

Abstract

This project plan covers the deployment requirements, guidance, plans, and communication necessary to deploy <Insert App Name> to <Insert Company Name> as part of the Modern IT initiative being undertaken by the IAM Business Group.

Azure Active Directory Project Deployment Guide

**<Insert App Name> Single Sign-on Configuration**

<Insert Author Name>

Table of Contents

Contents

[How to use these documents 3](#_Toc504984826)

[Introduction 1](#_Toc504984828)

[Purpose of Document 1](#_Toc504984829)

[Confidentiality Statement 1](#_Toc504984830)

[Selected Solutions 1](#_Toc504984831)

[Azure Active Directory Single Sign-on? 1](#_Toc504984832)

[Azure Active Directory Provisioning 1](#_Toc504984833)

[Azure Active Directory Conditional Access 1](#_Toc504984834)

[Current State 1](#_Toc504984835)

[Current Authentication Methods 1](#_Toc504984836)

[Costs of Current Authentication 2](#_Toc504984837)

[Benefits of Azure Active Directory Single Sign-On for «APPLICATIONNAME» 3](#_Toc504984838)

[Planning Your Implementation 2](#_Toc504984839)

[2](#_Toc504984840)

[General Planning 2](#_Toc504984841)

[Solution Components 2](#_Toc504984842)

[In Scope 2](#_Toc504984843)

[Out of scope 4](#_Toc504984844)

[Licensing 4](#_Toc504984845)

[Planning Single Sign on 5](#_Toc504984846)

[Considerations for Password-based Single Sign on 6](#_Toc504984847)

[Planning Provisioning 7](#_Toc504984848)

[Planning Conditional Access and Application Security 7](#_Toc504984849)

[Network Definition 7](#_Toc504984850)

[Multi-Factor Authentication 7](#_Toc504984851)

[Device State 8](#_Toc504984852)

[Planning Reporting and Auditing 8](#_Toc504984853)

[Security Considerations 9](#_Toc504984854)

[Design 10](#_Toc504984855)

[Suggested Capabilities 10](#_Toc504984856)

[Recommended Device Access Policies 10](#_Toc504984857)

[Recommended Capacity and Performance Capabilities 10](#_Toc504984858)

[Recommended Disaster Recovery Capabilities 11](#_Toc504984859)

[Recommended Security Capabilities 11](#_Toc504984860)

[Recommended User Experience Capabilities 11](#_Toc504984861)

[Recommended Auditing and Governance Recommendations 12](#_Toc504984862)

[Capabilities Used 12](#_Toc504984863)

[Solution Architecture Diagram and Description 14](#_Toc504984864)

[Technical Requirements 14](#_Toc504984865)

[Established Timelines 17](#_Toc504984866)

[Implement 18](#_Toc504984867)

[Change Communications 18](#_Toc504984868)

[Implementation Steps 18](#_Toc504984869)

[Test Plan 22](#_Toc504984870)

[Rollback steps 24](#_Toc504984871)

[Operational Doc 24](#_Toc504984872)

[Purpose of Document 24](#_Toc504984873)

[Confidentiality Statement 24](#_Toc504984874)

[Required Roles 24](#_Toc504984875)

[Supportability 25](#_Toc504984876)

[Support Case Scenarios 25](#_Toc504984877)

[Troubleshooting Guide & Steps 25](#_Toc504984878)

[Expected SLA 27](#_Toc504984879)

[Auditing 27](#_Toc504984880)

[Single Sign-on Certificate Lifecycle [Azure Active Directory] 28](#_Toc504984881)

[Access Management 28](#_Toc504984882)

[Documentation 29](#_Toc504984883)

# How to use these documents

This document is based upon following your selections:

User provisioning



Federated single sign on



Would you like to automate the creation, management, and removal of user accounts within <<ApplicationName>> through AAD provisioning?

Would you like to enable users to seamlessly access <<ApplicationName>> using a single account?

Conditional access



Would you like to improve the security of your application through conditional access policies (e.g. MFA, trusted devices, etc.)?

There are three documents included here:

* [**Business case** You can modify this document to use with your business stakeholders.](#businesscase)
* [**Implementation guide** This will guide you through the implementation.](#implement)
* [**Operational guide**](#_Operational_Doc) This will guide you through the operating and supporting your application.

Each document addresses common information for your selections, such as overall planning, as well as specific information such as planning federated SSO.

AZURE ACTIVE DIRECTORY

IMPLEMENTATION GUIDE

BUSINESS CASE FOR <<APPLICATION NAME>> INTEGRATION

# Introduction

## Purpose of Document

This document presents an executive summary of the business case for moving forward with enabling Azure Active Directory Single Sign-on (SSO) and associated security and provisioning capabilities for application (“The Application”).

## Confidentiality Statement

This document may provide certain information that is and must be kept confidential. To ensure the protection of such information you should not disclose any part of this plan summary to anyone who does not have a legitimate reason for .

## Selected Solutions

Many organizations rely upon software as a service (SaaS) applications such as Office 365, Box and Salesforce for end user productivity. Historically, IT staff needs to individually create and update user accounts in each SaaS application, and users must remember a password for each SaaS application.

Azure Active Directory extends on-premises Active Directory into the cloud, enabling users to use their primary organizational account to not only sign in to their domain-joined devices and company resources, but also all web and SaaS applications needed for their job.

### Azure Active Directory Single Sign-on

Single sign-on means being able to access all the applications and resources that a user needs to do business, by signing in only once using a single user account. Once signed in, the user can access all needed applications without being required to authenticate (e.g. type a password) a second time.

### Azure Active Directory Provisioning

In addition to single sign-on applications, access can be automatically provisioned or de-provisioned based on their organization group members, and their status as employees

### Azure Active Directory Conditional Access

Azure Active Directory introduces security and access governance controls that enable you to centrally manage users' access across SaaS applications.

# Current State

<<this is an optional section in which you can detail your current state to help your stakeholders and decision makers understand the benefits of moving to SSO specific to your enterprise. >>

<< Insert your summary text here. Eg: By moving to single sign-on with this application, we will save XX dollars in people costs, and reduce necessary infrastructure and its associated costs by xxx>>

## Current Authentication Methods

<<In this section, detail the current authentication method, and the infrastructure that supports it.>

## Costs of Current Authentication

<< In this section, detail the infrastructure and people costs that support the current infrastructure, and which parts of that cost can be eliminated by moving to SSO. Include support costs related to sign on issues, for example, the number of password resets or lockouts, and what those cost both from a support and a time loss perspective. You may also want to include the costs of manual provisioning that could be eliminated.

## Benefits of Azure Active Directory Single Sign-On for

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | \\MAGNUM\Projects\Microsoft\Cloud Power FY12\Design\ICONS_PNG\Increase.png | **INCREASE PRODUCTIVITY**  Enabling **single sign-on** and **provisioning** across multiple enterprise applications alongside Office 365 provides a superior log in experience for existing users, reducing or eliminating log on prompts. The user’s environment feels more cohesive and is less distracting without multiple prompts, or the need to manage multiple passwords. Access control can be managed and approved by the business group, saving IT management costs through self-service and dynamic membership, and improving the overall security of our identity system by ensuring the right people in the business manage access to this application. | | \\MAGNUM\Projects\Microsoft\Cloud Power FY12\Design\ICONS_PNG\Confidentiality.png | **MANAGE RISK**  Enabling **conditional access** policies can offer significantly improved security experiences. These include cloud-scale identity protection, risk-based access control capabilities, native multi-factor authentication support, and conditional access policies which allow for granular control policies based on applications, or on groups that need higher levels of security. | | C:\Users\mitchellg\Desktop\Simple_Licensing.png | **ADDRESS COMPLIANCE AND GOVERNANCE**  Auditing access requests and approvals for the application, as well as understanding overall application usage, becomes easier with Azure Active Directory, which supports native audit logs for every application access request performed. Auditing includes requester identity, requested date, business justification, approval status, and approver identity. This data is also available from an API, which will enables importing this data into a Security Incident and Event Monitoring (SIEM) system of choice. | | \\MAGNUM\Projects\Microsoft\Cloud Power FY12\Design\ICONS_PNG\Within_Your_Reach.png | **MANAGE COST**  The replacement of the current access management and provisioning process and migration to Azure Active Directory to manage self-service access to the application (as well as other SaaS applications in the future) will allow for significant cost reductions related to running, managing, and maintaining our current infrastructure. Additionally, eliminating application specific passwords eliminates costs related to password reset for that application, and lost productivity while retrieving passwords. | |  |

AZURE ACTIVE DIRECTORY

IMPLEMENTATION GUIDE

<<APPLICATION NAME>>

How to use this guide

This step-by-step guide walks you through deploying and securing your application in a five-step process. The links below take you to each of those steps.



**1**

**Include**

Stakeholders

**2**

**Plan**

Your project

**3**

**Design**

Policies and integration

**5**

**Manage**

Your implementation



**4**

**Implement**

Your design

Within sections two, three, and four, it is further broken down by the elements that you chose.

Below, find links for the planning, design, and implementation for each of your selections.

|  |  |  |
| --- | --- | --- |
| User provisioning | Federated single sign on | Conditional access |
| Plan user provisioning | Plan federated SSO | Plan access policies |
| Design user provisioning | Design Federated SSO | Design access policies |
| Implement user provisioning | Implement Federated SSO | Implement access policies |

# Planning Your Implementation

## 

**Note:**

Throughout this section, you will see items marked as

* **Microsoft Recommends**

These are general recommendations, and you should only implement if they apply to your specific enterprise needs.

## General Planning

### In Scope

The following is in scope for this project:



**Federated single sign on**

* Enabling single sign-on to the application using Microsoft Azure Active Directory federation technologies.
* Extension of on-premises AD to include new attributes which will be provisioned to the Azure AD or application environments.

