have the latest version of the My Apps browser extension *installed and enabled*.
- Make sure that your browser isn't in *incognito*, *inPrivate*, or *Private* mode during the capture process. The My Apps extension isn't supported in these modes.
- Make sure that your users aren't trying to sign in to the app from My Apps while in *incognito*, *inPrivate*, or *Private mode*.
- Try the manual capture process again. Make sure that the red markers are over the correct fields.
- If the manual capture process seems to stop responding or the sign-in page doesn't respond, try the manual capture process again. But this time, after completing the process, press the F12 key to open your browser's developer console. Select the **console** tab. Type `window.location=<the sign-in URL that you specified when configuring the app>`, and then press Enter. This forces a page redirect that ends the capture process and stores the fields that were captured.

I can't add another user to my password-based SSO app

A user cannot have more than 48 credentials configured across all password SSO apps where the user is directly assigned.

If you want to add more apps with password-based SSO to a user, consider assigning the app to a group the user is a direct member of, and configuring the credential for the group. Note that the credentials configured for the group will be available for all members of the group.

I can't add another group to my password-based SSO app

Each password-based SSO app has a limit of 48 groups which are assigned and have had credentials configured for them. If you want to add additional groups, you can either:

- Add additional instance of the app
- Remove groups who are no longer using the app

Request support

If you get an error message when you set up SSO and assign users, open a support ticket. Include as much of the following information as possible:

- Correlation error ID
- UPN (user email address)
- TenantID
- Browser type
- Time zone and time/time frame when the error occurred
- Fiddler traces

View portal notification details

To see the details of any portal notification, follow these steps:

1. Select the **Notifications** icon (the bell) in the upper-right corner of the Microsoft Entra admin center.
2. Select any notification that shows an **Error** state. (They have a red "!".)

 **Note**

You can't select notifications that are in the *Successful* or *In Progress* state.

3. The **Notification Details** pane opens. Read the information to learn about the problem.
4. If you still need help, share the information with a support engineer or the product group. Select the **copy** icon to the right of the **Copy error** box to copy the notification details to share.

Send notification details to a support engineer to get help

It's important that you share *all* the details that are listed in this section with support so that they can help you quickly. To record it, you can take a screenshot or select **Copy error**.

The following information explains what each notification item means and provides examples.

Essential notification items

- **Title:** the descriptive title of the notification.

Example: *Application proxy settings*

- **Description:** what occurred as a result of the operation.

Example: *Internal URL entered is already being used by another application.*

- **Notification ID:** the unique ID of the notification.

Example: *clientNotification-2adbfc06-2073-4678-a69f-7eb78d96b068*

- **Client Request ID:** the specific request ID that your browser made.

Example: *0000aaaa-11bb-cccc-dd22-eeeeee333333*

- **Time Stamp UTC:** the timestamp of when the notification occurred, in UTC.

Example: *2017-03-23T19:50:43.7583681Z*

- **Internal Transaction ID:** the internal ID that's used to look up the error in our systems.

Example: *71a2f329-ca29-402f-aa72-bc00a7aca603*

- **UPN:** The user who ran the operation.

Example: *tperkins@f128.info*

- **Tenant ID:** the unique ID of the tenant that the user who ran the operation is a member of.

Example: *aaaabbbb-0000-cccc-1111-dddd2222eeee*

- **User object ID:** The unique ID of the user who ran the operation.

Example: *aaaaaaaa-0000-1111-2222-bbbbbbbbbb*

Detailed notification items

- **Display Name:** (can be empty) a more-detailed display name for the error.

Example: *Application proxy settings*

- **Status:** the specific status of the notification.

Example: *Failed*

- **Object ID:** (can be empty) the object ID against which the operation was run.

Example: *aaaaaaaa-0000-1111-2222-bbbbbbbbbbbb*

- **Details:** the detailed description of what occurred as a result of the operation.

Example: *Internal url '<https://bing.com/>' is invalid since it is already in use.*

- **Copy error:** Enables you to select the **copy icon** to the right of the **Copy error** textbox to copy the notification details to help with support.

Example:

```
{"errorCode": "InternalUrl\\_Duplicate", "localizedErrorDetails":  
{"errorDetail": "Internal url 'https://google.com/' is invalid since it is  
already in use"}, "operationResults": [  
{"objectId": null, "displayName": null, "status": 0, "details": "Internal url  
'https://bing.com/' is invalid since it is already in  
use"}], "timeStampUtc": "2017-03-  
23T19:50:26.465743Z", "clientRequestId": "00aa00aa-bb11-cc22-dd33-  
44ee44ee44ee", "internalTransactionId": "aaaaaaaa-0000-1111-2222-  
bbbbbbbbbbbb", "upn": "tperkins@f128.info", "tenantId": "aaaabbbb-0000-cccc-1111-  
dddd2222eeee", "userObjectId": "aaaaaaaa-0000-1111-2222-bbbbbbbbbbbb"}
```

Next steps

- [Quickstart Series on Application Management](#)
- [Plan a My Apps deployment](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Troubleshoot SAML-based single sign-on

Article • 04/25/2025

If you encounter a problem when configuring an application, verify you followed all the steps in the tutorial for the application. In the application's configuration, you have inline documentation on how to configure the application. Also, you can access the [List of tutorials on how to integrate SaaS apps with Microsoft Entra ID](#) for a detail step-by-step guidance.

Can't add another instance of the application

To add a second instance of an application, you need to be able to:

- Configure a unique identifier for the second instance. You aren't able to configure the same identifier used for the first instance.
- Configure a different certificate than the one used for the first instance.

If the application doesn't support any of the listed options, you aren't able to configure a second instance.

Can't add the Identifier or the Reply URL

If you're not able to configure the Identifier or the Reply URL, confirm the Identifier and Reply URL values match the patterns preconfigured for the application.

To know the patterns preconfigured for the application:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#). Go to step 4 if you're already in the application configuration pane in Microsoft Entra ID.
2. Browse to **Entra ID > Enterprise apps > All applications**.
3. Select the application you want to configure single sign-on.
4. Once the application loads, select the **Single sign-on** from the application's left-hand navigation menu.
5. Select **SAML-based Sign-on** from the **Mode** dropdown.
6. Go to the **Identifier** or **Reply URL** textbox, under the **Domain and URLs** section.
7. There are three ways to know the supported patterns for the application.
 - In the textbox, you see the supported pattern as a placeholder, for example:
`https://contoso.com`.
 - if the pattern isn't supported, you see a red exclamation mark when you try to enter the value in the textbox. If you hover your mouse over the red exclamation mark, you see the supported patterns.

- In the tutorial for the application, you can also get information about the supported patterns. Under the **Configure Microsoft Entra single sign-on** section. Go to the step for configured the values under the **Domain and URLs** section.

If the values don't match with the patterns preconfigured in Microsoft Entra ID, you can work with the application vendor to get values that match the pattern preconfigured in Microsoft Entra ID. If the application is multi-tenant, a single identifier is allowed when the principal object isn't the primary instance of the app.

Where do I set the EntityID (User Identifier) format

You won't be able to select the EntityID (User Identifier) format that Microsoft Entra ID sends to the application in the response after user authentication.

Microsoft Entra ID selects the format for the NameID attribute (User Identifier) based on the value selected or the format requested by the application in the SAML AuthRequest. For more information visit the article [Single sign-on SAML protocol](#) under the section NameIDPolicy,

Can't find the Microsoft Entra metadata to complete the configuration with the application

To download the application metadata or certificate from Microsoft Entra ID, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Entra ID > Enterprise apps > All applications**.
3. Select the application you configure for single sign-on.
4. Once the application loads, select **Single sign-on** from the application's left-hand navigation menu.
5. Go to **SAML Signing Certificate** section, then select **Download** column value. Depending on what the application requires configuring single sign-on, you see either the option to download the Metadata XML or the Certificate.

Microsoft Entra doesn't provide a URL to get the metadata. The metadata can only be retrieved as an XML file.

Customize SAML claims sent to an application

To learn how to customize the SAML attribute claims sent to your application, see [Claims mapping in Microsoft Entra ID](#) for more information.

Next steps

- [Quickstart Series on Application Management](#)

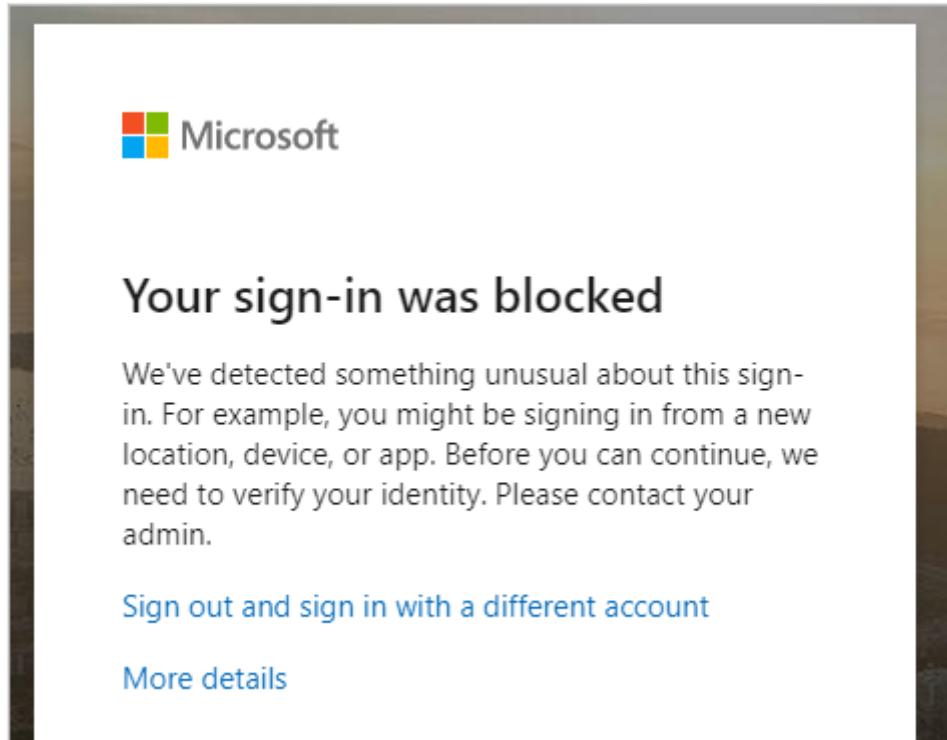
Your sign-in was blocked

Article • 04/29/2025

This article provides information for resolving a blocked sign-in to the Microsoft Application Network portal.

Symptoms

The user sees this message when trying to sign in to the Microsoft Application Network portal.



Cause

The guest user is federated to a home tenant that is also a Microsoft Entra tenant. The guest user is at high risk. High risk users aren't allowed to access resources. All high risk users (employees, guests, or vendors) must remediate their risk to access resources. For guest users, this user risk comes from the home tenant and the policy comes from the resource tenant.

Solutions

- MFA registered guest users remediate their own user risk. The guest user [resets or changes a secured password](#) at their home tenant (this needs MFA and self service password reset (SSPR) at the home tenant). The secured password change or reset must be initiated on Microsoft Entra ID and not on-premises.

- Guest users have their administrators remediate their risk. In this case, the administrator resets a password (temporary password generation). The guest user's administrator can go to <https://aka.ms/RiskyUsers> and select **Reset password**.
- Guest users have their administrators dismiss their risk. The admin can go to <https://aka.ms/RiskyUsers> and select **Dismiss user risk**. However, the administrator must do the due diligence to make sure the risk assessment was a false positive before dismissing the user risk. Otherwise, resources are put at risk by suppressing a risk assessment without investigation.

Debug SAML-based single sign-on to applications

Article • 01/29/2025

In this article, you learn how to find and fix [single sign-on](#) issues for applications in Microsoft Entra ID that use SAML-based single sign-on.

Before you begin

We recommend installing the [My Apps Secure Sign-in Extension](#). This browser extension makes it easy to gather the SAML request and SAML response information that you need to resolve issues with single sign-on. In case you can't install the extension, this article shows you how to resolve issues both with and without the extension installed.

To download and install the My Apps Secure Sign-in Extension, use one of the following links.

- [Chrome](#)
- [Microsoft Edge](#)

Test SAML-based single sign-on

To test SAML-based single sign-on between Microsoft Entra ID and a target application:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. From the list of enterprise applications, select the application for which you want to test single sign-on, and then from the options on the left, select **Single sign-on**.
4. On the **Select a single sign-on method** pane, select **SAML**.
5. To open the SAML-based single sign-on testing experience, go to **Test single sign-on** (step 5). If the **Test** button is greyed out, you need to fill out and save the required attributes first in the **Basic SAML Configuration** section.
6. In the **Test single sign-on** page, use your corporate credentials to sign in to the target application. You can sign in as the current user or as a different user. If you

sign in as a different user, a prompt asks you to authenticate.

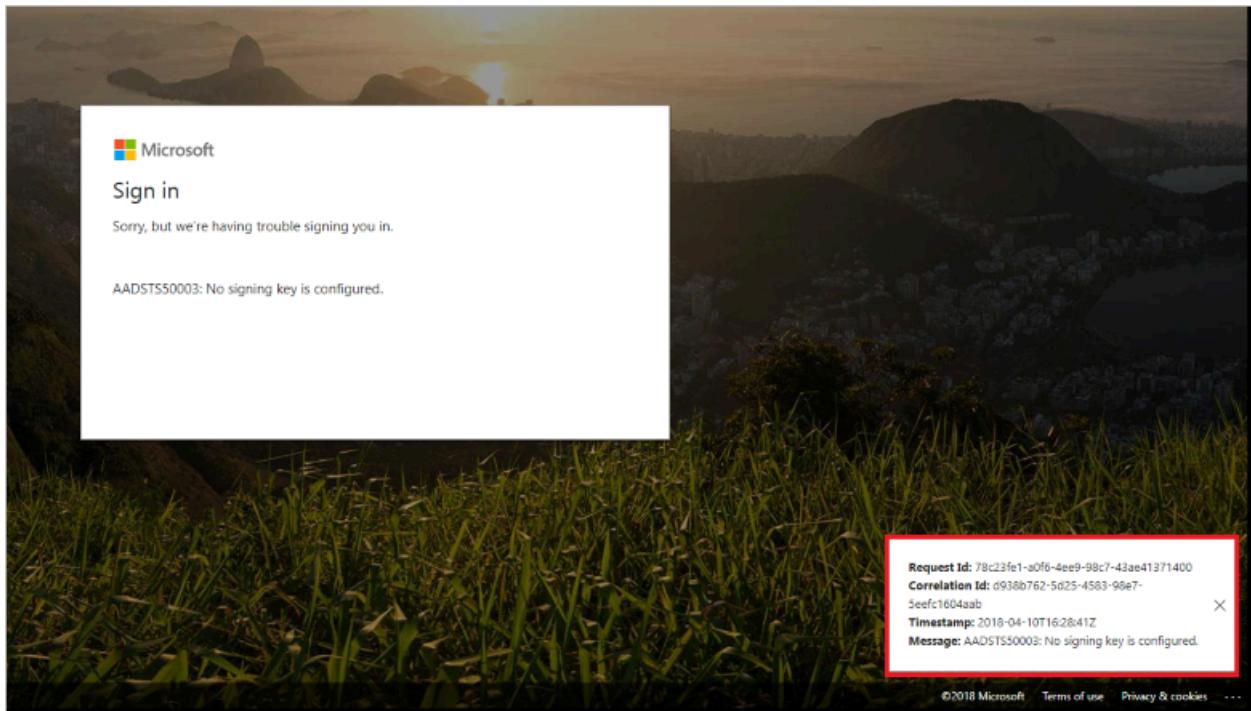
The screenshot shows the Microsoft Azure portal interface for managing application configurations. The main title is 'Test single sign-on with Salesforce - Ignite'. The left sidebar lists 'Applications > Salesforce - Ignite - Single sign-on > SSO'. The main content area is divided into sections: Step 3 (SAML Signing Certificate), Step 4 (Set up Salesforce - Ignite), and Step 5 (Test single sign-on with Salesforce - Ignite). Step 4 contains a note about configuring the application and a 'View step-by-step instructions' link. Step 5 contains a 'Test' button. A 'Resolving errors' panel is open, displaying an error message with Request ID, Correlation ID, Timestamp, and Message details. It also includes a 'Get resolution guidance' link and a 'Fix It' button.

If you're able to sign in, the test is successful. In this case, Microsoft Entra ID issued a SAML response token to the application. The application used the SAML token to successfully sign you in.

If you have an error on the company sign-in page or the application's page, use one of the next sections to resolve the error.

Resolve a sign-in error on your company sign-in page

When you try to sign in, you might see an error on your company sign-in page that's similar to the following example.



To debug this error, you need the error message and the SAML request. The My Apps Secure Sign-in Extension automatically gathers this information and displays resolution guidance on Microsoft Entra ID.

To resolve the sign-in error with the My Apps Secure Sign-in Extension installed

1. When an error occurs, the extension redirects you back to the Microsoft Entra ID [Test single sign-on](#) page.
2. On the [Test single sign-on](#) page, select [Download the SAML request](#).
3. You should see specific resolution guidance based on the error and the values in the SAML request.
4. You see a **Fix it** button to automatically update the configuration in Microsoft Entra ID to resolve the issue. If you don't see this button, then the sign-in issue isn't due to a misconfiguration on Microsoft Entra ID.

If no resolution is provided for the sign-in error, we suggest that you use the feedback textbox to inform us.

To resolve the error without installing the My Apps Secure Sign-in Extension

1. Copy the error message at the bottom right corner of the page. The error message includes:

- A CorrelationID and Timestamp. These values are important when you create a support case with Microsoft because they help the engineers to identify your problem and provide an accurate resolution to your issue.
 - A statement identifying the root cause of the problem.
2. Go back to Microsoft Entra ID and find the **Test single sign-on** page.
 3. In the text box above **Get resolution guidance**, paste the error message.
 4. Select **Get resolution guidance** to display steps for resolving the issue. The guidance might require information from the SAML request or SAML response. If you're not using the My Apps Secure Sign-in Extension, you might need a tool such as [Fiddler](#) to retrieve the SAML request and response.
 5. Verify that the destination in the SAML request corresponds to the SAML Single Sign-on Service URL obtained from Microsoft Entra ID.
 6. Verify the issuer in the SAML request is the same identifier configured for the application in Microsoft Entra ID. Microsoft Entra ID uses the issuer to find an application in your directory.
 7. Verify AssertionConsumerServiceURL is where the application expects to receive the SAML token from Microsoft Entra ID. You can configure this value in Microsoft Entra ID, but it's not mandatory if it's part of the SAML request.

Resolve a sign-in error on the application page

You might sign in successfully and then see an error on the application's page. This error occurs when Microsoft Entra ID issued a token to the application, but the application doesn't accept the response.

To resolve the error, follow these steps, or watch this [short video about how to use Microsoft Entra ID to troubleshoot SAML SSO](#):

1. If the application is in the Microsoft Entra Gallery, verify that you followed all the steps for integrating the application with Microsoft Entra ID. To find the integration instructions for your application, see the [list of SaaS application integration tutorials](#).
2. Retrieve the SAML response.
 - If the My Apps Secure Sign-in extension is installed, from the **Test single sign-on** page, select **download the SAML response**.
 - If the extension isn't installed, use a tool such as [Fiddler](#) to retrieve the SAML response.
3. Notice these elements in the SAML response token:

- User unique identifier of NameID value and format
- Claims issued in the token
- Certificate used to sign the token.

For more information on the SAML response, see [Single Sign-on SAML protocol](#).

4. Now that you're done reviewing the SAML response, see [Error on an application's page after signing in](#) for guidance on how to resolve the problem.
5. If you're still not able to sign in successfully, you can ask the application vendor what is missing from the SAML response.

Next steps

Now that single sign-on is working to your application, you could [Automate user provisioning and deprovisioning to SaaS applications](#) or [get started with Conditional Access](#).

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Unexpected consent prompt when signing in to an application

Article • 10/23/2023

Many applications that integrate with Microsoft Entra ID require permissions to various resources in order to run. When these resources are also integrated with Microsoft Entra ID, permission to access them is requested using the Microsoft Entra consent framework. These requests result in a consent prompt being shown the first time an application is used, which is often a one-time operation.

In certain scenarios, additional consent prompts can appear when a user attempts to sign-in. In this article, we diagnose the reason for the unexpected consent prompts showing, and how to troubleshoot.

<https://www.youtube-nocookie.com/embed/a1AjdvNDda4> ↗

Scenarios in which users see consent prompts

Further prompts can be expected in various scenarios:

- The application has been configured to require assignment. Individual user consent isn't currently supported for apps that require assignment; thus the permissions must be granted by an admin for the whole directory. If you configure an application to require assignment, be sure to also grant tenant-wide admin consent so that assigned user can sign-in.
- The set of permissions required by the application has changed by the developer and needs to be granted again.
- The user who originally consented to the application wasn't an administrator, and now a different (nonadmin) user is using the application for the first time.
- The user who originally consented to the application was an administrator, but they didn't consent on-behalf of the entire organization.
- The application is using [incremental and dynamic consent](#) to request further permissions after consent was initially granted. Incremental and dynamic consent is often used when optional features of an application require permissions beyond those required for baseline functionality.
- Consent was revoked after being granted initially.

- The developer has configured the application to require a consent prompt every time it's used (note: this behavior isn't best practice).

ⓘ Note

Following Microsoft's recommendations and best practices, many organizations have disabled or limited users' permission to grant consent to apps. If an application forces users to grant consent every time they sign in, most users will be blocked from using these applications even if an administrator grants tenant-wide admin consent. If you encounter an application which is requiring user consent even after admin consent has been granted, check with the app publisher to see if they have a setting or option to stop forcing user consent on every sign in.

ⓘ Tip

Steps in this article may vary slightly based on the portal you start from.

Troubleshooting steps

Compare permissions requested and granted for the applications

To ensure the permissions granted for the application are up-to-date, you can compare the permissions that are being requested by the application with the permissions already granted in the tenant.

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. Enter the name of the existing application in the search box, and then select the application from the search results.
4. Under Security in the left-hand navigation, choose **Permissions**
5. View the list of already granted permissions from the table on the Permissions page
6. To view the requested permissions, select the **Grant admin consent** button. This opens a consent prompt listing all of the requested permissions. Don't select **Accept** on the consent prompt unless you're sure you want to grant tenant-wide admin consent.

7. Within the consent prompt, expand the listed permissions and compare with the table on the permissions page. If any are present in the consent prompt but not the permissions page, that permission has yet to be consented to. Unconsented permissions may be the cause for unexpected consent prompts showing for the application.

View user assignment settings

If the application requires assignment, individual users can't consent for themselves. To check if assignment is required for the application, do the following:

1. On the application's page, Select **Properties** under **Manage**.
2. Check to see if **Assignment required?** is set to **Yes**.
3. If set to yes, then an admin must consent to the permissions on behalf of the entire organization.

Review tenant-wide user consent settings

Determining whether an individual user can consent to an application can be configured by every organization, and may differ from directory to directory. Even if every permission doesn't require admin consent by default, your organization may have disabled user consent entirely, preventing an individual user to consent for themselves for an application. To view your organization's user consent settings, do the following:

1. Navigate to the **Enterprise applications** page of the Microsoft Entra admin center.
2. Under **Security**, choose **Consent and permissions**.
3. View the user consent settings. If set to **Do not allow user consent**, users are never able to consent on behalf of themselves for an application.

Next steps

- [Apps, permissions, and consent in Azure Active Directory \(v1.0 endpoint\)](#)
- [Scopes, permissions, and consent in the Microsoft identity platform \(v2.0 endpoint\)](#)
- [Unexpected error when performing consent to an application](#)

Unexpected error when performing consent to an application

Article • 02/27/2025

This article discusses errors that can occur during the process of consenting to an application. If you're troubleshooting unexpected consent prompts that don't contain any error messages, see [Authentication Scenarios for Microsoft Entra ID](#).

Many applications that integrate with Microsoft Entra ID require permissions to access other resources in order to function. When these resources are also integrated with Microsoft Entra ID, the permission to access them is often requested using the common consent framework. A consent prompt is displayed, which generally occurs the first time an application is in use. It can also occur on a subsequent use of the application.

Certain conditions must be true for a user to consent to the permissions an application requires. If these conditions aren't met, the following errors can occur.

Requesting not authorized permissions error

- **AADSTS90093:** `clientAppDisplayName` is requesting one or more permissions that you aren't authorized to grant. Contact an administrator, who can consent to this application on your behalf.
- **AADSTS90094:** `clientAppDisplayName` needs permission to access resources in your organization that only an admin can grant. Ask an admin to grant permission to this app before you can use it.

This error occurs when a user who isn't an administrator attempts to use an application that is requesting permissions that only an administrator can grant. To resolve this error, an administrator should grant access to the application on behalf of their organization.