**User provisioning**



* Enabling end-to-end user provisioning to the application, so that a user created in our organizational Active Directory infrastructure is also automatically synchronized to the application as a new account, in accordance with business policies.
* Enable the Azure Active Directory Connect infrastructure required to synchronize data from our corporate Active Directory system into Microsoft Azure Active Directory.
* Enable the business policy rules required in Microsoft Azure Active Directory to ensure that newly-created users have application access.
* Enable us to leverage existing user attributes from corporate Active Directory in Azure Active Directory for new accounts within the application.



**Conditional access**

* Securing access to the application by leveraging the appropriate Microsoft Azure Active Directory security technologies, for example:[[1]](#footnote-2)
  + Multi-factor authentication access for all application administrative roles.
  + Multi-factor authentication access for all users accessing the application from outside the organizational intranet.
  + Multi-factor authentication access for all users accessing the application from any device which has not been previously registered with the organization.
  + Risk-based Conditional Access policies against all Azure Active Directory sign-ins to ensure the application, and any other future Azure Active Directory federated applications, is accessed securely.

**For all changes**

* Enabling the support organization to support and manage this new change, ensuring the right helpdesk processes are in place to ensure on-going end-user success.
* Documenting and testing a recovery plan.
* Approving a business continuity plan.
* Approving an information security risk assessment.
* Designing operational support for the production service.
* The following environments are in scope for this design:
  + Production
  + Test / QA

### Out of scope

The following are out of scope of this project:

* Enabling any other application for federated single-sign or provisioning.
* Extending the corporate Active Directory system with any additional or new attributes that are require by the application. Any new attributes necessary will be created in Azure Active Directory.
* Disabling of the existing federation relationship between the application and our corporate federation solution.

### Licensing

#### Azure Active Directory Licensing

You will need an [Azure AD License.](https://azure.microsoft.com/en-us/pricing/details/active-directory/) The number of objects in your directory and the features you wish to deploy will affect your licensing choices. While many features are included with Azure Free and Azure Basic, some features require Azure AD Premium (P1 or P2). Common Azure AD scenarios include the following recommended security features:

* [Conditional Access (CA)](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-conditional-access-azure-portal) (P1 Required)
* [Azure Multi-Factor Authentication (MFA) (P1 Required)](https://docs.microsoft.com/en-us/azure/multi-factor-authentication/multi-factor-authentication-how-it-works)
* Group based membership (P1 required)
* [Identity Protection](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-identityprotection) (P2 Required)

The following table describes some of the license requirements that may be relevant. For a full list of license requirements, click [here](https://azure.microsoft.com/en-us/pricing/details/active-directory/).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Free | BASIC | PREMIUM P1 | PREMIUM P2 |
| Directory Objects | 500,000 Object Limit | No Object Limit | No Object Limit | |
| Single Sign-On | 10 apps per user (pre-integrated SaaS and developer-integrated apps) | 10 apps per user (free tier + Application proxy apps) | No Limit (free, Basic tiers + Self-Service App Integration templates) | |
| Access based on group membership | Not Available | | Available | |
| CA based on group and location | Not available | | Available | |
| CA based on device state (Allow access from managed devices) | Not available | | Available | |
| MFA (cloud and on-premises) | Not available | | Available | |
| Identity protection | Not available | | Not available | Available |

If you have an existing Enterprise Mobility and Security (EMS) subscription with Microsoft, you may already have Azure Premium.

**Enterprise Mobility and Security (EMS) subscriptions:**

* EMS E3 includes P1
* EMS E5 includes P2.

If you have an existing Enterprise Agreement or Server and Cloud Enrollment, you may already have Azure Premium. Check the details of your agreement.

#### Application Licensing

You will also need the appropriate license for your application to meet your business needs.

Discuss with the application owner whether the users assigned to and accessing the application have the appropriate licenses for their roles within the application. If Azure AD manages the automatic provisioning based on roles, the roles that are assigned in Azure AD must align with the correct number of licenses owned within the application; improper number of licenses owned in the application may lead to errors during the provisioning/updating of a user.

## Planning Single Sign on

An SSO implementation based on federation protocols improves security, reliability, and end user experiences while reducing the amount of work you must do to implement. Many applications are pre-integrated in Azure AD with step-by step guides. See our Azure Marketplace.

* **Federated single sign-on** enables applications to redirect to Azure AD for user authentication instead of prompting for its own password. This is supported for applications that support protocols such as SAML 2.0, WS-Federation, or OpenID Connect, and is the richest mode of single sign-on.
* **Password-based single sign-on** enables secure application password storage and replay using a web browser extension or mobile app. This leverages the existing sign-in process provided by the application, and enables an administrator to manage the passwords and does not require the user to know the password.
* **Microsoft recommends** using Federated SSO over password-based SSO.

When you enable Federated SSO for your application, Azure AD creates a certificate that is by default valid for three years. You can, however, customize the expiration date for that certificate if desired.

* Learn more: [Azure AD Managing Certificates](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-sso-certs)

Whatever your expiration policies, ensure that you have processes in place to renew certificates prior to their expiration.

**Planning for External Users**

External users are users who you want to access your instance of the SaaS application, but who do not have corporate identities within your organization. To facilitate their access, you will need to plan business-to-business access. You can use [Azure AD B2B for this](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-b2b-what-is-azure-ad-b2b).

### Considerations for Password-based Single Sign on

**Preparing user devices for password SSO applications**   
Using Azure AD for password SSO applications requires deploying a browser extension that will securely retrieve the credentials and fill out the login forms. As a result, you should define a mechanism to deploy the extension at scale. Options include:

* [Group Policy for Internet Explorer](https://azure.microsoft.com/en-us/documentation/articles/active-directory-saas-ie-group-policy/)
* [System Center Configuration Manager (SCCM) for Internet Explorer](https://docs.microsoft.com/en-us/sccm/" \l "pivot=tasks)
* User driven download and configuration for Chrome or Firefox

Learn more: [How to configure password single sign on](https://docs.microsoft.com/en-us/azure/active-directory/application-config-sso-how-to-configure-password-sso-non-gallery)

**Capturing Login forms metadata for applications that are not in the gallery**   
Work with the application owner to find out the exact login URL. You have to navigate to the login URL during the process of configuring the application to capture the forms metadata.

Learn more: [What is application access and SSO with Azure AD? – Password-based SSO](https://azure.microsoft.com/en-us/documentation/articles/active-directory-appssoaccess-whatis/)

**User training**   
Users need to access the password SSO from the myapps portal or the office portal for the browser extension to supply the password.

Learn more: End user experiences

**Help desk training**   
When applications change their HTML layout, you might need to re-capture the metadata to adjust for the changes. Common symptoms to watch for:

* Users report that clicking on the application gets “stuck” in the login page
* Users report that username is populated, but not the password or vice versa.

**Shared Accounts**  
From the sign-in perspective, applications with Shared Accounts are not different from a gallery application that uses password SSO for individual users. However, there are some additional steps when planning and configuring an application meant to use shared accounts:

1. Work with application business users to get a specific mapping of:
2. Set of users in the organization who will use the application
3. Existing set of credentials in the application associated with the set of users
4. For each combination determined above, create security groups (in the cloud or on-premises based on your requirements) with target actors.
5. Reset the shared credentials. Once deployed with Azure AD, individuals do not (and should not) need the password of the shared account. Since Azure AD will store the password, consider setting it to be very long and complex.
6. Configure automatic rollover of the password if the application supports it. That way, not even the administrator who did the initial set up will know the password of the shared account.

## Planning Provisioning

Azure Active Directory (Azure AD) can automatically provision users and groups to any application or identity store that is fronted by a web service with the interface defined in the System for Cross-Domain Identity Management (SCIM) 2.0 protocol specification. Azure Active Directory can send requests to create, modify, or delete assigned users and groups to the web service. The web service can then translate those requests into operations on the target identity store.

Some of the applications in the gallery have pre-configured connectors that you can use to enable provisioning.

If they do not, you can configure the connector yourself.

* Learn more: [Using System for Cross-Domain Identity Management to automatically provision users and groups from Azure AD to applications](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-scim-provisioning)

## Planning Conditional Access and Application Security

Azure AD provides options to secure access to applications, and you must determine which you will use. These include:

* Determining if users can access the application from within and outside the internal network.
* Whether to use multi-factor authentication (MFA), and if so, how.
* Which devices may access the application, and under what circumstances.

We recommend that you design a set of policies based on these criteria and frame the conversation with the business owners in terms of the risks associated with access.

Additionally, you may want to use Tenant Restrictions to ensure that only your users can access your instances of the SaaS application.

* **Learn More**: [Tenant Restrictions](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-tenant-restrictions)

Azure AD will continue to evolve and refine access control.

Network Definition

You must define your internal network. This is done in one of two ways.

* If you will use password hash sync, or cloud only identities, you must define your network by providing Azure AD with the list of IP addresses that constitute the internal network.   
  Learn more: [Trusted IPs for federated users](https://azure.microsoft.com/en-us/documentation/articles/multi-factor-authentication-get-started-adfs-cloud/).
* If you are not using password hash sync or cloud only identities, an on-premises identity provider (Such as AD FS) can provide a claim that indicates the network location.
* **Microsoft recommends** using IP addresses to define the network.

Multi-Factor Authentication  
There are times when you will want to increase application security by asking users to provide additional authentication of their identity. Azure AD Multi-factor Authentication (MFA) enables you to collect this additional authentication, typically via a device such as a phone, for the following patterns:

1. For a given application:
   1. all users, at all times
   2. all users when accessing from external networks
2. For a given user or users:
   1. all apps, at all times
   2. all apps, when accessing from outside the network
   3. Only on logins identified as high risk

\**Note: Risk-based MFA is available with* [*Azure AD Identity Protection*](https://azure.microsoft.com/en-us/documentation/articles/active-directory-identityprotection/)*. Additional licensing might be required. Consult with your License Specialist before deploying in production*.