This error can also occur when a user is prevented from consenting to an application due to Microsoft detecting that the permissions request is risky. In this case, an audit event is also logged with a category of **ApplicationManagement**, Activity Type of **Consent to application** and Status Reason of **Risky application detected**.

Another scenario in which this error might occur is when the user assignment is required for the application, but no administrator consent was provided. In this case, the administrator must first provide tenant-wide admin consent for the application.

Policy prevents granting permissions error

- **AADSTS90093:** An administrator of `tenantDisplayName` set a policy that prevents you from granting `name of app` the permissions it's requesting. Contact an administrator of `tenantDisplayName`, who can grant permissions to this app on your behalf.

This error occurs when an administrator turns off the ability for users to consent to applications. Then a nonadministrator user attempts to use an application that requires consent. To resolve this error, an administrator should grant access to the application on behalf of their organization.

Intermittent problem error

- **AADSTS90090:** It looks like the sign-in process encountered an intermittent problem recording the permissions you attempted to grant to `clientAppDisplayName`. Try again later.

This error indicates that an intermittent service side issue occurred. It can be resolved by attempting to consent to the application again.

Resource not available in tenant error

- **AADSTS65005:** `clientAppDisplayName` is requesting access to a resource `resourceAppDisplayName` that isn't available in your organization `tenantDisplayName`.

Ensure that these resources that provide the permissions requested are available in your tenant or contact an administrator of `tenantDisplayName`. Otherwise, there's a misconfiguration in how the application requests resources, and you should contact the application developer.

Permissions mismatch error

- **AADSTS65005:** The app requested consent to access resource `resourceAppDisplayName`. This request failed because it doesn't match how the app was preconfigured during app registration. Contact the app vendor.**

These errors occur when the app a user is trying to consent to requests permissions to access a resource application that can't be found in the organization's directory (tenant).

This situation can occur for several reasons:

- The client application developer configured their application incorrectly, causing it to request access to an invalid resource. In this case, the application developer must update the configuration of the client application to resolve this issue.
- A service principal representing the target resource application doesn't exist in the organization, or existed in the past but is removed. To resolve this issue, a service principal for the resource application must be provisioned in the organization so the client application can request permissions to it. The service principal can be provisioned in many ways, depending on the type of application, including:
 - Acquiring a subscription for the resource application (Microsoft published applications)
 - Consenting to the resource application
 - Granting the application permissions via the Microsoft Entra admin center
 - Adding the application from the Microsoft Entra Application Gallery

Risky app error and warning

- **AADSTS900941:** Administrator consent is required. App is considered risky.
(AdminConsentRequiredDueToRiskyApp)
- This app might be risky. If you trust this app, ask your admin to grant you access.
- **AADSTS900981:** An admin consent request was received for a risky app.
(AdminConsentRequestRiskyAppWarning)
- This app might be risky. Only continue if you trust this app.

Both of these messages are displayed when Microsoft determines that the consent request might be risky. Among many other factors, this error might occur if a [verified publisher](#) isn't added to the app registration. The first error code and message is shown to end-users when the [Admin consent workflow](#) is disabled. The second code and message is shown to end-users when the admin consent workflow is enabled and to admins.

End-users aren't able to grant consent to apps that are detected as risky. Admins are able to, but should evaluate the app carefully and proceed with caution. If the app seems suspicious upon further review, it can be reported to Microsoft from the consent screen.

Next steps

[Scopes, permissions, and consent in the Microsoft identity platform \(v2.0 endpoint\)](#)

[Unexpected consent prompt when signing in to an application](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Troubleshoot application sign-in

Article • 04/25/2025

My Apps is a web-based portal that enables a user with a work or school account in Microsoft Entra ID to view and start cloud-based applications that the Microsoft Entra administrator has granted them access to. My Apps is accessed using a web browser at <https://myapps.microsoft.com>.

To learn more about using Microsoft Entra ID as an identity provider for an app, see the [What is Application Management in Microsoft Entra ID](#). To get up to speed quickly, check out the [Quickstart Series on Application Management](#).

These applications are configured on behalf of the user in the Microsoft Entra admin center. The application must be configured properly and assigned to the user or a group the user is a member of to see the application in My Apps.

The type of apps a user may be seeing fall in the following categories:

- Microsoft 365 Applications
- Microsoft and third-party applications configured with federation-based SSO
- Password-based SSO applications
- Applications with existing SSO solutions

Here are some things to check if an app is appearing or not appearing:

- Make sure the app is added to Microsoft Entra ID and make sure the user is assigned. To learn more, see the [Quickstart Series on Application Management](#).
- If an app was recently added, have the user sign out and back in again.
- If the app requires a license, such as Office, then make sure the user is assigned the appropriate license.
- The time it takes for licensing changes can vary depending on the size and complexity of the group.

General issues to check first

- Make sure the web browser meets the requirements, see [My Apps supported browsers](#).
- Make sure the user's browser has added the URL of the application to its **trusted sites**.
- Make sure to check the application is **configured** correctly.
- Make sure the user's account is **enabled** for sign-ins.
- Make sure the user's account is **not locked out**.
- Make sure the user's **password is not expired or forgotten**.
- Make sure **Multi-Factor Authentication** isn't blocking user access.

- Make sure a **Conditional Access policy** or legacy **Identity Protection** policy isn't blocking user access.
- Make sure that a user's **authentication contact info** is up to date to allow Multi-Factor Authentication or Conditional Access policies to be enforced.
- Make sure to also try clearing your browser's cookies and trying to sign in again.

Problems with the user's account

Access to My Apps can be blocked due to a problem with the user's account. Following are some ways you can troubleshoot and solve problems with users and their account settings:

- [Check if a user account exists in Microsoft Entra ID](#)
- [Check a user's account status](#)
- [Reset a user's password](#)
- [Enable self-service password reset](#)
- [Check a user's multi-factor authentication status](#)
- [Check a user's authentication contact info](#)
- [Check a user's group memberships](#)
- [Check if a user has more than 999 app role assignments](#)
- [Check a user's assigned licenses](#)
- [Assign a user a license](#)

Check if a user account exists in Microsoft Entra ID

To check if a user's account is present, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#).
2. Browse to **Entra ID > Users**.
3. Search for the user you're interested in and select the row to view the details of the user.
4. Check the properties of the user object to be sure that they look as you expect and no data is missing.

Check a user's account status

To check a user's account status, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#).
2. Browse to **Entra ID > Users**.
3. Search for the user you're interested in and select the row with the user's details.
4. Select **Profile**.
5. Under **Settings** ensure that **Block sign in** is set to **No**.

Reset a user's password

To reset a user's password, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#).
2. Browse to **Entra ID > Users**.
3. Search for the user you're interested in and select the row with the user's details.
4. Select the **Reset password** button at the top of the user pane.
5. Select the **Reset password** button on the Reset password pane that appears.
6. Copy the **temporary password** or enter a new password for the user.
7. Communicate this new password to the user. They might be required to change this password during their next sign-in to Microsoft Entra ID.

Enable self-service password reset

To enable self-service password reset, follow these deployment steps:

- [Enable users to reset their Microsoft Entra passwords](#)
- [Enable users to reset or change their Active Directory on-premises passwords](#)

Check a user's multi-factor authentication status

To check a user's multi-factor authentication status, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#).
2. Browse to **Entra ID > Users**.
3. Select the **Per-user MFA** button at the top of the pane.
4. Once the **Multi-Factor Authentication** administration portal loads, ensure you are on the **Users** tab.
5. Find the user in the list of users by searching, filtering, or sorting.
6. Select the user from the list of users and **Enable**, **Disable**, or **Enforce** multi-factor authentication as desired.

Note

If a user is in an **Enforced** state, you may set them to **Disabled** temporarily to let them back into their account. Once they are back in, you can then change their state to **Enabled** again to require them to re-register their contact information during their next sign-in. Alternatively, you can follow the steps in the [Check a user's authentication contact info](#) to verify or set this data for them.

Check a user's authentication contact info

To check a user's authentication contact info used for multifactor authentication, Conditional Access, and Password Reset, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#).
2. Browse to **Entra ID > Users**.
3. Search for the user you're interested in and select the row with the user's details.
4. Select **Authentication method** under **Manage**.
5. Review the data registered for the user and update as needed.

Check a user's group memberships

To check a user's group memberships, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#).
2. Browse to **Entra ID > Users**.
3. Search for the user you're interested in and select the row with the user's details.
4. Select **Groups** to see which groups the user is a member of.

Check if a user has more than 999 app role assignments

If a user has more than 999 app role assignments, then they may not see all of their apps on My Apps.

This is because My Apps currently reads up to 999 app role assignments to determine the apps to which users are assigned. If a user is assigned to more than 999 apps, it isn't possible to control which of those apps show in the My Apps portal.

To check if a user has more than 999 app role assignments, follow these steps:

1. Install the [Microsoft.Graph](#) PowerShell module.
2. Run `Connect-MgGraph -Scopes "User.ReadBasic.All Application.Read.All"` and sign in as at least a [User Administrator](#).
3. Run `(Get-MgUserAppRoleAssignment -UserId "<user-id>" -PageSize 999).Count` to determine the number of app role assignments the user currently has granted.
4. If the result is 999, the user likely has more than 999 app roles assignments.

Check a user's assigned licenses

To check a user's assigned licenses, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#).
2. Browse to **Entra ID > Users**.
3. Search for the user you're interested in and select the row with the user's details.
4. Select **Licenses** to see which licenses the user currently has assigned.

Assign a user a license

To assign a license to a user, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#).
2. Browse to **Entra ID > Users**.
3. Search for the user you're interested in and select the row with the user's details.
4. Select **Licenses** to see which licenses the user currently has assigned.
5. Select the **Assignments** button.
6. Select one or more licenses from the list of available products.
7. Optional: Select **Review license options** to granularly assign products.
8. Select **Save**.

Troubleshooting deep links

Deep links or User access URLs are links your users may use to access their password-SSO applications directly from their browsers URL bars. By navigating to this link, users are automatically signed into the application without having to go to My Apps first. The link is the same one that users use to access these applications from the Microsoft 365 application launcher.

Checking the deep link

To check if you have the correct deep link, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Entra ID > Enterprise apps > All applications**.
3. Enter the name of the existing application in the search box, and then select the application from the search results.
4. Find the label **User Access URL**. Your deep link should match this URL.

Contact support

Open a support ticket with the following information if available:

- Correlation error ID
- UPN (user email address)
- TenantID
- Browser type
- Time zone and time/timeframe during error occurs
- Fiddler traces

Next steps

- [Quickstart Series on Application Management](#)

An app page shows an error message after the user signs in

Article • 04/29/2025

In this scenario, Microsoft Entra ID signs the user in. But the application displays an error message and doesn't let the user finish the sign-in flow. The problem is that the app didn't accept the response that Microsoft Entra ID issued.

There are several possible reasons why the app didn't accept the response from Microsoft Entra ID. If there's an error message or code displayed, use the following resources to diagnose the error:

- [Microsoft Entra authentication and authorization error codes](#)
- [Troubleshooting consent prompt errors](#)

If the error message doesn't clearly identify what's missing from the response, try the following steps:

- If the app is in the Microsoft Entra gallery, verify that you followed the steps in [How to debug SAML-based single sign-on to applications in Microsoft Entra ID](#).
- Use a tool like [Fiddler](#) to capture the SAML request, response, and token.
- Send the SAML response to the app vendor and ask them what's missing.

Attributes are missing from the SAML response

To add an attribute in the Microsoft Entra configuration that is sent in the Microsoft Entra response, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Entra ID > Enterprise apps > All applications**.
3. Enter the name of the existing application in the search box, and then select the application that you want to configure for single sign-on.
4. After the app loads, select **Single sign-on** in the navigation pane.
5. In the **User Attributes** section, select **View and edit all other user attributes**. Here you can change which attributes to send to the app in the SAML token when users sign in.

To add an attribute:

- a. Select **Add attribute**. Enter the **Name**, and select the **Value** from the drop-down list.

- b. Select **Save**. You see the new attribute in the table.
6. Save the configuration.

The next time that the user signs in to the app, Microsoft Entra ID will send the new attribute in the SAML response.

The app can't identify the user

Signing in to the app fails because the SAML response is missing an attribute such as a role. Or it fails because the app expects a different format or value for the **NameID** (User Identifier) attribute.

If you're using [Microsoft Entra ID automated user provisioning](#) to create, maintain, and remove users in the app, verify that the user is provisioned to the SaaS app. For more information, see [No users are being provisioned to a Microsoft Entra Gallery application](#).

Add an attribute to the Microsoft Entra app configuration

To change the User Identifier value, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Entra ID > Enterprise apps > All applications**.
3. Select the app that you want to configure for SSO.
4. After the app loads, select **Single sign-on** in the navigation pane.
5. Under **User attributes**, select the unique identifier for the user from the **User Identifier** drop-down list.

Change the NameID format

If the application expects another format for the **NameID** (User Identifier) attribute, see the [Edit nameID](#) section to change the NameID format.

Microsoft Entra ID selects the format for the **NameID** attribute (User Identifier) based on the value selected or the format that's requested by the app in the SAML AuthRequest. For more information, see the "NameIDPolicy" section of [Single sign-on SAML protocol](#).

The app expects a different signature method for the SAML response

To change which parts of the SAML token are digitally signed by Microsoft Entra ID, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Entra ID > Enterprise apps > All applications**.
3. Select the application that you want to configure for single sign-on.
4. After the application loads, select **Single sign-on** in the navigation pane.
5. Under **SAML Signing Certificate**, select **Show advanced certificate signing settings**.
6. Select the **Signing Option** that the app expects from among these options:
 - **Sign SAML response**
 - **Sign SAML response and assertion**
 - **Sign SAML assertion**

The next time that the user signs in to the app, Microsoft Entra ID will sign the part of the SAML response that you selected.

The app expects the SHA-1 signing algorithm

By default, Microsoft Entra ID signs the SAML token by using the most-secure algorithm. We recommend that you don't change the signing algorithm to *SHA-1* unless the app requires SHA-1.

To change the signing algorithm, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Cloud Application Administrator**.
2. Browse to **Entra ID > Enterprise apps > All applications**.
3. Select the app that you want to configure for single sign-on.
4. After the app loads, select **Single sign-on** from the navigation pane on the left side of the app.
5. Under **SAML Signing Certificate**, select **Show advanced certificate signing settings**.
6. Select **SHA-1** as the **Signing Algorithm**.

The next time that the user signs in to the app, Microsoft Entra ID will sign the SAML token by using the SHA-1 algorithm.

Related content

- [How to debug SAML-based single sign-on to applications in Microsoft Entra ID](#)
- [Microsoft Entra authentication and authorization error codes](#)
- [Troubleshooting consent prompt errors](#)

Problems signing in to a Microsoft application

Article • 04/25/2025

Microsoft Applications (like Exchange, SharePoint, Yammer, and so on) are assigned and managed a bit differently than third-party SaaS applications or other applications you integrate with Microsoft Entra ID for single sign-on.

There are three main ways that a user can get access to a Microsoft-published application.

- For applications in the Microsoft 365 or other paid suites, users are granted access through **license assignment** either directly to their user account, or through a group using our group-based license assignment capability.
- For applications that Microsoft or a Third Party publishes freely for anyone to use, users may be granted access through **user consent**. This means that they sign in to the application with their Microsoft Entra work or school account and allow it to have access to some limited set of data on their account.
- For applications that Microsoft or a third-party publishes freely for anyone to use, users may also be granted access through **administrator consent**. This means that an administrator has determined the application may be used by everyone in the organization, so they sign in to the application with a Privileged Role Administrator account and grant access to everyone in the organization.

To troubleshoot your issue, start with the [General Problem Areas with Application Access to consider](#) and then read the Walkthrough: Steps to troubleshoot Microsoft Application access to get into the details.

General Problem Areas with Application Access to consider

Following is a list of the general problem areas that you can drill into if you have an idea of where to start, but we recommend you read the walkthrough to get going quickly:
Walkthrough: Steps to troubleshoot Microsoft Application access.

- [Problems with the user's account](#)
- [Problems with groups](#)
- [Problems with Conditional Access policies](#)

- Problems with application consent

Steps to troubleshoot Microsoft Application access

Following are some common issues folks run into when their users can't sign in to a Microsoft application.

- General issues to check first
 - Make sure the user is signing in to the **correct URL** and not a local application URL.
 - Make sure the user's account is **not locked out**.
 - Make sure the **user's account exists** in Microsoft Entra ID. [Check if a user account exists in Microsoft Entra ID](#)
 - Make sure the user's account is **enabled** for sign-ins. [Check a user's account status](#)
 - Make sure the user's **password is not expired or forgotten**. [Reset a user's password](#) or [Enable self-service password reset](#)
 - Make sure **multifactor authentication** isn't blocking user access. [Check a user's multifactor authentication status](#) or [Check a user's authentication contact info](#)
 - Make sure a **Conditional Access policy** or **legacy Identity Protection policy** isn't blocking user access. [Check a specific Conditional Access policy](#) or [Check a specific application's Conditional Access policy](#) or [Disable a specific Conditional Access policy](#)
 - Make sure that a user's **authentication contact info** is up to date to allow multifactor authentication or Conditional Access policies to be enforced. [Check a user's multifactor authentication status](#) or [Check a user's authentication contact info](#)
- For **Microsoft applications that require a license** (like Office365), here are some specific issues to check once you've ruled out the general issues above:
 - Ensure the user or has a **license assigned**. [Check a user's assigned licenses](#) or [Check a group's assigned licenses](#)
 - If the license is **assigned to a static group**, ensure that the **user is a member** of that group. [Check a user's group memberships](#)
 - If the license is **assigned to a dynamic group**, ensure that the **dynamic group rule is set correctly**. [Check a dynamic group's membership criteria](#)

- If the license is **assigned to a dynamic group**, ensure that the dynamic group has **finished processing** its membership and that the **user is a member** (this can take some time). [Check a user's group memberships](#)
- Once you make sure the license is assigned, make sure the license is **not expired**.
- Make sure the license is **for the application** they're accessing.
- For **Microsoft applications that don't require a license**, here are some other things to check:
 - If the application is requesting **user-level permissions** (for example "Access this user's mailbox"), make sure that the user has signed in to the application and has performed a **user-level consent operation** to let the application access their data.
 - If the application is requesting **administrator-level permissions** (for example "Access all user's mailboxes"), make sure that a Privileged Role Administrator has performed an **administrator-level consent operation on behalf of all users** in the organization.

Problems with the user's account

Application access can be blocked due to a problem with a user that is assigned to the application. Following are some ways you can troubleshoot and solve problems with users and their account settings:

- [Check if a user account exists in Microsoft Entra ID](#)
- [Check a user's account status](#)
- [Reset a user's password](#)
- [Enable self-service password reset](#)
- [Check a user's multifactor authentication status](#)
- [Check a user's authentication contact info](#)
- [Check a user's group memberships](#)
- [Check a user's assigned licenses](#)
- [Assign a user a license](#)

Check if a user account exists in Microsoft Entra ID

To check if a user's account is present, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#).
2. Browse to **Entra ID > Users**.
3. **Search** for the user you're interested in and select the row with the user's details.
4. Check the properties of the user object to be sure that they look as you expect and no data is missing.

Check a user's account status

To check a user's account status, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#).
2. Browse to **Entra ID > Users**.
3. **Search** for the user you're interested in and select the row with the user's details.
4. Select **Profile**.
5. Under **Settings** ensure that **Block sign in** is set to **No**.

Reset a user's password

To reset a user's password, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#).
2. Browse to **Entra ID > Users**.
3. **Search** for the user you're interested in and select the row with the user's details.
4. Select the **Reset password** button at the top of the user pane.
5. Select the **Reset password** button on the **Reset password** pane that appears.
6. Copy the **temporary password** or **enter a new password** for the user.
7. Communicate this new password to the user. They might be required to change this password during their next sign-in to Microsoft Entra ID.

Enable self-service password reset

To enable self-service password reset, follow the deployment steps in the following section:

- [Enable users to reset their Microsoft Entra passwords](#)
- [Enable users to reset or change their Active Directory on-premises passwords](#)

Check a user's multifactor authentication status

To check a user's multifactor authentication status, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#).
2. Browse to **Entra ID > Users**.
3. Select the **multifactor authentication** button at the top of the pane.
4. Once the **multifactor authentication Administration portal** loads, ensure you are on the **Users** tab.
5. Find the user in the list of users by searching, filtering, or sorting.
6. Select the user from the list of users and **Enable**, **Disable**, or **Enforce** multifactor authentication as desired.
 - **Note:** If a user is in an **Enforced** state, you may set them to **Disabled** temporarily to let them back into their account. Once they're back in, you can then change their state to **Enabled** again to require them to re-register their contact information during their next sign-in. Alternatively, you can follow the steps in the [Check a user's authentication contact info](#) to verify or set this data for them.

Check a user's authentication contact info

To check a user's authentication contact info used for multifactor authentication, Conditional Access, and Password Reset, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#).
2. Browse to **Entra ID > Users**.
3. **Search** for the user you're interested in and select the row with the user's details.
4. Select **Profile**.
5. Scroll down to **Authentication contact info**.
6. Review the data registered for the user and update as needed.

Check a user's group memberships

To check a user's group memberships, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#).
2. Browse to **Entra ID > Users**.

3. **Search** for the user you're interested in and select the row with the user's details.

4. Select **Groups** to see which groups the user is a member of.

Check a user's assigned licenses

To check a user's assigned licenses, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **user administrator**.

2. Browse to **Entra ID > Users**.

3. **Search** for the user you're interested in and select the row with the user's details.

4. Select **Licenses** to see which licenses the user currently has assigned.

Assign a user a license

To assign a license to a user, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **user administrator**.

2. Browse to **Entra ID > Users**.

3. **Search** for the user you're interested in and select the row with the user's details.

4. Select **Licenses** to see which licenses the user currently has assigned.

5. Select the **Assign** button.

6. Select **one or more products** from the list of available products.

7. **Optional** select the **assignment options** item to granularly assign products. Select **Ok** when this is completed.

8. Select the **Assign** button to assign these licenses to this user.

Problems with groups

Application access can be blocked due to a problem with a group that is assigned to the application. Following are some ways you can troubleshoot and solve problems with groups and group memberships:

- [Check a group's membership](#)
- [Check a dynamic group's membership criteria](#)

- Check a group's assigned licenses
- Reprocess a group's licenses
- Assign a group a license

Check a group's membership

To check a group's membership, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#) or [groups administrator](#).
2. Browse to **Entra ID > Groups > All Groups**.
3. Search for the group you're interested in and select the row with the group's details.
4. Select **Members** to review the list of users assigned to this group.

Check a dynamic group's membership criteria

To check a dynamic group's membership criteria, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#) or [groups administrator](#).
2. Browse to **Entra ID > Groups > All Groups**.
3. Search for the group you're interested in and select the row with the group's details.
4. Select **Dynamic membership rules**.
5. Review the **simple** or **advanced** rule defined for this group and ensure that the user you want to be a member of this group meets these criteria.

Check a group's assigned licenses

To check a group's assigned licenses, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#) or [groups administrator](#).
2. Browse to **Entra ID > Groups > All Groups**.
3. Search for the group you're interested in and select the row with the group's details.
4. Select **Licenses** to see which licenses the group currently has assigned.

Reprocess a group's licenses

To reprocess a group's assigned licenses, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#) or [groups administrator](#).
2. Browse to **Entra ID > Groups > All Groups**.
3. Search for the group you're interested in and select the row with the group's details.
4. Select **Licenses** to see which licenses the group currently has assigned.
5. Select the **Reprocess** button to ensure that the licenses assigned to this group's members are up-to-date. This may take a long time, depending on the size and complexity of the group.

ⓘ Note

To do this faster, consider temporarily assigning a license to the user directly. [Assign a user a license](#).

Assign a group a license

To assign a license to a group, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [user administrator](#) or [groups administrator](#).
2. Browse to **Entra ID > Groups > All Groups**.
3. Search for the group you're interested in and select the row with the group's details.
4. Select **Licenses** to see which licenses the group currently has assigned.
5. Select the **Assign** button.
6. Select **one or more products** from the list of available products.
7. **Optional** select the **assignment options** item to granularly assign products. Select **Ok** when this is completed.
8. Select the **Assign** button to assign these licenses to this group. This may take a long time, depending on the size and complexity of the group.

 **Note**

To do this faster, consider temporarily assigning a license to the user directly. [Assign a user a license](#).

Problems with Conditional Access policies

Check a specific Conditional Access policy

To check or validate a single Conditional Access policy:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Conditional Access Administrator](#).
2. Browse to **Entra ID > Enterprise apps**.
3. Select the **Conditional Access** navigation item.
4. Select the policy you're interested in inspecting.
5. Review that there are no specific conditions, assignments, or other settings that may be blocking user access.

 **Note**

You may wish to temporarily disable this policy to ensure it is not affecting sign-ins. To do this, set the **Enable policy** toggle to **No** and click the **Save** button.