Device State  
You must determine if you will allow corporate devices to connect to the application from both inside and outside the corporate network. And, you must decide if you will allow personal devices to connect at all, and if so if you will allow them only inside the network, or also allow them outside the network.

* **Microsoft Recommends:**
  + Allowing both secure personal and enterprise devices from inside the network.
  + Allowing only devices registered with Azure AD from outside the network.

## Planning Reporting and Auditing

Azure AD provides reports that provide technical and business insights. It is recommended that you work with your business and technical application owners to assume ownership of and consume these reports on a regular basis based on your organization’s requirements. The table below provides some examples of typical reporting scenarios.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Manage Costs |  | Manage Risk |  | Increase Productivity |  | Governance & Compliance |
|  |  |  |  |  |  |  |  |
| Report types |  |  | Application permissions and usage. |  | Account provisioning activity |  |  |
| Potential actions | Budget reconciliation |  | Audit access; revoke permissions |  | Remediate any provisioning errors |  | Document compliance |

Azure AD retains most auditing data for 30 days, and makes the data available via an API for you to download into your analysis systems.

Learn more: [View your access and usage reports](https://azure.microsoft.com/en-us/documentation/articles/active-directory-view-access-usage-reports/)

### Planning Your Security Review

The following describes security considerations related to this change:

If a security review is required for Azure AD using Single Sign-on, please review to the following documentation:

* [Overview whitepapers for Azure AD](https://www.microsoft.com/en-us/download/details.aspx?id=36391)

# 

## Established Timelines

### Project stages (Pilot Prod, etc)

This table provides the timelines for addressing the scenarios outlined.

|  |  |  |
| --- | --- | --- |
| Scenarios | Timelines | Scope |
|  | | |
| POC – sandbox environment - Config |  |  |
| POC – sandbox environment - Testing |  |  |
| PROD – Configure |  |  |
| PROD – Testing |  |  |

# Design

This section is used to design the business capabilities that you want to enableYou should add capabilities necessary for your environment. The capabilities necessary can then from the basis for your test planning. . When there are choices among options, and Microsoft has a clear recommendation, it is indicated.

Within each table below, indicate in the required column the business capabilities that you want to include in your design. This will form the basis for your implementation

## Designing Single Sign on

Indicate which capabilities are necessary for your SSO solution.

|  |  |  |
| --- | --- | --- |
| SSO Capability | Description | Required |
| Implement federated single sign on | * Microsoft Recommended   Use SAML or OIDC capabilities to access your application |  |
| Implement password based SSO |  |  |

## Designing Provisioning

## Designing Conditional Access

### Network Access Definition

|  |  |  |
| --- | --- | --- |
| Network Definition Type | Description | Required |
| Define network based on IP ranges. | * **Microsoft Recommended**   Define the internal network based on IP ranges. You must use this if you are using password hash sync or pass-through authentication, or cloud-only identities. |  |
| Define network based on an on-premises identity system claim | Define the inside network based on a claim by the on premises identity system, such as AD FS. |  |

### Recommended Device Access Policies

All of the following are recommended. Add your additional requirements.

|  |  |  |
| --- | --- | --- |
| Device and Network Status | Description | Required |
| Corporate devices  inside the corporate network | Application must be accessible from PC’s, Macs, Android, iOS, and Windows Phone devices that are registered with Azure Active Directory, from within the corporate network. |  |
| Personal devices  inside the corporate network | Application must be accessible from PC’s, Macs, Android, iOS, and Windows Phone devices that are not registered with Azure Active Directory, from within the corporate network. |  |
| Corporate devices  outside the corporate network | Application must be accessible from PC’s, Macs, Android, iOS, and Windows Phone devices that are registered with Azure Active Directory, from outside of the corporate network. |  |
| Personal devices  outside the corporate network | Application must **not** be accessible from outside the corporate network unless a user’s device has been registered with the Azure Active Directory. |  |

## Designing Governance and Auditing

All of the following are recommended. Add your additional requirements.

|  |  |  |
| --- | --- | --- |
| Recommendation | Description | Required |
| Audit/Monitoring | Reporting data to measure the services provided e.g., performance, availability, number of successful connections. |  |
| Governance / Attestation | Requires process to manage Lifecycle of user assignment to application |  |

## Designing Security and Disaster Recovery

All of the following are recommended. Add your additional requirements.

|  |  |  |
| --- | --- | --- |
| Security Capability | Description | Required |
| Control access via user/group assignment | It will not be possible for users with access to gain access to any other services other than Application unless they are explicitly assigned those services. |  |
| SIEM integration | The solution must be integrated to the existing Security Incident and Event monitoring tool system using an API |  |
| Monitoring-Audit logs | Audit logs are available as required covering all transactions in Azure AD. |  |

### Designing Disaster Recovery Capabilities

All of the following are recommended. Add your additional requirements.

|  |  |  |
| --- | --- | --- |
| Disaster Recovery Capability | Description | Required |
| Fail over process | In the event of a failure of the SSO integration, ensure users can continue to access the application. |  |
| Disaster recovery - planning | Microsoft provides fully tested disaster recovery and failover plans. |  |
| Disaster recovery - testing | Disaster recovery testing shall be done on a minimally annual basis to ensure continuity of access to <Insert App Name>. |  |

### Designing the End User Experience

All of the following are recommended. Add your additional requirements.

|  |  |  |
| --- | --- | --- |
| Experience | Description | Required |
| Account Provisioning | As soon as an account is provisioned to <Insert App Name>, there is no delay to users being able to access the application. |  |
| Browser Compatibility Information | Browser compatibility information must be available and there must be the ability to publish to users. |  |
| Log on | Users are able to log on and authenticate to <Insert App Name> using existing credentials from corporate owned devices. |  |
| Log on | Users are able to log on and authenticate to <Insert App Name> using existing credentials from personally owned devices. |  |
| Log on | Reduce the number of prompts when accessing <Insert App Name>. |  |
| Log on | Users are able to log on seamlessly with <Insert App Name> Lightning for Outlook Add-on |  |
| Log on | Browser and Mobile Client support |  |

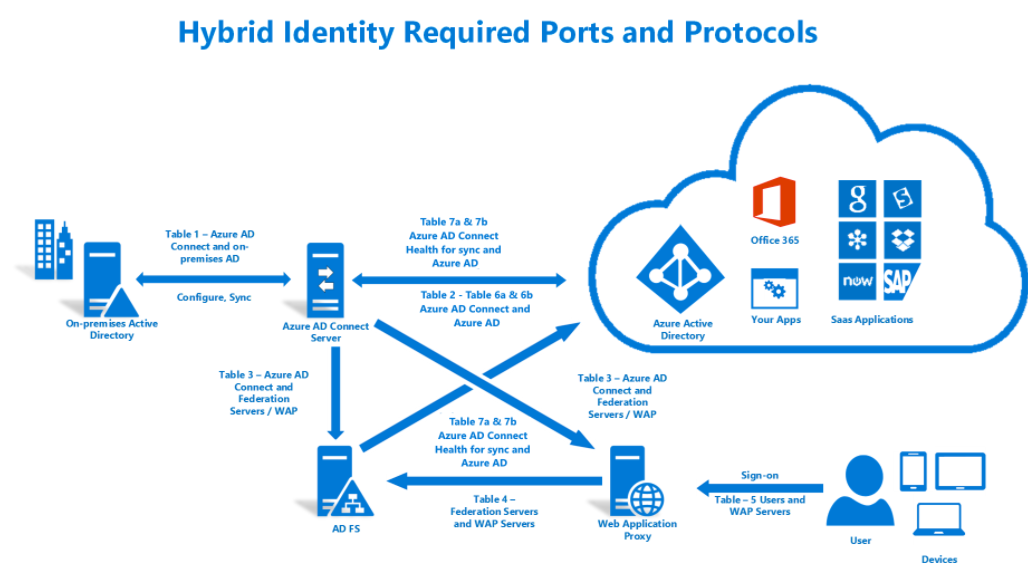
## Solution Architecture Diagrams and Description

Several topologies are represented below. You should choose the one that most closely matches your specific scenario, and delete the rest.

### Azure AD Single Sign-on

This solution is a combination of hybrid identity sync using Azure AD Connect while maintaining a trust with on-premises AD FS as the Identity provider for Azure Active Directory. Azure Active Directory acts as the IdP for <Insert App Name> while applying Conditional Access policies on the servicePrincipal object. User and Group objects are maintained by continued synchronization between on-premises Active Directory and Azure Active Directory to ensure continuity and maintained Lifecycle for devices, users, and groups.

In case of an emergency, an administrator must be able to access the Azure Active Directory infrastructure from a non-registered device from off of the corporate network, but only if that administrator is accessing the application using multi-factor authentication.