Check a specific application's Conditional Access policy

To check or validate a single application's currently configured Conditional Access policy:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Entra ID > Enterprise apps > All applications**.
3. Search for the application you're interested in, or the user is attempting to sign in to by application display name or application ID.
4. Select the **Conditional Access** navigation item.

5. Select the policy you're interested in inspecting.
6. Review that there are no specific conditions, assignments, or other settings that may be blocking user access.

 **Note**

You may wish to temporarily disable this policy to ensure it is not affecting sign-ins. To do this, set the **Enable policy** toggle to **No** and click the **Save** button.

Disable a specific Conditional Access policy

To check or validate a single Conditional Access policy:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Conditional Access Administrator**.
2. Browse to **Entra ID > Enterprise apps**.
3. Select the **Conditional Access** navigation item.
4. Select the policy you're interested in inspecting.
5. Disable the policy by setting the **Enable policy** toggle to **No** and select the **Save** button.

Problems with application consent

Application access can be blocked because the proper permissions consent operation hasn't occurred. Following are some ways you can troubleshoot and solve application consent issues:

- [Perform a user-level consent operation](#)
- [Perform administrator-level consent operation for any application](#)
- [Perform administrator-level consent for a single-tenant application](#)
- [Perform administrator-level consent for a multitenant application](#)

Perform a user-level consent operation

- For any OpenID Connect-enabled application that requests permissions, navigating to the application's sign-in screen performs a user level consent to the application for the signed-in user.
- If you wish to do this programmatically, see [Requesting individual user consent](#).

Perform administrator-level consent operation for any application

- For **only applications developed using the V1 application model**, you can force this administrator level consent to occur by adding "?prompt=admin_consent" to the end of an application's sign-in URL.
- For **any application developed using the V2 application model**, you can enforce this administrator-level consent to occur by following the instructions under the **Request the permissions from a directory admin** section of [Using the admin consent endpoint](#).

Perform administrator-level consent for a single-tenant application

- For **single-tenant applications** that request permissions (like those you're developing or own in your organization), you can perform an **administrative-level consent** operation on behalf of all users by signing in as a Privileged Role Administrator and clicking on the **Grant permissions** button at the top of the **Application Registry -> All Applications -> Select an App -> Required Permissions** pane.
- For **any application developed using the V1 or V2 application model**, you can enforce this administrator-level consent to occur by following the instructions under the **Request the permissions from a directory admin** section of [Using the admin consent endpoint](#).

Perform administrator-level consent for a multitenant application

- For **multitenant applications** that request permissions (like an application a third party, or Microsoft, develops), you can perform an **administrative-level consent** operation. Sign in as a Privileged Role Administrator and select the **Grant permissions** button under the **Enterprise Applications -> All Applications -> Select an App -> Permissions** pane (available soon).
- You can also enforce this administrator-level consent to occur by following the instructions under the **Request the permissions from a directory admin** section of [Using the admin consent endpoint](#).

Next steps

[Using the admin consent endpoint](#)

Delete an enterprise application

Article • 03/06/2025

In this article, you learn how to delete an enterprise application that was added to your Microsoft Entra tenant.

When you delete an enterprise application, it remains in a suspended state in the recycle bin for 30 days. During the 30 days, you can [Restore the application](#). Deleted items are automatically hard deleted after the 30-day period. For more information on frequently asked questions about deletion and recovery of applications, see [Deleting and recovering applications FAQs](#).

Prerequisites

To delete an enterprise application, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free](#).
- One of the following roles:
 - Cloud Application Administrator
 - Application Administrator
 - owner of the service principal
- An [enterprise application added to your tenant](#).

Delete an enterprise application using Microsoft Entra admin center

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Cloud Application Administrator](#).
2. Browse to **Identity > Applications > Enterprise applications > All applications**.
3. Enter the name of the existing application in the search box, and then select the application from the search results. In this article, we use the **Microsoft Entra SAML Toolkit 1** as an example.
4. In the **Manage** section of the left menu, select **Properties**.
5. At the top of the **Properties** pane, select **Delete**, and then select **Yes** to confirm you want to delete the application from your Microsoft Entra tenant.

The screenshot shows the 'Properties' page for the 'Microsoft Entra SAML Toolkit 1' application in the Azure portal. The left sidebar includes sections like Home, Favorites, Identity, Overview, Users, Groups, Devices, Applications, Enterprise applications, App registrations, Protection, and Identity governance. The main content area shows the application's details: Name is 'Microsoft Entra SAML Toolkit 1', Homepage URL is 'https://www.microsoft.com/', and User access URL is '<user-access-URL>'. There are tabs for Overview, Deployment Plan, Diagnose and solve problems, and Manage. Under Manage, the 'Properties' tab is selected and highlighted with a red box. Other tabs include Owners, Roles and administrators, Users and groups, Single sign-on, Provisioning, Self-service, and Custom security attributes. Buttons at the top right include Save, Discard, Delete (which is highlighted with a red box), and Got feedback?.

Related content

- [Restore a deleted enterprise application](#)

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Restore a soft deleted enterprise application

Article • 03/06/2025

In this article, you learn how to restore a soft deleted enterprise application in your Microsoft Entra tenant. Soft deleted enterprise applications can be restored from the recycle bin within the first 30 days after their deletion. After the 30-day window, the enterprise application is permanently deleted and can't be restored.

If you deleted an [application registration](#) in its home tenant through app registrations in the Microsoft Entra admin center, the enterprise application, which is its corresponding service principal also got deleted.

If you restore the deleted application registration through the Microsoft Entra admin center, its corresponding service principal, is also restored. You therefore be able to recover the service principal's previous configurations, except its previous policies such as Conditional Access policies, which aren't restored.

Prerequisites

To restore an enterprise application, you need:

- A Microsoft Entra user account. If you don't already have one, you can [Create an account for free ↗](#).
- One of the following roles:
 - Cloud Application Administrator
 - Application Administrator
 - owner of the service principal.
- A [soft deleted enterprise application](#) in your tenant.

Take the following steps to recover a recently deleted enterprise application. For more information on frequently asked questions about deletion and recovery of applications, see [Deleting and recovering applications FAQs](#).

View restorable enterprise applications using Microsoft Entra PowerShell

Make sure you're using the [Microsoft Entra PowerShell](#) module.

You need to sign in as at least a [Cloud Application Administrator](#).

1. Run the following command to view the recently deleted enterprise application.

```
PowerShell
```

```
Connect-Entra -Scopes 'Application.Read.All'  
Get-EntraDeletedServicePrincipal
```

Replace ID with the object ID of the service principal that you want to restore.

Restore an enterprise app using Microsoft Entra PowerShell

1. To restore the soft-deleted enterprise application, run the following command:

```
PowerShell
```

```
Connect-Entra -Scopes 'Application.ReadWrite.All'  
#get the deleted service principal by filtering by the display name.  
$deletedServicePrincipal = Get-EntraDeletedServicePrincipal -Filter  
"DisplayName eq 'test-App1-Deleted'"  
  
#assign the value returned to a variable and restore the deleted  
service principal  
Id = $deletedServicePrincipal.Id  
Restore-EntraDeletedDirectoryObject -Id $deletedServicePrincipal.Id
```

Soft-deleted managed identity service principals can be viewed but can't be recovered or permanently deleted by customers.

Warning

Permanently deleting an enterprise application is an irreversible action. Any present configurations on the app are lost. Carefully review the details of the enterprise application to be sure you still want to hard delete it.

Permanently delete an enterprise app using Microsoft Entra PowerShell

To permanently delete a soft deleted enterprise application, run the following command:

```
PowerShell
```

```
Connect-Entra -Scopes 'Application.ReadWrite.All'  
#get the deleted service principal by filtering by the display name.  
$deletedServicePrincipal = Get-EntraDeletedServicePrincipal -Filter  
"DisplayName eq 'test-App1-Deleted'"  
  
#assign the value returned to a variable and permanently delete the  
service principal  
$Id = $deletedServicePrincipal.Id  
Remove-EntraDeletedDirectoryObject -Id $deletedServicePrincipal.Id
```

Related content

- [Recovery and deletion FAQ](#)
- [Applications and service principals](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) ↗

Manage Microsoft Entra applications and service principals by using Microsoft Graph

Article • 04/30/2025

This guide provides an overview of key concepts, API use cases, and resources to help you automate the lifecycle management of Microsoft Entra applications.

Applications and service principals

In Microsoft Entra, an application is defined by an [application object](#) and a [service principal object](#). There's only one application object for your application across Microsoft Entra, but there can be multiple service principal objects for your application.

The application object is located in the tenant where the app is registered. A service principal is created in the tenant where the app is registered, and in every tenant where it's installed and used. For more information, see [Application and service principal objects in Microsoft Entra ID](#).

In Microsoft Graph, an application is represented by the [application resource type](#), and a service principal is represented by the [servicePrincipal resource type](#). The details of the two objects can be accessed on the Microsoft Entra admin center through the [Entra ID > App registrations](#) and [Entra ID > Enterprise applications](#) menus respectively.

API use cases for managing applications

The following API use cases are supported for managing applications through the [application resource type](#) in Microsoft Graph.

[+] Expand table

Use cases	API operations
Register an application and configure its basic properties	Create application
Configure properties for a registered application including: <ul style="list-style-type: none">Basic properties such as display name, logo, and tagsPermissionsAssign apps to users	Update application

Use cases	API operations
<ul style="list-style-type: none"> Set the basic identifier URIs The Microsoft accounts that the app supports App roles 	
Delete an application	Delete application
Manage deleted applications	<ul style="list-style-type: none"> List deletedItems List deletedItems owners by a user Get deleted item Permanently delete item Restore deleted item
Manage password credentials for an application	<ul style="list-style-type: none"> application: addPassword application: removePassword
Manage federated identity credentials for an application	Start managing federated identity credentials using Microsoft Graph
Manage certificate-based credentials for an application	<ul style="list-style-type: none"> application: addKey application: removeKey Update the keyCredentials property through the update application API operation.
Manage directory extensions on applications	<ul style="list-style-type: none"> extensionProperty resource type and its associated methods. For more information, see Add custom data to resources using extensions.
Track changes to an application	<ul style="list-style-type: none"> application: delta directoryObject: delta with the following filter: <code>...? \$filter=isof('microsoft.graph.application')</code>
Manage owners	<ul style="list-style-type: none"> List owners Add owner Remove owner
Manage publisher verification	<ul style="list-style-type: none"> Set verifiedPublisher Unset verifiedPublisher

API use cases for managing service principals

The following API use cases are supported for managing service principals through the [servicePrincipal resource type](#) in Microsoft Graph.

 [Expand table](#)

Use cases	API operations
Register service principal	Create servicePrincipal
Configure properties for a service principal including: - Basic properties such as display name and logo - Permissions - Configure SSO mode	Update servicePrincipal
Delete a service principal	Delete servicePrincipal
Manage deleted service principals: view, restore, or permanently delete	<ul style="list-style-type: none">- List deletedItems- List deletedItems owned by a user- Get deleted item- Permanently delete item- Restore deleted item
Manage password credentials for a service principal	<ul style="list-style-type: none">- servicePrincipal: addPassword- servicePrincipal: removePassword
Manage certificate-based credentials for a service principal	<ul style="list-style-type: none">- servicePrincipal: addKey- servicePrincipal: removeKey
Add a SAML token signing certificate	servicePrincipal: addTokenSigningCertificate
Track changes to a service principal	<ul style="list-style-type: none">- servicePrincipal: delta- directoryObject: delta with the following filter: <code>...? \$filter=isof('microsoft.graph.servicePrincipal')</code>
Manage owners	<ul style="list-style-type: none">- List owners- Add owner- Remove owner

Application templates

Application templates are apps available in the [Microsoft Entra app gallery](#). Use the `applicationTemplate` resource type and its associated methods to:

- Identify apps from the application gallery.
- Identify apps by the SSO mode they support.
- Instantiate an app and service principal from an application gallery.