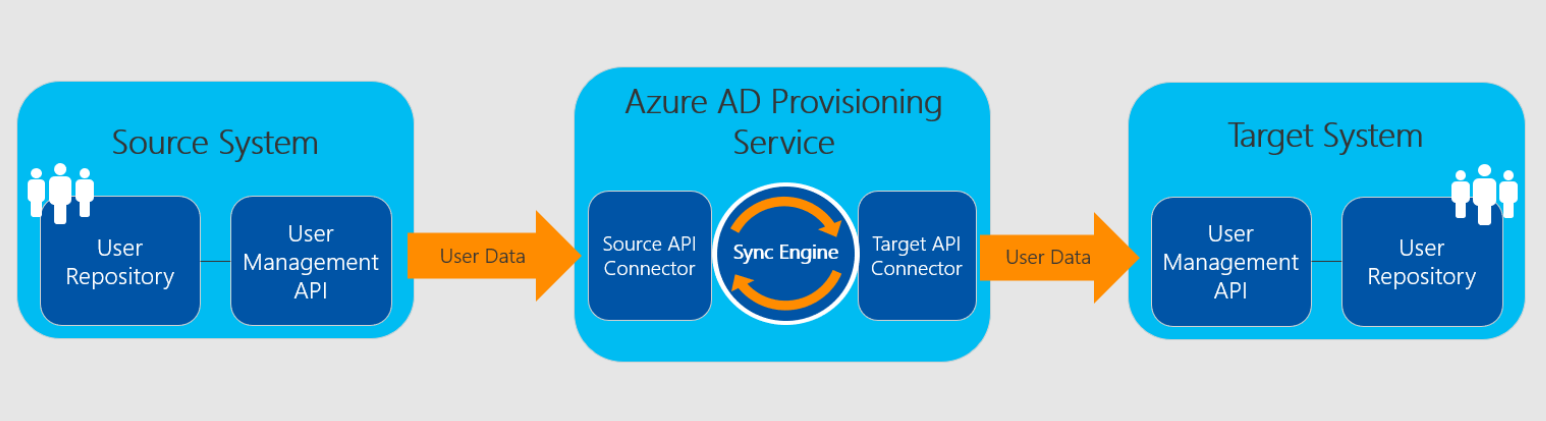


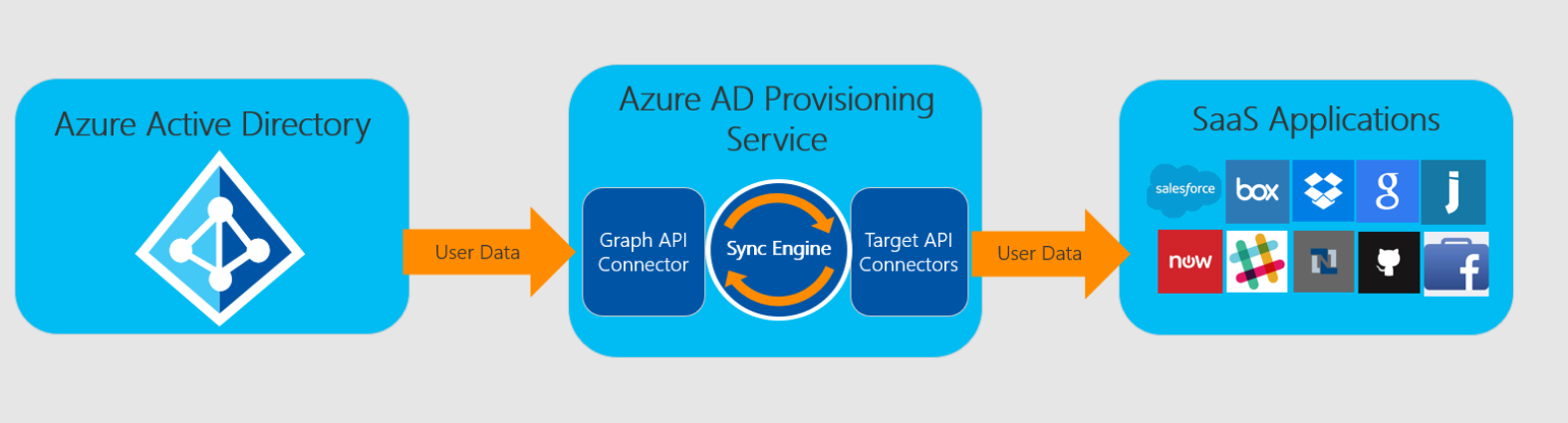
[Communication, ports, and protocols reference](https://docs.microsoft.com/en-us/azure/active-directory/connect/active-directory-aadconnect-ports)

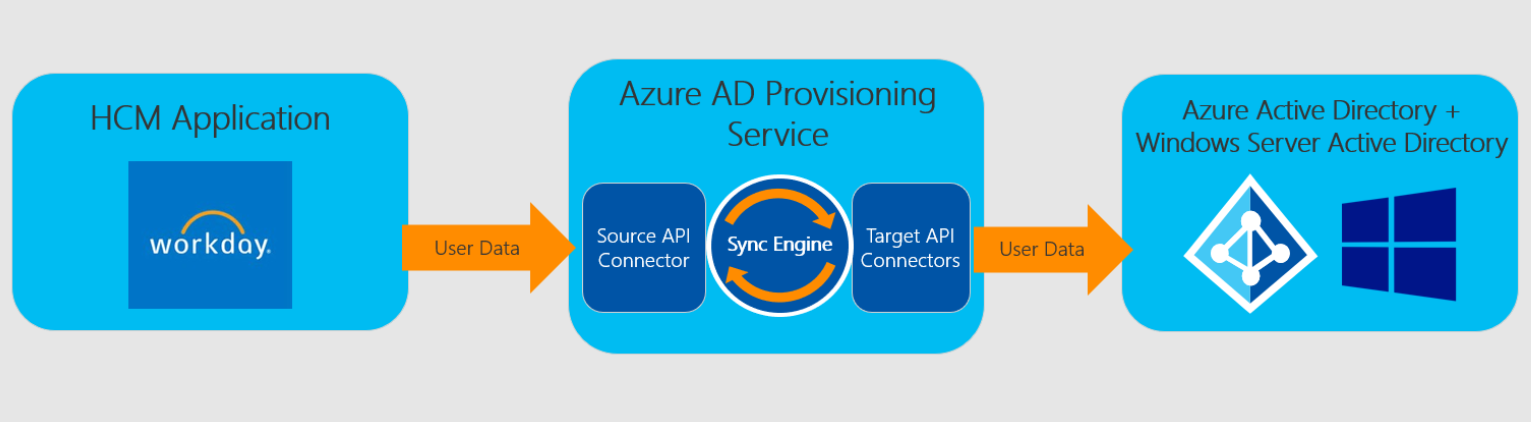
### Azure Automated Provisioning

#### How does automatic provisioning work?

The Azure AD Provisioning Service provisions users to SaaS apps and other systems by connecting to user management API endpoints provided by each SaaS application vendor. These user management API endpoints allow Azure AD to programmatically create, remove, update, and delete (CRUD) users. For selected applications the provisioning service can also create, update, and remove additional identity-related objects, such as groups and roles.

*Figure 1: The Azure AD Provisioning Service*

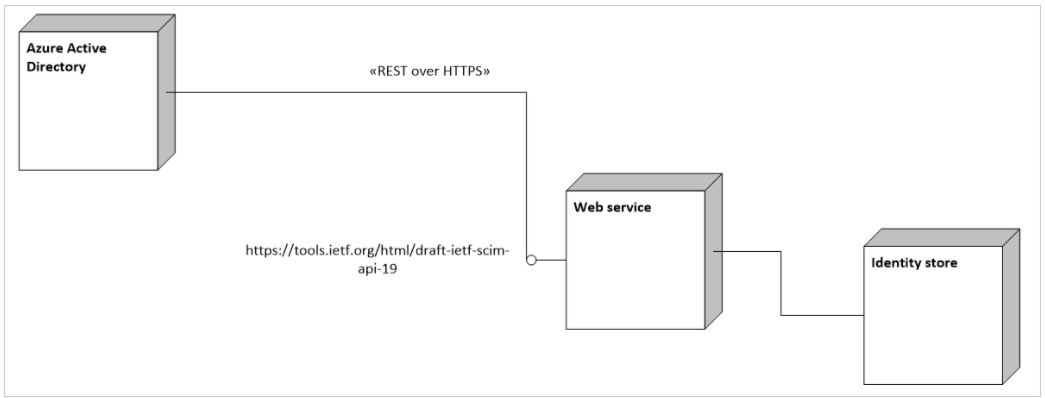
*Figure 2: "Outbound" user provisioning workflow from Azure AD to popular SaaS applications*

*Figure 3: "Inbound" user provisioning workflow from popular Human Capital Management (HCM) applications to Azure Active Directory and Windows Server Active Directory*

For more information review our [detailed documentation](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-saas-app-provisioning).

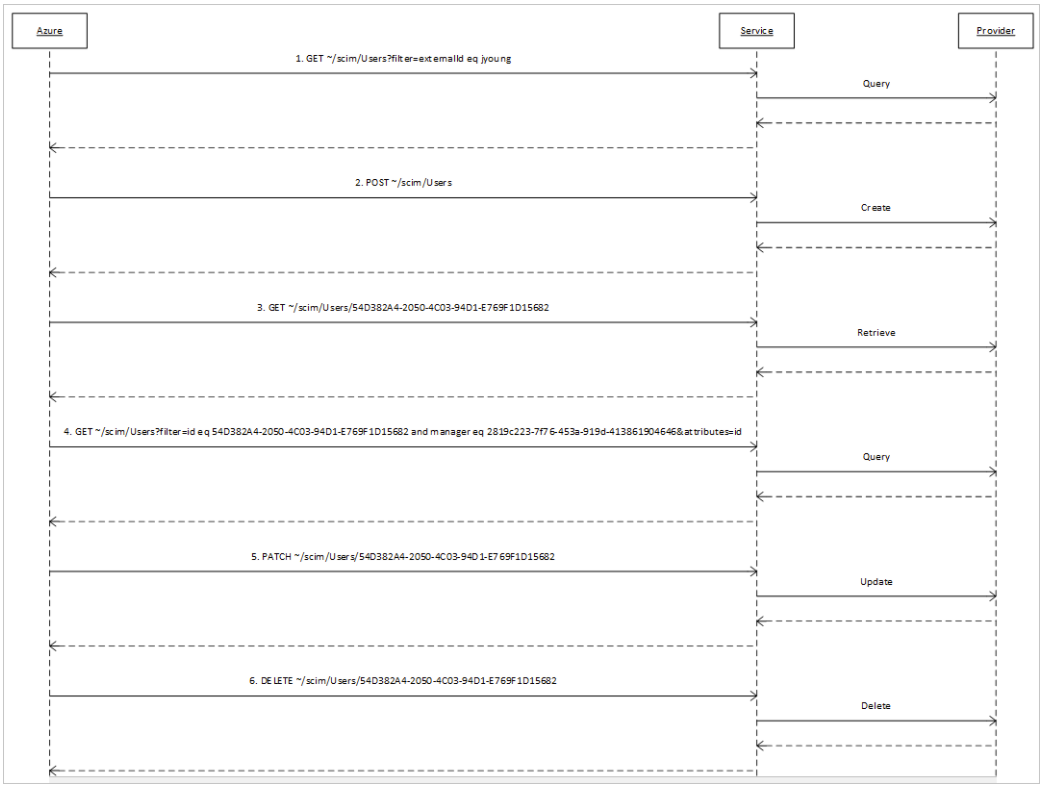
#### User provisioning and de-provisioning

Azure Active Directory can automatically provision users and groups to any application or identity store that is fronted by a web service with the interface defined in the System for Cross-Domain Identity Management (SCIM) 2.0 protocol specification. Azure Active Directory can send requests to create, modify, or delete assigned users and groups to the web service. The web service can then translate those requests into operations on the target identity store.