Policies applicable to applications and service principals

[] Expand table

Policy description	API operations	Applies to
Manage Microsoft Entra ID Remote Desktop Services (RDS) authentication protocol	remoteDesktopSecurityConfiguration resource type and its associated methods	Service principals
Configure SAML tokens policy	tokenIssuancePolicy resource type and its associated methods	Applications Service principals
Configure policies for access, SAML, and ID tokens	Token lifetime policy - tokenLifetimePolicy resource type and its associated methods Token issuance policy - tokenIssuancePolicy resource type and its associated methods	Applications Service principals
Manage idle session time-out for Microsoft 365 web apps, for all device types Note: To trigger the policy only for unmanaged devices, you also need to add a Conditional Access policy.	activityBasedTimeoutPolicy resource type and its associated methods	Microsoft 365 web apps
Manage policies for how certificates and password secrets can be used in your organization. Create tenant-wide policies or app-specific policies such as blocking the use of or restricting the lifetime of password secrets or symmetric keys and enforcing trusted certificate authorities	Application authentication methods policies	Applications
Manage claims mapping policies for WS-Fed, SAML, OAuth 2.0, and OpenID Connect	claimsMappingPolicy resource type and its associated methods	Service principals

Policy description	API operations	Applies to
protocols, and the applications the policies apply to		
Manage Home Realm Discovery (HRD) for the tenant and assignment of the policy to a service principal	homeRealmDiscoveryPolicy resource type and its associated methods	Service principals

Identity synchronization (provisioning)

Provisioning APIs in Microsoft Graph let you automate and manage the provisioning and deprovisioning of identities in these scenarios:

- From your on-premises Active Directory to Microsoft Entra ID
- From other cloud directories to Microsoft Entra ID
- From Microsoft Entra ID to cloud applications like Dropbox, Salesforce, ServiceNow, and more

For more information, see [Microsoft Entra synchronization API overview](#).

Related content

- [Quick reference guide: API operations for managing applications](#)
- [Application management in Microsoft Entra ID](#)
- [Tutorials for integrating applications with Microsoft Entra ID](#)
- [What is the Microsoft identity platform?](#)

Manage a Microsoft Entra application using Microsoft Graph

Article • 12/17/2024

Your app must be registered in Microsoft Entra ID before the Microsoft identity platform can authorize it to access data stored in the Microsoft cloud. This condition applies to apps that you develop yourself, that your tenant owns, or that you access through an active subscription.

Many settings for apps are recorded as objects that can be accessed, updated, or deleted using Microsoft Graph. In this article, you learn how to use Microsoft Graph to manage details of app and service principal objects including the properties, permissions, and role assignments.

Prerequisites

To test the API operations, you need the following resources and privileges:

- A working Microsoft Entra tenant.
- Sign in to [Graph Explorer](#) as a user with privileges allowed to create and manage applications in the tenant.
- Grant yourself the least privilege delegated permission indicated for the operation.

Register an application with Microsoft Entra ID

The following request creates an app by specifying only the required `displayName` property. Other properties are assigned the default values.

Least privileged delegated permission: `Application.ReadWrite.All`.

```
HTTP

msgraph

POST https://graph.microsoft.com/v1.0/applications
Content-type: application/json

{
    "displayName": "My application"
}
```

The request returns a `201 Created` response with the application object in the response body. The application is assigned an `id` that's unique for apps in the tenant, and an `appId` that's globally unique in the Microsoft Entra ID ecosystem.

Create a service principal for an application

Least privileged delegated permission: `Application.ReadWrite.All`.

```
HTTP
HTTP
POST https://graph.microsoft.com/v1.0/servicePrincipals
Content-type: application/json

{
  "appId": "fc876dd1-6bcb-4304-b9b6-18ddf1526b62"
}
```

The request returns a `201 Created` response with the service principal object in the response body.

Addressing an application or a service principal object

You can address an application or a service principal by its ID or by its `appId`, where ID is referred to as *Object ID* and `appId` is referred to as *Application (client) ID* on the Microsoft Entra admin center. These syntaxes are supported for all HTTP CRUD operations on applications and service principals.

To address an application or a service principal by its ID.

```
HTTP
https://graph.microsoft.com/v1.0/applications/{applicationObjectId}
https://graph.microsoft.com/v1.0/servicePrincipals/{servicePrincipalObjectId}
```

To address an application or a service principal by its `appId`.

```
HTTP
```

```
https://graph.microsoft.com/v1.0/applications(appId='appId')
https://graph.microsoft.com/v1.0/servicePrincipals(appId='appId')
```

In addition, you can address an application object unique its **uniqueName**. You can use this property to create an application with the unique name if it doesn't exist, or update it if it exists; an operation referred to as "Upsert".

Create an application with the specified uniqueName if it doesn't exist, otherwise, update it.

HTTP

PATCH

```
https://graph.microsoft.com/v1.0/applications(uniqueName='{uniqueName}')
Content-Type: application/json
Prefer: create-if-missing
```

```
{
  "displayName": "Display name"
}
```

Configure other basic properties for your app

Least privileged delegated permission: `Application.ReadWrite.All`.

You configure the following basic properties for the app.

- Add tags for categorization in the organization. Also, use the `HideApp` tag to hide the app from My Apps and the Microsoft 365 Launcher.
- Add basic information including the logo, terms of service, and privacy statement.
- Store contact information about the application

HTTP

HTTP

```
PATCH https://graph.microsoft.com/v1.0/applications/0d0021e2-eaab-4b9f-
a5ad-38c55337d63e/
Content-type: application/json
```

```
{
  "tags": [
    "HR",
    "Payroll",
    "HideApp"
}
```

```
],
  "info": {
    "logoUrl": "https://cdn.pixabay.com/photo/2016/03/21/23/25/link-1271843_1280.png",
    "marketingUrl": "https://www.contoso.com/app/marketing",
    "privacyStatementUrl": "https://www.contoso.com/app/privacy",
    "supportUrl": "https://www.contoso.com/app/support",
    "termsOfServiceUrl":
      "https://www.contoso.com/app/termsofservice"
  },
  "web": {
    "homePageUrl": "https://www.contoso.com/",
    "logoutUrl": "https://www.contoso.com/frontchannel_logout",
    "redirectUris": [
      "https://localhost"
    ]
  },
  "serviceManagementReference": "Owners aliases: Finance @ contosofinance@contoso.com; The Phone Company HR consulting @ hronsite@thephone-company.com;"
}
```

Limit app sign-in to only assigned identities

The following operation limits the identities that can sign in to an app to only those that are assigned all roles on the app.

Least privileged delegated permission: `Application.ReadWrite.All`.

HTTP

HTTP

```
PATCH https://graph.microsoft.com/v1.0/servicePrincipals/89473e09-0737-41a1-a0c3-1418d6908bcd

{
  "appRoleAssignmentRequired": true
}
```

Assign permissions to an app

While you can assign permissions to an app through the Microsoft Entra admin center, you also assign permissions through Microsoft Graph by updating the `requiredResourceAccess` property of the app object. You must pass in both existing and

new permissions. Passing in only new permissions overwrites and removes the existing permissions that haven't yet been consented to.

Assigning permissions doesn't automatically grant them to the app. You must still grant admin consent using the Microsoft Entra admin center. To grant permissions without interactive consent, see [Grant or revoke API permissions programmatically](#).

Least privileged delegated permission: `Application.ReadWrite.All`.

```
HTTP

HTTP

PATCH https://graph.microsoft.com/v1.0/applications/581088ba-83c5-4975-
b8af-11d2d7a76e98
Content-Type: application/json

{
  "requiredResourceAccess": [
    {
      "resourceAppId": "00000002-0000-0000-c000-000000000000",
      "resourceAccess": [
        {
          "id": "311a71cc-e848-46a1-bdf8-97ff7156d8e6",
          "type": "Scope"
        },
        {
          "id": "3afa6a7d-9b1a-42eb-948e-1650a849e176",
          "type": "Role"
        }
      ]
    }
  ]
}
```

Create app roles

Create app roles on an application object

To keep any existing app roles, include them in the request. Otherwise, they're replaced with the new object.

```
HTTP

HTTP
```

```
PATCH https://graph.microsoft.com/v1.0/applications/bbd46130-e957-4c38-a116-d4d02af1057
Content-Type: application/json

{
    "appRoles": [
        {
            "allowedMemberTypes": [
                "User",
                "Application"
            ],
            "description": "Survey.Read",
            "displayName": "Survey.Read",
            "id": "7a9ddfc4-cc8a-48ea-8275-8ecbffffd5a0",
            "isEnabled": false,
            "origin": "Application",
            "value": "Survey.Read"
        }
    ]
}
```

Manage owners

Identify ownerless service principals and service principals with one owner

Least privileged delegated permission: Application.ReadWrite.All.

This request requires the **ConsistencyLevel** header set to `eventual` because `$count` is in the request. For more information about the use of **ConsistencyLevel** and `$count`, see [Advanced query capabilities on directory objects](#).

This request also returns the count of the apps that match the filter condition.

HTTP

msgraph

```
GET https://graph.microsoft.com/v1.0/servicePrincipals?
$filter=owners/$count eq 0 or owners/$count eq 1&$count=true
ConsistencyLevel: eventual
```

Assign an owner to an app

Least privileged delegated permission: `Application.ReadWrite.All`.

In the following request, `8afc02cb-4d62-4dba-b536-9f6d73e9be26` is the object ID for a user or service principal.

```
HTTP

HTTP

POST https://graph.microsoft.com/v1.0/applications/7b45cf6d-9083-4eb2-92c4-a7e090f1fc40/owners/$ref
Content-Type: application/json

{
    "@odata.id":
    "https://graph.microsoft.com/v1.0/directoryObjects/8afc02cb-4d62-4dba-b536-9f6d73e9be26"
}
```

Assign an owner to a service principal

Least privileged delegated permission: `Application.ReadWrite.All`.

The following request references the service principal using its `appId`. You can alternatively reference it using the object ID in the pattern `.../servicePrincipals/{objectId}/owners/$ref`. `8afc02cb-4d62-4dba-b536-9f6d73e9be26` is the object ID for a user or service principal.

```
HTTP

HTTP

POST https://graph.microsoft.com/v1.0/servicePrincipals(appId='46e6adf4-a9cf-4b60-9390-0ba6fb00bf6b')/owners/$ref
Content-Type: application/json

{
    "@odata.id":
    "https://graph.microsoft.com/v1.0/directoryObjects/8afc02cb-4d62-4dba-b536-9f6d73e9be26"
}
```

Lock sensitive properties for service principals

The *app instance lock* feature allows you to protect sensitive properties of your multitenant apps from unauthorized tampering. The following properties of the service principal object can be locked:

- **keyCredentials** where the usage type is `Sign` or `Verify`.
- **passwordCredentials** where the usage type is `Sign` or `Verify`.
- **tokenEncryptionKeyId** property.

You manage the app instance lock feature through the **servicePrincipalLockConfiguration** property of the [application](#) object of the multitenant app.

To lock all sensitive properties of a service principal

When **isEnabled** and **allProperties** is set to `true`, even if other properties of the **servicePrincipalLockConfiguration** object are `null`, then all sensitive properties of the service principal are locked.

```
HTTP  
  
HTTP  
  
PATCH https://graph.microsoft.com/beta/applications/a0b7f39e-3139-48aa-9397-f46fb63102f7  
  
{  
    "servicePrincipalLockConfiguration": {  
        "isEnabled": true,  
        "allProperties": true  
    }  
}
```

To lock specific sensitive properties of a service principal

The following example locks the **keyCredentials** and **passwordCredentials** properties of the service principal and enables the app instance lock feature.

```
HTTP  
  
HTTP  
  
PATCH https://graph.microsoft.com/beta/applications/a0b7f39e-3139-48aa-9397-f46fb63102f7
```

```
{  
    "servicePrincipalLockConfiguration": {  
        "isEnabled": true,  
        "credentialsWithUsageSign": true,  
        "credentialsWithUsageVerify": true  
    }  
}
```

Configure trusted certificate authorities for apps

You can restrict certificate credential usage for apps in your tenant to only the certificates issued by trusted certificate authorities. This policy is enforced when you add a certificate to an app and doesn't affect existing certificates unless they are rotated. When an app tries to rotate its certificate credentials, it goes through the policy evaluation to ensure the credentials being added comply with the trusted certificate authority restriction.

Step 1: Create a certificate chain of trust

Least privileged delegated permission: `AppCertTrustConfiguration.Read.All` Least privileged Microsoft Entra role: `Application Administrator`

HTTP

HTTP

POST
`https://graph.microsoft.com/beta/certificateAuthorities/certificateBasedApplicationConfigurations`

```
{  
    "displayName": "Trusted Certificate Chain of Trust for Contoso",  
    "description": "The Trusted Certificate Chain of Trust containing a certificate chain used by app policy, to only allow application certificates from selected issuer.",  
    "trustedCertificateAuthorities": [  
        {  
            "isRootAuthority": true,  
            "certificate": "MIIFVjCCAz6gAwIBAgIQJdrL...UyNDIyNTcwM1owPDE  
...="  
        },  
        {  
            "isRootAuthority": false,  
            "certificate": "MIIFVjCCAz6gAwIBAgIQJdrL...UyNDIyNTcwM1owPDE  
...="  
        }  
    ]  
}
```

```
        "certificate": "QAAAAAAWjABAQsFADA8M...UyNDIyNTcwM1o ...="  
    }  
}  
]
```

The request returns a `200 OK` response object. The response includes the ID of the certificate chain of trust object. Assume that the ID is `eec5ba11-2fc0-4113-83a2-ed986ed13743`.