#### Group provisioning and de-provisioning

The following illustration shows the messages that Azure Active Directory sends to a SCIM service to manage the lifecycle of a user in another identity store. The diagram also shows how a SCIM service implemented using the CLI libraries provided by Microsoft for building such services translate those requests into calls to the methods of a provider.



The following illustration shows the messages that Azure AcD sends to a SCIM service to manage the lifecycle of a group in another identity store. Those messages differ from the messages pertaining to users in three ways:

* The schema of a group resource is identified as http://schemas.microsoft.com/2006/11/ResourceManagement/ADSCIM/Group.
* Requests to retrieve groups stipulate that the members attribute is to be excluded from any resource provided in response to the request.
* Requests to determine whether a reference attribute has a certain value are requests about the members attribute.

**Note:** (*For more information on Azure AD automated provisioning for* [SCIM-enabled apps please review our documentation*.*](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-scim-provisioning))

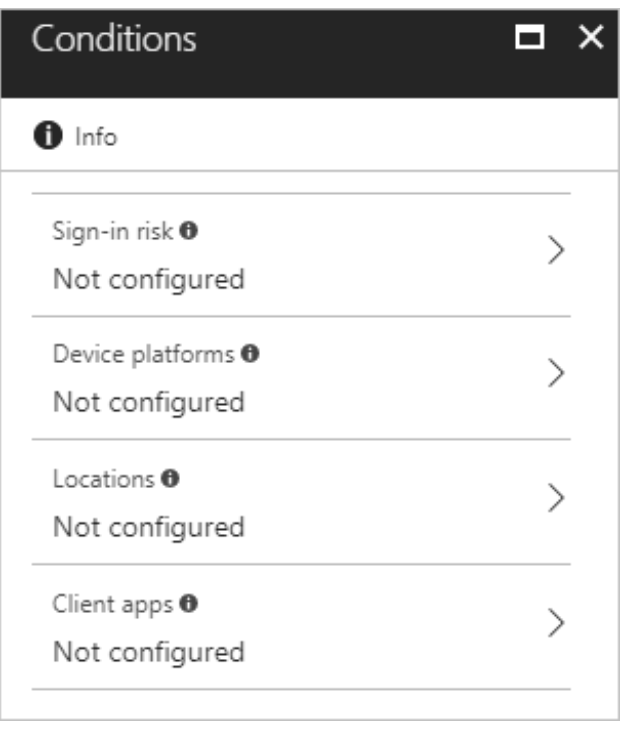
### Azure Conditional Access

In a conditional access policy, controls define what happens when a condition statement has been satisfied. With controls, you can either block access or allow access with additional requirements. When you configure a policy that allows access, you need to select at least one requirement.

There are two types of controls:

* **Grant controls** govern whether a user can complete authentication and sign-in to the application. If you have multiple controls selected, you can configure whether all of them are required when your policy is processed.
* **Session controls** enable limiting experience within a cloud app. The session controls are enforced by SaaS applications and rely on additional information provided by Azure AD to the application about the session. For example…..

Learn more via the [Conditional Access documentation.](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-conditional-access-azure-portal)



# Technical Requirements

## Azure AD & Application Environments

In the following tables, document the tenant names and URLs

| **Environment** | **Product** | **Tenant Information** |
| --- | --- | --- |
| Test Q/A | Azure AD | Example:  contosotest.onmicrosoft.com |
| Prod | Azure AD | Contoso.onmicrosoft.com |
| Test Q/A | <Insert App Name> | Example:  https://ec-crm-contoso--qa.cs19.my.<Insert App Name>.com |
| Prod | <Insert App Name> |  |

## Azure Single Sign-on Requirements

The following tables detail the requirements to configure your specific application including the necessary environment(s), endpoints, claim mapping, required attributes, certificates, and protocols used. You will be required to use this information to configure the Single Sign-on portion of your deployment in the [Azure AD admin portal.](aad.portal.azure.com)

#### Protocols Supported by Application

For all pre-integrated SaaS apps, Microsoft provides a tutorial and you will not need this information. If the application is not in our application marketplace, you may need this information.

|  |  |  |
| --- | --- | --- |
| Current State for Authentication | Protocols Supported by Application | Protocol Being Configured with Azure AD |
| Forms-Based  SSO with AD FS  SSO with PingFederate  SSO with Okta | SAML 2.0  SAML 1.1  Open ID Connect  OAuth  Forms-Based Auth  WS-Fed  WS-Trust | Click to select a protocol. |

### Attribute Requirements

Below, select the attribute matching scheme you will use, and then document the attribute names and mapping.

**Attribute Requirements (if applicable)**

Primary identifier value matches identically to the value in Azure AD

Primary identifier value matches identically to the value in AD

Caps matches between Azure AD and within the application

(**Note:** *Case-sensitivity exists for some applications such as Salesforce with federationID)*

All attributes are available in the application that are required

All attributes are available in Azure AD that are required

**Attribute Mapping**

|  |  |  |
| --- | --- | --- |
| AD Attribute Name | Azure AD Attribute Name  *If Join() or ExtractMailPrefix(), write below values: N/A* | Application Attribute Name |
| <Input attribute name> | **<Input attribute name>**  Click here to select. | **<Input attributes if applicable>** |
|  | Click here to select. |  |
|  | Click here to select. |  |
|  | Click here to select. |  |
|  | Click here to select. |  |

### Certificate Requirements

The certificate for the application must be up to date, or there is a risk of users not being able to access the application. By default, most SaaS application certificates are good for 36 months. However, you may change that length in the application blade. Ensure you document the expiration and know how you will manage your certificate renewal.

|  |  |  |
| --- | --- | --- |
| Length of Certifcate | Date of Expiration | Metadata URL for Cert |
| <Input how many months> | **<Input cert expiration date>** | **<Input URL for Cert>** |

There are two ways to manage your certificates.

* ***Automatic Cert Rollover:***Microsoft supports [Signing key rollover in Azure AD](https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-signing-key-rollover). While this is our preferred method for managing certs. Not all ISV’s supports this scenario.
* **Manually Updating:** Every application has its own certificate that expires based on how you have defined. Before the application’s cert expires, you must create a new cert and send to the ISV. Alternatively, this can be pulled from the federation metadata. [Read more on federation metadata here.](https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-federation-metadata)

### Endpoint Requirements

Prior to configuring the application, you will need the following information.

**Production Environment**

|  |  |
| --- | --- |
| **Sign on URL** |  |
| **Identifier** |  |
| **Reply URL** |  |
| **SAML SSO Service URL** |  |
| **SAML entity ID** |  |

**Test Environment**

|  |  |
| --- | --- |
| **Sign on URL** |  |
| **Identifier** |  |
| **Reply URL** |  |

### Azure Automated Provisioning Requirements

|  |  |
| --- | --- |
| Scenarios | Results |
| Current State | Configured with Azure AD  Configured with 3rd Party Provisioning Connector; Which:  Not Configured yet |
| Provisioning workflows interested in | Outbound: Synchronizing users from Azure AD to app  Outbound: Synchronizing groups from Azure AD to app  Inbound: Importing roles from the app selected during assignment  Inbound: Synchronizing users from the app to Azure AD  Other: |
| Support user management API | Yes  No |
| Types of Actions required | Update  Create  Delete  Disable |
| UserName Admin | **Note:** *Username and password from the application is required to allow Azure AD to have permissions for read/write operations* |
| Provisioning Key Lifecycle | <If applicable> |
| Known Pain Points | <If applicable>  *Example: Resetting Password on UserName Admin account nulls the Provisioning Key* |
| Attribute Mapping | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Mapping Type | Source Attribute | Target attribute | Apply this mapping | Default Value if null (optional) | Match Objects & Precedence | | Choose an item. | Choose an item. | <application Schema Name> | Choose an item. |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  |   Note: (*Each application will have its own schema name for attributes. If there are custom attributes within the application, when setting up the provisioning connector it will import the attribute names. These will become available in the UX to choose after you establish a successful auto provisioning connection*) | |

### Azure Conditional Access Requirements

Conditional access is a capability of Azure Active Directory that enables you to enforce controls on the access to apps in your environment based on specific conditions. With controls, you can either tie additional requirements to the access or you can block it. The implementation of conditional access is based on policies. A policy-based approach simplifies your configuration experience because it follows the way you think about your access requirements.

Choose an item.

The information below documents the Azure CA policies that will be configured in the [Azure AD Admin portal](http://aad.portal.azure.com).

|  |  |  |  |
| --- | --- | --- | --- |
| Policy Name | Group | Conditions | Grant |
| <Insert Name of CA Policy> | <Insert Scope users or groups> | Sign-in Risk: Choose an item.  Device Platforms: Choose an item.  Locations:  Include: All trusted IPs  Exclude: All trusted IPs  Client Apps:  Browser  Mobile apps and desktop clients | Grant access  Require MFA  Require complaint  Require Hybrid Azure AD  Require client app (preview)  Multiple Controls: Choose an item. |

# Implementing Your Solution

The foundation of proper planning is the basis upon which you can deploy an application successfully with Azure Active Directory. It provides intelligent security and integration that simplifies onboarding while reducing the time for successful deployments. This combination ensure that your application is integrated with ease while mitigating down time for your end users.

Use the following phases to plan for and deploy your solution in your organization:

* [Phase 1: Implementation Steps](#_Phase_1:_Implementation)
* [Phase 2: Change Communications](#_Phase_2:_Change)
* [Phase 3: User Acceptance Testing (UAT) for SSO](#_Phase_3:_User)
* [Phase 4: Rollback Steps](#_Phase_4:_Rollback)

## Phase 1: Implementation Steps

In this section you will be able to get the instructions to deploy your solution. Use the following steps to implement your project:

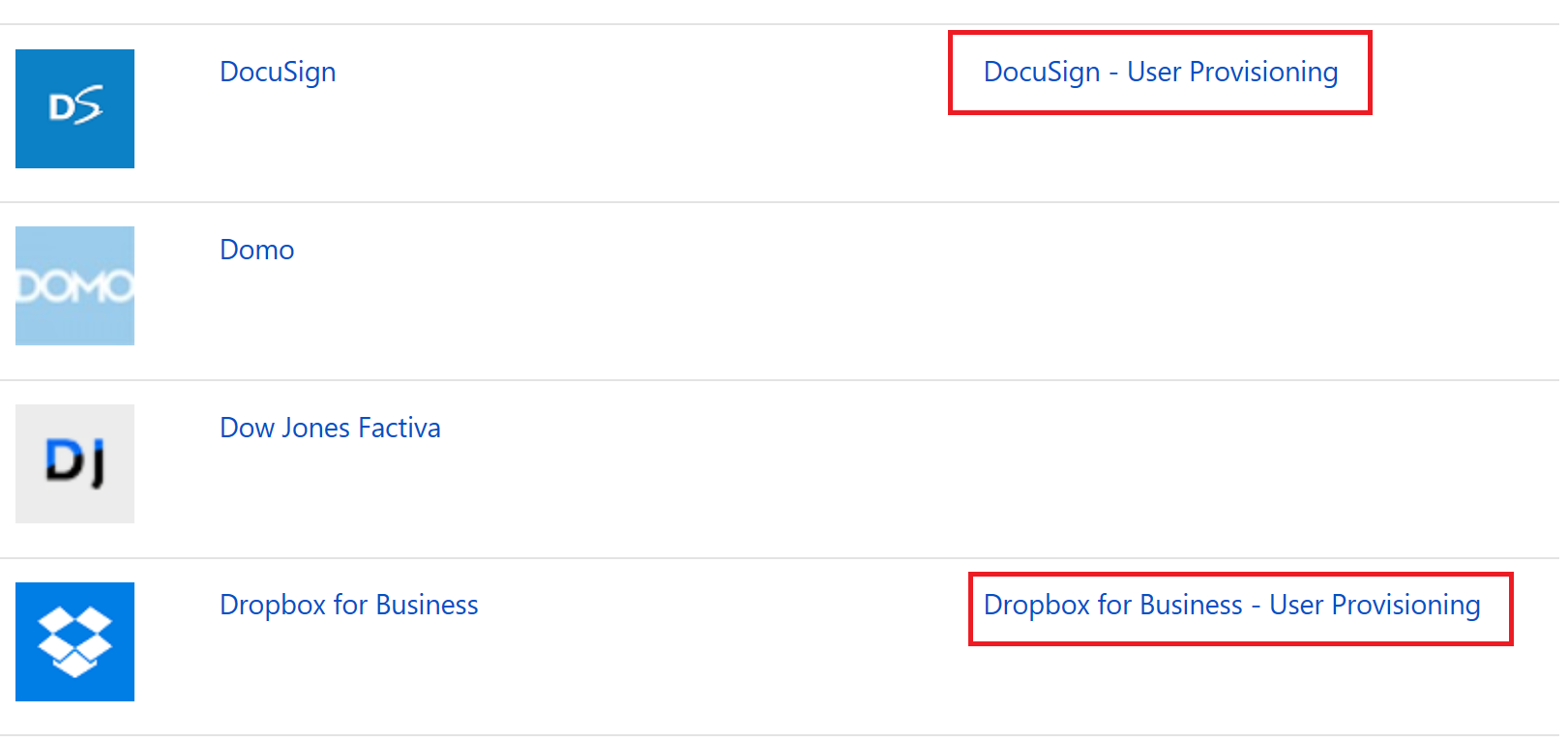
* [Step 1: Identify your Test Users](#_Step_1:_Identify)
* [Step 2: Configure Azure Automated Provisioning](#_Step_2:_Azure)
* [Step 3: Configure Azure Single Sign-on](#_Step_3:_Configure)
* [Step 4: Configure Azure Conditional Access](#_Step_4:_Configure)
* [Step 5: Configure Home Realm Discovery](#_Step_5:_Configure)

### Step 1: Identify your Test Users

1. Reach out to the app owner and request they create a minimum of 2-3 test users within the application.
2. Ensure the information that you will be using later as the primary attribute is populated correctly and will match what will be available in Azure AD.
   1. **Note** (*In most cases this will map to the NameID for SAML-based applications. For JWOT tokens it’s the value name that you provide*).
3. Create the user in Azure AD either manually as a cloud-based user or sync the user from on-premises using Azure AD Connect sync engine. Ensure the information matches that you will be using in the claims being sent to the application.

### Step 2: Configure Azure Automated Provisioning

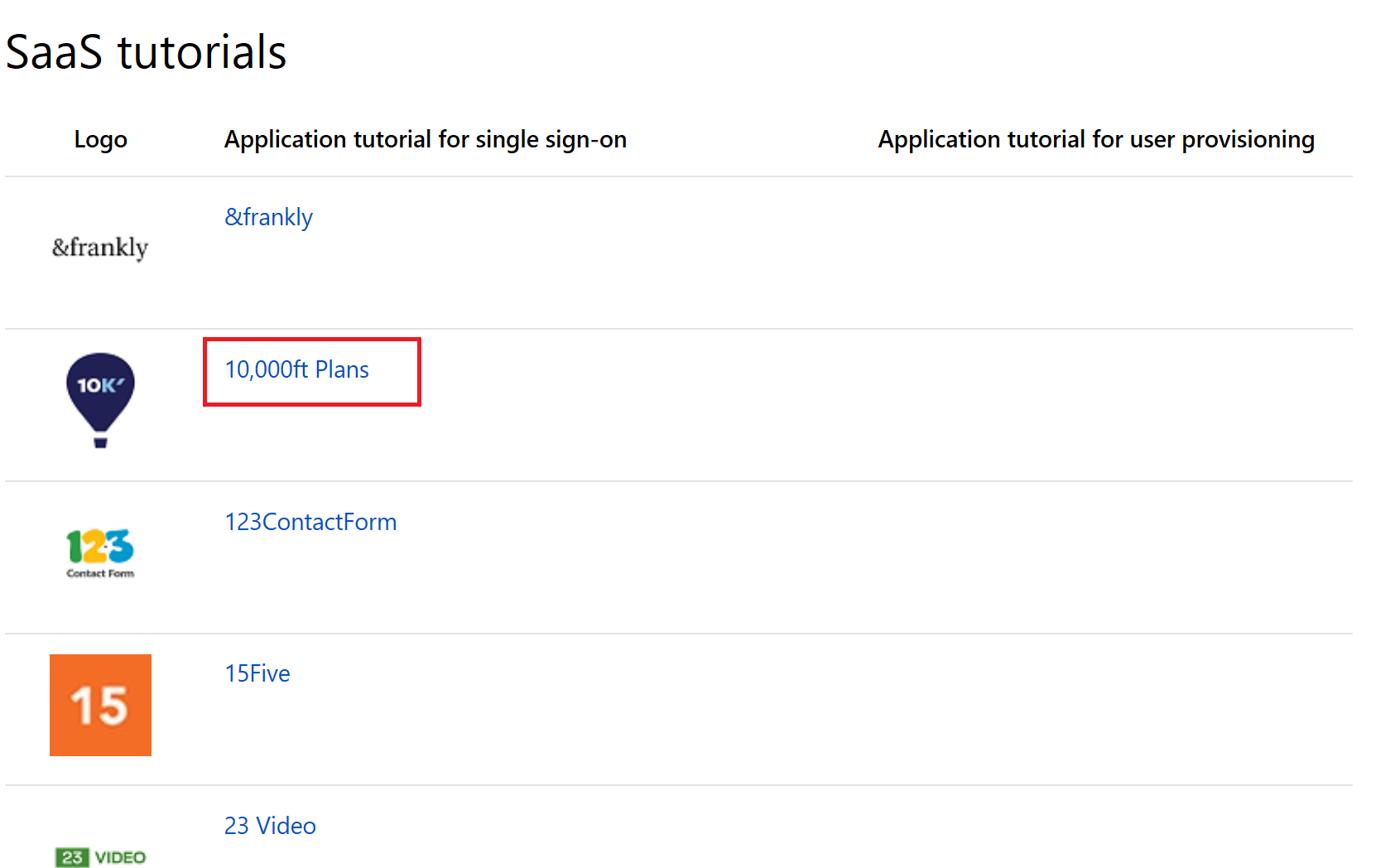
1. Navigate to the general tutorial to configure the Azure AD Components: <https://docs.microsoft.com/en-us/azure/active-directory/active-directory-saas-tutorial-list>
2. Locate the application name under Application tutorial for user provision. Click on the hyperlink and continue to follow the tutorial’s steps on to successfully configure your SaaS application.