Step 2: Assign the certificate chain of trust to an application management policy

The following sample sets up a policy to ensure that only certificates issued by the intermediate certificate authority defined in the previous step can be added to apps in the tenant. The **applicationRestrictions > keyCredentials** object defines a **restrictionType** with the value `trustedCertificateAuthority`, which references the ID that was created. Because this policy is applied to the default tenant-level app management policy, it's enforced for all apps created in the tenant and rejects attempts to add noncompliant certificates as part of an app's certificate credentials.

This policy ensures that only certificates from the specified intermediate certificate authority can be added to apps. The **applicationRestrictions > keyCredentials** object sets a **restrictionType** to `trustedCertificateAuthority`, referencing the created ID. This policy applies to all apps in the tenant, rejecting any noncompliant certificates.

Least privileged delegated permission: `Policy.Read.ApplicationConfiguration` Least privileged Microsoft Entra role: `Security Administrator`

The screenshot shows the Microsoft Graph API Explorer interface. At the top, there is a navigation bar with tabs for 'HTTP' and 'Graph'. Below the navigation bar, the URL is set to 'PATCH https://graph.microsoft.com/v1.0/policies/defaultAppManagementPolicy'. The request body contains the following JSON payload:

```
{  
  "id": "d015220e-9789-4e8e-bbcc-270fe419229d",  
  "description": "Lorem ipsum",  
  "displayName": "Credential management policy",  
  "isEnabled": true,  
  "applicationRestrictions": {  
    "passwordCredentials": [  
      {  
        "restrictionType": "passwordLifetime",  
        "value": 10  
      }  
    ]  
  }  
}
```

```
        "maxLifetime": "P14D",
        "restrictForAppsCreatedAfterDateTime": "2020-01-01T07:00:00Z"
    }
],
"keyCredentials": [
{
    "restrictionType": "certificateLifetime",
    "restrictForAppsCreatedAfterDateTime": "2020-01-01T10:37:00Z",
    "maxLifetime": "P90D"
},
{
    "restrictionType": "trustedCertificateAuthority",
    "certificateBasedApplicationConfigurationIds": [
        "eec5ba11-2fc0-4113-83a2-ed986ed13743"
    ],
    "restrictForAppsCreatedAfterDateTime": "2019-10-19T10:37:00Z"
}
]
}
```

Related content

- [The Microsoft Entra app manifest](#)
- [Properties of an enterprise application \(service principal\)](#)
- [Add a certificate to an app using Microsoft Graph](#)

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback

Deletion and recovery of applications

FAQ

FAQ

The following are some frequently asked questions (FAQs) on deletion and recovery of applications.

When I create applications, and get a Directory_QuotaExceeded error, how can I avoid this problem?

A nonadmin user can create no more than 250 Microsoft Entra resources that include applications and service principals. Both active resources and deleted resources that are available to restore count toward this quota. Even if you delete more applications that you don't need, they still add count to the quota. To free up the quota, you need to [permanently delete](#) objects in the deleted items container.

For more information about the service limits, see [Azure resource management](#).

Where can I find all the deleted applications and service principals?

Soft-deleted application and service principal objects go into the deleted items container and remain available to restore for up to 30 days. After 30 days, they're permanently deleted, thus freeing up the quota.

To learn how to view deleted application objects through the Microsoft Entra admin center, see [View restorable applications](#).

Deleted service principals can't be viewed through the Microsoft Entra admin center. To learn how to view your restorable service principals using PowerShell or Microsoft Graph API, see [View restorable service principals](#).

How do I restore deleted applications or service principals?

To learn how to restore recently deleted application registrations through the Microsoft Entra admin center, see [Restore application registrations](#). If the application registration and its corresponding service principal got deleted, the service principal is also restored.

To learn how to restore recently deleted service principals, see [Restore service principals](#). This method is also applicable for restoring recently deleted application registrations using PowerShell or Microsoft Graph API.

How do I permanently delete soft deleted applications or service principals?

To permanently delete application registrations through the Microsoft Entra admin center, see [Permanently delete an application](#).

To permanently delete a service principal, see [Permanently delete a service principal](#). This method is also applicable for permanently deleting application registrations using PowerShell or Microsoft Graph API.

Can I configure the interval in which applications and service principals are permanently deleted by Microsoft Entra ID?

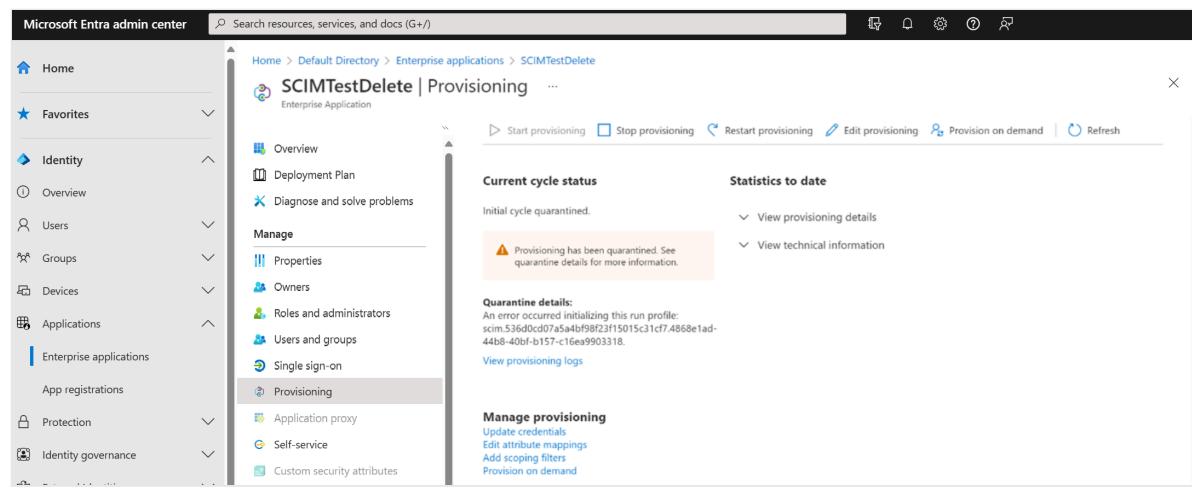
No. You can't configure the periodicity of hard deletion.

Are managed identities soft-deleted?

Yes, Managed identities are soft-deleted. You can view the soft-deleted managed identity service principal from the recycle bin within 30 days after deletion, but you can't restore or permanently delete it. The managed identity service principal is permanently deleted after 30 days. For more information on how to view soft-deleted managed identities service principals, see [View deleted service principals](#).

I can't see the provisioning data from a recovered service principal. How can I recover it?

After recovering a service principal, you may initially see the error in the following screenshot. This issue resolves itself between 40 mins and 1 day. If you'd like the provisioning job to start immediately, you can hit restart to force the provisioning service to run again. Hitting restart triggers an initial cycle that can take time for customers with 100 K+ users or group memberships.



I recovered my application that was configured for application proxy. I can't see app proxy configurations after the recovery. How can I recover it back?

App proxy configurations can't be recovered through the portal UI. Use the API to recover app proxy settings. Expect a delay of up to 24 hours as the app proxy data gets synced back.

I can't see the policies I set on the service principal object after the recovery. How can I recover them?

Policies can't be recovered currently. When you restore a service principal, you have to configure the policies again.

Next steps

- [Restore a service principal](#)
 - [Restore an application registration](#)
-

Feedback

Was this page helpful?



Yes



No

[Provide product feedback](#) ↗

Microsoft Entra admin consent workflow frequently asked questions

Article • 10/23/2023

I enabled a workflow, but when testing the functionality, why can't I see the new "Approval required" prompt that allows me to request access?

After enabling the feature, it may take up to 60 minutes for users to see the update, though it's usually available to all users within a few minutes.

As a reviewer, why can't I see all pending requests?

Reviewers can only see admin requests that are created after they're designated as a reviewer. If you've recently been added as a reviewer, you won't see requests that were created before your assignment.

As a reviewer, why do I see multiple requests for the same application?

If an application is configured to use static and dynamic consent to request access to their user's data, you'll see two admin consent requests. One request represents the static permissions, and the other represents the dynamic permissions.

As a requestor, can I check the status of my request?

No, requestors are only able to receive updates using email notifications.

As a reviewer, is it possible to approve the application, but not for everyone?

If you're concerned about granting admin consent and allowing all users in the tenant to use the application, you should deny the request. You can then manually grant admin consent by restricting access to the application. Configure the application to require user assignment, and assign users or groups to the application to restrict access. For more information, see [Methods for assigning users and groups](#).

I have an application that requires user assignment. A user that I assigned to an application is being asked to request admin consent instead of being able to consent themselves. Why is that?

When access to an application is restricted using the "user assignment required" setting, an administrator needs to consent to all the permissions requested by the application.

Application Management certificates frequently asked questions

Article • 05/17/2024

This page answers frequently asked questions about managing the certificates for apps using Microsoft Entra ID as an Identity Provider (IdP).

Is there a way to generate a list of expiring SAML signing certificates?

You can export all app registrations with expiring secrets, certificates, and their owners for the specified apps from your directory in a CSV file through [PowerShell scripts](#).

Where can I find the information about soon to expire certificates renewal steps?

You can find the steps [here](#).

How can I customize the expiration date for the certificates issued by Microsoft Entra ID?

By default, Microsoft Entra ID configures a certificate to expire after three years when it's created automatically during SAML single sign-on configuration. Because you can't change the date of a certificate after you save it, you need to create a new certificate. For steps on how to do so, refer [Customize the expiration date for your federation certificate and roll it over to a new certificate](#).

Note

The recommended way to create SAML applications is through the Microsoft Entra Application Gallery, which will automatically create a three-year valid X509 certificate for you.

How can I automate the certificates expiration notifications?

Microsoft Entra ID sends an email notification 60, 30, and 7 days before the SAML certificate expires. You might add more than one email address to receive notifications.

 **Note**

You can add up to 5 email addresses to the Notification list (including the email address of the admin who added the application). If you need more people to be notified, use the distribution list emails.

To specify the emails you want the notifications to be sent to, see [Add email notification addresses for certificate expiration](#).

There's no option to edit or customize these email notifications received from `aadnotification@microsoft.com`. However, you can export app registrations with expiring secrets and certificates through [PowerShell scripts](#).

Who can update the certificates?

The owner of the application or Application Administrator can update the certificates through Microsoft Entra admin center UI, PowerShell, or Microsoft Graph.

I need more details about certificate signing options

In Microsoft Entra ID, you can set up certificate signing options and the certificate signing algorithm. To learn more, see [Advanced SAML token certificate signing options for Microsoft Entra apps](#).

What type of certificate can I use for configuring the SAML Certificate for single sign-on?

The recommendation for the SAML single sign-on certificate depends on your organization's security requirements and policies. If your organization has an internal certificate authority (PKI), using a certificate from the internal PKI can provide a higher level of security and trust. This is because the internal PKI is under the control of your organization and can be managed and monitored to ensure the security of the certificate.

On the other hand, if your organization doesn't have an internal certificate authority, using a certificate from an external certificate authority such as DigiCert can provide a higher level of trust and security. This is because external certificate authorities are trusted by many organizations and are subject to strict security and validation requirements.

I need to replace the certificate for Microsoft Entra application proxy applications and need more instructions

To replace certificates for Microsoft Entra application proxy applications, see [PowerShell sample - Replace certificate in Application Proxy apps](#).

How do I manage certificates for custom domains in Microsoft Entra application proxy?

To configure an on-premises app to use a custom domain, you need a verified Microsoft Entra custom domain, a PFX certificate for the custom domain, and an on-premises app to configure. To learn more, see [Custom domains in Microsoft Entra application proxy](#).

I need to update the token signing certificate on the application side. Where can I get it on Microsoft Entra ID side?

You can renew a SAML X.509 Certificate [SAML Signing certificate](#).

What is Microsoft Entra ID signing key rollover?

You can find more details [here](#).

How do I renew application token encryption certificate?

To renew an application token encryption certificate, see [How to renew a token encryption certificate for an enterprise application](#).

How do I renew application token signing certificate?

To renew an application token signing certificate, see [How to renew a token signing certificate for an enterprise application](#).