**Example:**

1. If you do not locate your application, a provisioning connector has not been configured yet.
2. (Optional) Customize the user provisioning attribute mapping by using [Microsoft’s documentation](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-saas-customizing-attribute-mappings).
3. If you encounter issues during configuring, review the Audit logs through the Azure Management Portal. Learn more at our [Tutorial: Reporting on automatic user account provisioning](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-saas-provisioning-reporting).

### Step 3: Configure Azure Single Sign-on

1. Navigate to the general tutorial to configure the Azure AD Components: <https://docs.microsoft.com/en-us/azure/active-directory/active-directory-saas-tutorial-list>
2. Locate the application name under Application tutorial for single sign-on. Click on the hyperlink and continue to follow the tutorial’s steps on to successfully configure your SaaS application.

**Example:** 

1. If you do not locate your application, navigate to our Custom Application documentation. This will walk you through on how to add an application that is not located in the Azure AD gallery.
2. (Optional) Customize claims issued in the SAML token for enterprise application using [Microsoft’s guidance documentation](https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-saml-claims-customization). Ensure these map to what is expected to be received in the SAML response for your application.
3. If you encounter issues during configuration, use our [guidance on how to Debug SSO integration](https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-saml-debugging)
4. Note: (Custom application is an Azure AD Premium P1 or P2 licenses feature)

#### Step 4: Configure Azure Conditional Access

##### Trusted IP’s

|  |  |
| --- | --- |
| Scenarios | Results |
| Trusted IP’s | Already Configured  If Not, List named location/IP’s: |

### Step 5: Configure Home Realm Discovery

Some organizations configure their Azure Active Directory tenant to federate with another IdP, such as ADFS, for user authentication. In these cases, when signing into an application, the user is first presented with an Azure AD sign-in page and once they have typed their UPN they are then taken to the IdP sign-in page to complete the process.

Administrators may wish to have users directed straight to the sign-in page when signing in to specific applications, skipping the initial Azure Active Directory page. This is referred to as “sign-in auto-acceleration”.

Home Realm Discovery is the process through which the Azure Active Directory figures out at sign-in time where a user needs to authenticate. When signing in to an Azure AD tenant to access a resource, or the Azure AD common sign-in page, the user types a user name (UPN). Azure AD uses that to discover where the user needs to sign-in.

In cases where the tenant is federated to another IdP for sign-in, enabling auto-acceleration makes user sign-in more streamlined in cases where you know that everyone signing in can be authenticated by that IdP. You can configure auto-acceleration for individual applications.

**Note:** If you configure an application for auto-acceleration, **guest users will not be able to sign-in**. Taking the user straight to a federated IdP for authentication is a one-way street as there is no way to get back to the Azure Active Directory sign-in page. Guest users, who may need to be directed to other tenants or an external IdP like Microsoft account to be authenticated, won’t be able to sign in to that application as the Home Realm Discovery step is being skipped.

In the examples below we'll be creating, updating, linking, and deleting policies on application service principals in Azure AD. If you are new to Azure AD and unfamiliar with the terminology and concepts, checkout [this article](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-howto-tenant) to help you get started before proceeding with these samples.

1. To begin, download the latest [Azure AD PowerShell Cmdlet Preview](https://www.powershellgallery.com/packages/AzureADPreview).
2. Once you have the Azure AD PowerShell Cmdlets, run Connect command to sign into your Azure AD admin account. You'll need to do this whenever you start a new session.

Connect-AzureAD -Confirm

1. Run the following command to see all policies that have been created in your organization. This command should be used after most operations in the following scenarios. It will also help you get the **Object ID** of your policies.

Get-AzureADPolicy

##### Setting up HRD Policy

There are three steps to setting sign-in auto-acceleration on an application

1. [Creating an HRD policy for auto-acceleration](#_Create_an_HDR)
2. [Locating the Service Principle to which to attach the policy](#_Locate_the_Service)
3. [Attaching the policy to the service principle](#_Assign_the_Policy).

In this example, we will create a policy that auto-accelerates users to an ADFS sign-in screen when signing in to an application without them having to enter a username at the Azure AD sign-in page first.

We’ll create an HRD policy and apply it to the application’s service principal.

###### Create an HDR Policy for auto-acceleration

New-AzureADPolicy -Definition @("{`"HomeRealmDiscoveryPolicy`":{`"AccelerateToFederatedDomain`": true,`"PreferredDomain`":`"contoso.com`"}}") -DisplayName Contoso\_HRD\_Policy -Type HomeRealmDiscoveryPolicy

To see your new policy and get its ObjectID, run the following command.

Get-AzureADPolicy

Once you have an HRD policy to enable auto-acceleration created, you can assign it to multiple application service principles.

###### Locate the Service Principal to which to attach the Policy

You need the **ObjectId** of the service principals you want to assign the policy to. There are several ways to find the object Id of service principals. You can use the portal, you can query the [Microsoft Graph](https://msdn.microsoft.com/Library/Azure/Ad/Graph/api/entity-and-complex-type-reference) or go to our [Graph Explorer Tool](https://graphexplorer.cloudapp.net/) and sign into your Azure AD account to see all your organization's service principals, or since we’re using powershell, you can use the get-AzureADServicePrincipal cmdlet to list the service principles and their Ids.

e.g. *Get-AzureADServicePrincipal (Pulled ObjectID from <Insert App Name> App)*

###### Attaching the Policy to the Service Principal

Once you have the **ObjectId** of the service principal of the application you want to configure auto-acceleration for, run the following command to associate the HRD policy you created in step 1 with the service principal you located in step 2.

**Example:**

*Add-AzureADServicePrincipalPolicy -Id 03ef8024-2683-4c0e-b892-f8cdcb9c398c -RefObjectId 60a1792d -9f84-4a74-94e2-deaa329bd7ad*

(First ID is object ID of <Insert App Name> App, second is ObjectID of Policy)

**Note:** *To check which applications have auto-acceleration policy configured use:*

*Get-AzureADPolicyAppliedObject -ObjectId <ObjectId of the Policy>*

*Policies may have been created in a tenant but they don’t have any effect until they are attached to an entity. In the case of an HRD policy, it can be attached to a Service Principal Only one HRD policy can be active on a given entity at any one time.*

## Phase 2: Change Communications

### Step 1: Provide Internal Change Communication to end users

1. The end user experience will change when accessing your application moving forward. Use the following template example to communicate to end users about these changes to reduce help desk calls and drive positive adoption for your deployment



## Phase 3: User Acceptance Testing (UAT) for SSO

### Step 1: Create test cases for your application deployment

1. The following tests will be conducted with both Corporate Own devices and personal devices. These test cases should reflect your Business Use Cases. These will be used to verify whether this solution meets your requirements.

**See Examples below:**

|  |  |  |
| --- | --- | --- |
| Scenarios | Expected Results | Actual Results |
|  | | |
| Authorized User logs into <Insert App Name> with IE while on corp (SP-initiated) | User navigates to <Insert App Name> URL and initiates SP-initiated flow. IWA occurs with no additional prompts |  |
| Authorized User logs into <Insert App Name> with IE while off corp (SP-initiated) with new login attempt | User navigates to <Insert App Name> URL and initiates SP-initiated flow. Forms-based prompt at AD FS Sever. User successfully logs in and browser prompts for MFA. |  |
| Authorized User logs into <Insert App Name> with IE while off corp (SP-initiated) with a current session and has never performed MFA | User navigates to <Insert App Name> URL and initiates SP-initiated flow. User does not receive prompt for first factor. User receives prompt for MFA. |  |
| Authorized User logs into <Insert App Name> with IE while off corp (SP-initiated) with a current session and has already performed MFA in this session | User navigates to <Insert App Name> URL and initiates SP-initiated flow. User does not receive prompt for first factor. User does not receive MFA. User SSO’s into <Insert App Name> |  |
| Authorized User logs into <Insert App Name> with Chrome/Firefox/Safari while off corp network (SP-initiated) with a current session and has already performed MFA in this session | User navigates to <Insert App Name> URL and initiates SP-initiated flow. User does not receive prompt for first factor. User does not receive MFA. User SSO’s into <Insert App Name> |  |
| Authorized User logs into <Insert App Name> with Chrome/Firefox/Safari while off corp network (SP-initiated) with new login attempt | User navigates to <Insert App Name> URL and initiates SP-initiated flow. Forms-based prompt at AD FS Sever. User successfully logs in and browser prompts for MFA. |  |
| Authorized User logs into <Insert App Name> with Chrome/Firefox while on corp network (SP-initiated) with a current session | User navigates to <Insert App Name> URL and initiates SP-initiated flow. User does not receive prompt for first factor. User does not receive MFA. User SSO’s into <Insert App Name> |  |
| Authorized User logs into <Insert App Name> with <Insert App Name> mobile app (SP-initiated) with a new login attempt | User navigates to <Insert App Name> URL and initiates SP-initiated flow. Forms-based prompt at AD FS Sever. User successfully logs in and ADAL client prompts for MFA. |  |
| Unauthorized User attempts to log into <Insert App Name> with login URL (SP-initiated) | User navigates to <Insert App Name> URL and initiates SP-initiated flow. Forms-based prompt at AD FS Sever. User fails to login with first factor |  |
| Unauthorized user with validate Azure SSO Credentials |  |  |
| Authorized user attempts to log in but enters an incorrect password |  |  |
| Authorized user attempts to log in from off-network but enters incorrect security code or pin |  |  |
| Authorized user logs out and then logs in again |  |  |
| Authorized user clicks on link in an email and is already authenticated |  |  |
| Authorized user clicks on link in an email and is not yet authenticated |  |  |

#### Step 2: Document your results

1. Document the outcomes for both **Expected Results** and **Actual Results** in [Step 1](#_Step_1:_Create). Use this to determine to move forward into production based on [your established timelines](#_Established_Timelines).