How do I update Microsoft Entra ID after changing my federation certificates?

To update Microsoft Entra ID after changing your federation certificates, see [Renew federation certificates for Microsoft 365 and Microsoft Entra ID](#).

Can I use the same SAML certificate across different apps?

When it's the first time configuring SSO on an enterprise app, we do provide a default SAML certificate that is used across Microsoft Entra ID. However, if you need to use the same certificate across multiple apps that aren't the default Microsoft Entra ones, use an external Certificate Authority and upload the PFX file. The reason is that Microsoft Entra ID doesn't provide access to private keys from internally issued certificates.

Applications listed in Enterprise applications

Article • 10/23/2023

The [Quickstart Series on Application Management](#) walks you the basics. In it, you learn how to view all of the apps using your Microsoft Entra tenant for identity management. This article dives a bit deeper into the types of apps you'll find.

Why does a specific application appear in my all applications list?

When filtered to **All Applications**, the **All Applications List** shows every Service Principal object in your tenant. Service Principal objects can appear in this list in a various ways:

- When you add any application from the application gallery, including:
 - **Microsoft Entra ID - Enterprise applications** – Apps added to your tenant using the **Enterprise applications** option on the Microsoft Entra admin center. Usually apps integrated using the SAML standard.
 - **Microsoft Entra ID - App registrations** – Apps added to your tenant using the **App registrations** option on the Microsoft Entra admin center. Usually custom developed apps using the OpenID Connect and OAuth standards.
 - **Application Proxy Applications** – An application running in your on-premises environment that you want to provide secure single-sign on to externally
- When signing up for, or signing in to, a third-party application integrated with Microsoft Entra ID. One example is [Smartsheet](#) or [DocuSign](#).
- Microsoft apps such as Microsoft 365.
- When you use managed identities for Azure resources. For more information, see [Managed identity types](#).
- When you add a new application registration by creating a custom-developed application using the [Application Registry](#)
- When you add a new application registration by creating a custom-developed application using the [V2.0 Application Registration portal](#)
- When you add an application, you're developing using Visual Studio's [ASP.NET Authentication Methods](#) or [Connected Services](#)

- When you create a service principal object using the [Microsoft Graph PowerShell](#) module.
- When you [consent to an application](#) as an administrator to use data in your tenant
- When a [user consents to an application](#) to use data in your tenant
- When you enable certain services that store data in your tenant. One example is Password Reset, which is modeled as a service principal to store your password reset policy securely.

Learn more about how, and why, apps are added to your directory, see [How applications are added to Microsoft Entra ID](#).

Next steps

[Managing Applications with Microsoft Entra ID](#)

Understand how users are assigned to apps

Article • 10/23/2023

This article helps you to understand how users get assigned to an application in your tenant.

How do users get assigned an application in Microsoft Entra ID?

There are several ways a user can be assigned an application. Assignment can be performed by an administrator, a business delegate, or sometimes, the user themselves. Below describes the ways users can get assigned to applications:

- An administrator [assigns a user](#) to the application directly
- An administrator [assigns a group](#) that the user is a member of to the application, including:
 - A group that was synchronized from on-premises
 - A static security group created in the cloud
 - A [dynamic security group](#) created in the cloud
 - A Microsoft 365 group created in the cloud
 - The [All Users](#) group
- An administrator enables [Self-service Application Access](#) to allow a user to add an application using [My Apps](#) Add App feature **without business approval**
- An administrator enables [Self-service Application Access](#) to allow a user to add an application using [My Apps](#) Add App feature, but only **with prior approval from a selected set of business approvers**
- An administrator enables [Self-service Group Management](#) to allow a user to join a group that an application is assigned to **without business approval**
- An administrator enables [Self-service Group Management](#) to allow a user to join a group that an application is assigned to, but only **with prior approval from a selected set of business approvers**
- One of the application's roles is included in an [entitlement management access package](#), and a user requests or is assigned to that access package

- An administrator assigns a license to a user directly, for a Microsoft service such as [Microsoft 365](#)
- An administrator assigns a license to a group that the user is a member of, for a Microsoft service.
- A user [consents to an application](#) on behalf of themselves.

Next steps

- [Quickstart Series on Application Management](#)
 - [What is application management?](#)
 - [What is single sign-on?](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) | [Get help at Microsoft Q&A](#)

Remove user access to applications

Article • 10/23/2023

This article provides several scenarios for removing user access to applications in Microsoft Entra ID.

Scenarios

Remove a specific user's or group's assignment to an application

To remove a user or group assignment to an application, follow the instructions in [Remove a user or group assignment from an enterprise app in Microsoft Entra ID](#).

Disable all user access to an application

To disable all user sign-ins to an application, follow the instructions in [Disable user sign-ins for an enterprise app in Microsoft Entra ID](#).

Delete an application entirely

To delete an application from your Microsoft Entra tenant, follow the guidance in the [Quickstart Series on Application Management](#).

Disable all future user consent operations in any application

To disable all future user consent operations in an application, follow the instructions in [Configure how end-users consent to applications](#).

Next steps

[Manage access to apps](#)

Resources for migrating applications to Microsoft Entra ID

Article • 05/31/2024

Resources to help you migrate application access and authentication to Microsoft Entra ID.

[] Expand table

Resource	Description
Migrating your apps to Microsoft Entra ID	This article is an introduction to a series of articles that describe how to plan for migration in four clearly-outlined phases: discovery, classification, migration, and ongoing management. You're guided through how to think about the process and break down your project into easy-to-consume pieces. Throughout the series are links to important resources that help you along the way.
Developer tutorial: AD FS to Microsoft Entra application migration playbook for developers	This set of ASP.NET code samples and accompanying tutorials help you learn how to safely and securely migrate your applications integrated with Active Directory Federation Services (AD FS) to Microsoft Entra ID. This tutorial is focused towards developers who not only need to learn how to configure apps on both AD FS and Microsoft Entra ID, but also become aware and confident of changes their code base will require in this process.
Tool: Active Directory Federation Services Migration Readiness Script	This is a script you can run on your on-premises Active Directory Federation Services (AD FS) server to determine the readiness of apps for migration to Microsoft Entra ID.
Deployment plan: Migrating from AD FS to password hash sync	With password hash synchronization, hashes of user passwords are synchronized from on-premises Active Directory to Microsoft Entra ID. This allows Microsoft Entra ID to authenticate users without interacting with the on-premises Active Directory.
Deployment plan: Migrating from AD FS to pass-through authentication	Microsoft Entra pass-through authentication helps users sign in to both on-premises and cloud-based applications by using the same password. This feature provides your users with a better experience since they have one less password to remember. It also reduces IT help desk costs because users are less likely to forget how to sign in when they only need to remember one password. When people sign in using Microsoft Entra ID, this feature validates users' passwords directly against your on-premises Active Directory.
Deployment plan: Enabling single sign-on	Single sign-on (SSO) helps you access all the apps and resources you need to do business, while signing in only once, using a single user

Resource	Description
to a SaaS app with Microsoft Entra ID	account. For example, after a user has signed in, the user can move from Microsoft Office, to SalesForce, to Box without authenticating (for example, typing a password) a second time.
Deployment plan: Extending apps to Microsoft Entra ID with Application Proxy	Providing access from employee laptops and other devices to on-premises applications has traditionally involved virtual private networks (VPNs) or demilitarized zones (DMZs). Not only are these solutions complex and hard to make secure, but they're costly to set up and manage. Microsoft Entra application proxy makes it easier to access on-premises applications.
Other deployment plans	Find more deployment plans for deploying features such as Microsoft Entra multifactor authentication, Conditional Access, user provisioning, seamless SSO, self-service password reset, and more!
Migrating apps from Symantec SiteMinder to Microsoft Entra ID	Get step by step guidance on application migration and integration options with an example that walks you through migrating applications from Symantec SiteMinder to Microsoft Entra ID.
Identity governance for applications	This guide outlines what you need to do if you're migrating identity governance for an application from a previous identity governance technology, to connect Microsoft Entra ID to that application.
Active Directory Federation Services (AD FS) decommission guide	This guide explains the prerequisites for decommissioning, including migrating user authentication and applications to Microsoft Entra ID. It also provides step-by-step instructions for decommissioning the AD FS servers, including removing load balancer entries, uninstalling WAP and AD FS servers, and deleting SSL certificates and databases.
Phases of migrating apps from ADFS to Microsoft Entra ID	This set of articles illustrates the five phases of a typical migration of an application from ADFS to Microsoft Entra ID.
Migrate identity management scenarios from SAP IDM to Microsoft Entra	If you've been using SAP Identity Management (IDM), then you can migrate identity management scenarios from SAP IDM to Microsoft Entra.
Migrating Identity and Access Management scenarios to Microsoft Entra from Microsoft Identity Manager	This document provides guidance on migration options and approaches for moving Identity and Access Management (IAM) scenarios from Microsoft Identity Manager to Microsoft Entra cloud-hosted services.

Support and help options for developers

Article • 05/15/2024

If you need an answer to a question or help in solving a problem not covered in our documentation, it might be time to reach out to experts for help. Here are several suggestions for getting answers to your questions as you develop applications that integrate with the Microsoft identity platform.

Create an Azure support request



Explore the range of [Azure support options and choose the plan ↗](#) that best fits you. The following options for creating and managing support requests are available in the Microsoft Entra admin center:

- If you already have an Azure Support Plan, [open a support request here ↗](#).
- If you're using Microsoft Entra External ID in an external tenant, the support request feature is currently unavailable for external tenant technical issues. However, you can use the **Give Feedback** link on the **New support request** page to provide feedback. Or, you can switch to your Microsoft Entra workforce tenant and [open a support request ↗](#).
- If you're not an Azure customer, you can open a support request with [Microsoft Support for business ↗](#).

Post a question to Microsoft Q&A



Get answers to your identity app development questions directly from Microsoft engineers, Azure Most Valuable Professionals (MVPs), and members of our expert community.

[Microsoft Q&A](#) is Azure's recommended source of community support.

If you can't find an answer to your problem by searching Microsoft Q&A, submit a new question. Use one of following tags when you ask your [high-quality question](#):

[+] Expand table

Component/area	Tags
Microsoft Entra External ID / External Identities	Microsoft Entra External ID ↗
Microsoft Entra B2B / External Identities	Microsoft Entra External ID
Azure AD B2C	Microsoft Entra External ID
All other Microsoft Entra areas	Microsoft Entra ID
Azure RBAC	Azure Role-Based access control
Azure Key Vault	Azure Key Vault
Microsoft Security	Microsoft Defender for Cloud
Microsoft Sentinel	Microsoft Sentinel
Microsoft Entra Domain Services	Microsoft Entra Domain Services
Azure Windows and Linux Virtual Machines	Azure Virtual Machines

Create a GitHub issue



If you need help with one of the Microsoft Authentication Libraries (MSAL), open an issue in its repository on GitHub.

[+] Expand table

MSAL	GitHub issues URL
MSAL for Android	https://github.com/AzureAD/microsoft-authentication-library-for-android/issues ↗
MSAL Angular	https://github.com/AzureAD/microsoft-authentication-library-for-js/issues ↗
MSAL for iOS and macOS	https://github.com/AzureAD/microsoft-authentication-library-for-objc/issues ↗
MSAL Java	https://github.com/AzureAD/microsoft-authentication-library-for-java/issues ↗
MSAL.js	https://github.com/AzureAD/microsoft-authentication-library-for-js/issues ↗

MSAL	GitHub issues URL
MSAL.NET	https://github.com/AzureAD/microsoft-authentication-library-for-dotnet/issues ↗
MSAL Node	https://github.com/AzureAD/microsoft-authentication-library-for-js/issues ↗
MSAL Python	https://github.com/AzureAD/microsoft-authentication-library-for-python/issues ↗
MSAL React	https://github.com/AzureAD/microsoft-authentication-library-for-js/issues ↗

Stay informed of updates and new releases



- [Azure Updates](#) : Learn about important product updates, roadmap, and announcements.
- [What's new in docs](#): Get to know what's new in the Microsoft identity platform documentation.
- [Microsoft Entra Blog](#) : Get news and information about Microsoft Entra ID.
- [Tech Community](#): Share your experiences, engage, and learn from experts.

Share your product ideas

Have an idea for improving the Microsoft identity platform? Browse and vote for ideas submitted by others or submit your own:

[https://feedback.azure.com/d365community/forum/22920db1-ad25-ec11-b6e6-000d3a4f0789 ↗](https://feedback.azure.com/d365community/forum/22920db1-ad25-ec11-b6e6-000d3a4f0789)