#### Step 3: Moving into Production

1. After you complete all of testing based on your test cases, it’s time to move into production with your application.

## Phase 4: Rollback Steps

It’s important to plan what to do in the case during your deployment doesn’t go as planned. If the SSO configuration fails during the deployment, you must understand how to mitigate any outage and reduce impact to your users.

### Step 1: Identify available options for authentication for your specific application

Depending on what’s available, choose the best strategy below that aligns what’s available from the application.

1. If the application supports multiple IdP’s, leverage this. Do not delete existing federated SSO connection. Instead, disable existing federated SSO during the migration. This will allow you to switch back to existing federated SSO connection if you encounter issues during your deployment. More importantly, you should be able to maintain the existing endpoints and certificates that will decrease the change when reconfiguring during rolling back.
2. If the application is currently configured with forms-based authentication and the users are actively using username/password managed within the application, then verify with the application if there is an option to maintain this login information that could be used to switching back if federation configuration fails.
3. If the application does not support multiple IdP’s or you are not able to save the forms-based sign-in username/passwords, document the exact steps to fall back into your original configurations.
4. If the application is new and has not been configured yet, then rollback steps are not necessary

### Step 2: Document the SLA for changes

1. Document the SLA between updating endpoints, certificates, and user creation/updates. You will discover this during testing. While Azure AD allows you to make changes to the configurations in real-time, not every application has a self-service portal that enables you to make these changes. Instead, this may require you to open a support ticket to a representative from the application side.

# Operational Doc

## Purpose of Document

## Confidentiality Statement

It is understood and agreed to that project plan may provide certain information that is and must be kept confidential. To ensure the protection of such information you should not disclose any part of this plan to anyone unless required to do so by law.

## Required Roles

|  |  |  |  |
| --- | --- | --- | --- |
| Personas | Roles | Azure AD Role (if required) | Assign to |
|  | | |  |
| Help Desk Admin | Tier 1 Support | None |  |
| Identity Admin | Configure and Debug when issues impact Azure AD | Global Admin |  |
| <Insert App Name> Admin | User attestation in <Insert App Name>, configuration on users with permissions | None |  |
| Infrastructure Admins | Cert Rollover Owner | Global Admin |  |
| Business Owner/Stakeholder | User attestation in <Insert App Name>, configuration on users with permissions | None |  |

### Troubleshooting Guide & Steps

**Scenario 1 *Example***

|  |  |  |
| --- | --- | --- |
|  | Tier Levels | Steps |
| Debugging steps for a single account not being able to log into their account. |  | |
| Tier 1 | |  |  | | --- | --- | | Verify User login credential for device | If performing WIA | | Verify User is using the correct Login URL’s | In the scenario that the user owns multiple accounts | | Verify User has been assigned to <Insert App Name> | Located in Portal.Azure.com -> Locate App under “Enterprise Applications”. Confirm whether direct assignment or based on groups | | Verify User can log into MyApps and view the application in the Access Panel | Have user log into Access Panel at myapps.microsoft.com | | Verify User has an account in Azure AD and <Insert App Name> | Log into <Insert App Name>.com and verify within admin portal | | Verify User has the correct licenses for <Insert App Name> and Azure AD Premium licenses in Azure AD | View under the licenses section for both Portal.Azure.com and <Insert App Name>.com Amin portals | | Verify User’s account in <Insert App Name> is active and not disabled/inactive | If an account has become dormant, the account become inactive on <Insert App Name> that could lead to failure to authenticate | |
| Tier 2 | |  |  | | --- | --- | | Collect Fiddler Trace(s) of Repro | https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-saml-debugging | | Validate endpoints are correct & claim values | SAMLResponse to <Insert App Name> from Azure AD | | Validate claims (NameID) | Confirm both values match in Azure AD and <Insert App Name> for primary claim map | |
| Tier 3 | |  |  | | --- | --- | | Notify Business Owner | Communicate the current state to the Stakeholder for <Insert App Name> | | Create a Premier Support Ticket | Include Repro Steps, UPN, CorrelationID, Timestamp, and Fiddler Trace(s).  **Note:** *Make multiple attempts to authenticate and provide timeframe.* | |

**Scenario 2 *Example***

|  |  |  |
| --- | --- | --- |
|  | Tier Levels | Steps |
|  |  | |
| Debugging steps for complete outage of the <Insert App Name> | Tier 3 | |  |  | | --- | --- | | Notify Business Owner | Communicate the current state to the Stakeholder for <Insert App Name> | | Collect Fiddler Trace(s) of Repro | https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-saml-debugging | | Create a Premier Support Ticket | Include Repro Steps, UPN, CorrelationID, Timestamp, and Fiddler Trace(s).  **Note:** *Make multiple attempts to authenticate and provide timeframe.* | | Review sign-in logs in both Admin Portals | <Insert App Name> and Azure AD both has audit logs. Review these logs to determine whether there is a known issue. | | Review current documentation and ensure configured correctly | https://docs.microsoft.com/en-us/azure/active-directory/active-directory-saas-<Insert App Name>-tutorial | | Validate endpoints are correct | Compare Reply URL, Identifier, and Login URL between <Insert App Name> and Azure AD  View in the SAMLResponse/SAMLRequest that information matches | | Validate Certificate has not expired | This is located in both <Insert App Name>’s IdP setting as well in Azure AD’s Admin Portal. If cert has rolled over, you may have to update cert | | Validate claims mapping | Confirm both values match in Azure AD and <Insert App Name> for all claims, especially the primary claim. This can be viewed in Fiddler under SAMLResponse | | Work with MS to close support ticket | If this application is considered high impact, open a Sev 1 ticket against Premier | |

**Scenario 3 *Example***

|  |  |  |
| --- | --- | --- |
|  | Tier Levels | Steps |
| User Requesting to gain access to an aplication |  | |
| Tier 1 | Forward request to Key User |

**Scenario 4 *Example***

|  |  |  |
| --- | --- | --- |
|  | Tier Levels | Steps |
| User cannot locate Application |  | |
| Tier 1 | * Provide Single Sign-on URL * Give instructions to Access Panel (Myapps.microsoft.com) |

## Single Sign-on Certificate Lifecycle [Azure Active Directory]

|  |  |
| --- | --- |
| Scenarios | Results |
|  | |
| Owner for updating user properties in <Insert App Name> | **Example:**   1. *Request is submitted to Key Users from Support* 2. *Key Users submit a request to ISV Vendor* 3. *ISV Vendor creates a change within <Insert App Name>* |
| Owner On call for application break/fix support | **Example:**   1. *Escalate to Key Users* |
| Owner of rolling over certificate | **Example:**  ***Owner: App Owner***  ***Process:***   1. *Notification is received by Support team* 2. *Support Notifies Product Owner/Business Owner* 3. *Business Owner Approves* 4. *Support Creates a ticket to Ticketing System* 5. *Ticket is Assigned to the Infrastructure team* 6. *Infrastructure team will create Cert in Azure AD and deliver the certificate <Insert App Name> App Team for update* |
| Notification Email for cert notification: | <insert a DL> |
| Estimated lifetime of application |  |

## Access Management

|  |  |
| --- | --- |
| Scenarios | Results |
|  | |
| Group Management | Yes  No  If Yes, name of Groups and Owner(s): |
| Location of Group | On-Premises and synced  Azure AD |
| Application | Yes  No  If Yes, name of Owner(s): |
| Attestation | **Example:**  *Process:*   1. *Annual evaluation from audit team through manual steps*   *Deprovisioning process:*   1. *User is removed from HR System* 2. *Email is sent to Support & Business Owner(s)* 3. *User is manually removed from <Insert App Name>*   *Access Review iteration date (e.g. every 120 days): Annually* |

## Documentation

|  |  |
| --- | --- |
| Scenarios | Results |
|  | |
| <Insert App Name> SSO Tutorial | [https://docs.microsoft.com/en-us/azure/active-directory/active-directory-saas-<Insert App Name>-tutorial](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-saas-salesforce-tutorial) |
| <Insert App Name> Automatic User Provisioning Tutorial | [https://docs.microsoft.com/en-us/azure/active-directory/active-directory-saas-<Insert App Name>-provisioning-tutorial](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-saas-salesforce-provisioning-tutorial) |
| Debug SAML-based SSO | <https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-saml-debugging> |
| Customizing claim issued in SAML token | <https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-saml-claims-customization> |
| Single Sign-on SAML protocol | <https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-single-sign-on-protocol-reference> |
| Single Sign-Out SAML protocol | <https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-single-sign-out-protocol-reference> |
| Azure AD B2B | <https://docs.microsoft.com/en-us/azure/active-directory/active-directory-b2b-what-is-azure-ad-b2b> |
| Azure AD Conditional Access | <https://docs.microsoft.com/en-us/azure/active-directory/active-directory-conditional-access-azure-portal> |
| Azure Identity Protection | <https://docs.microsoft.com/en-us/azure/active-directory/active-directory-identityprotection> |
| SSO access | <https://docs.microsoft.com/en-us/azure/active-directory/active-directory-appssoaccess-whatis> |
| MFA Conditional Access for SaaS | <https://docs.microsoft.com/en-us/azure/active-directory/active-directory-playbook-building-blocks#mfa-conditional-access-for-saas-applications> |
| Configure Token Lifetimes | <https://docs.microsoft.com/en-us/azure/active-directory/active-directory-configurable-token-lifetimes> |
| Claim mapping for Apps via PowerShell | <https://docs.microsoft.com/en-us/azure/active-directory/active-directory-claims-mapping> |

# 

1. These recommended security features require an Azure AD Premium license, as opposed to a Basic or Free license. Microsoft recommends enabling them. [↑](#footnote-ref-2)