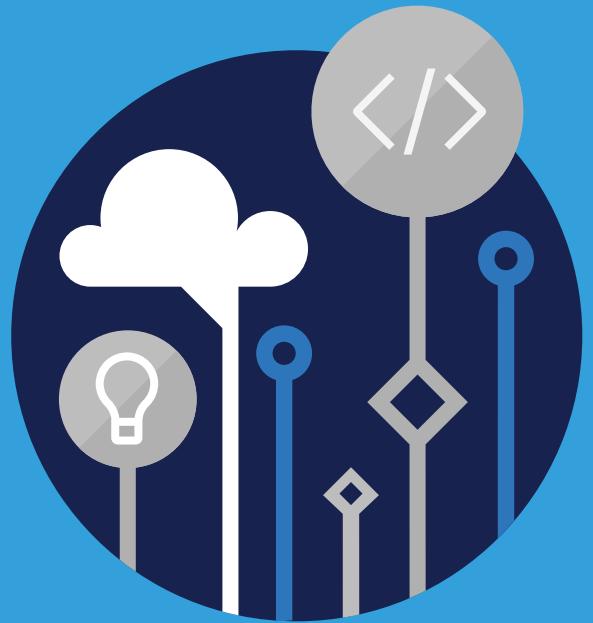


Microsoft
Official
Course



SC-300T00

Microsoft Identity and
Access Administrator

SC-300T00

**Microsoft Identity and Access
Administrator**

II Disclaimer

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2019 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at <http://www.microsoft.com/trademarks>¹ are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

¹ <http://www.microsoft.com/trademarks>

MICROSOFT LICENSE TERMS

MICROSOFT INSTRUCTOR-LED COURSEWARE

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to your use of the content accompanying this agreement which includes the media on which you received it, if any. These license terms also apply to Trainer Content and any updates and supplements for the Licensed Content unless other terms accompany those items. If so, those terms apply.

**BY ACCESSING, DOWNLOADING OR USING THE LICENSED CONTENT, YOU ACCEPT THESE TERMS.
IF YOU DO NOT ACCEPT THEM, DO NOT ACCESS, DOWNLOAD OR USE THE LICENSED CONTENT.**

If you comply with these license terms, you have the rights below for each license you acquire.

1. DEFINITIONS.

1. "Authorized Learning Center" means a Microsoft Imagine Academy (MSIA) Program Member, Microsoft Learning Competency Member, or such other entity as Microsoft may designate from time to time.
2. "Authorized Training Session" means the instructor-led training class using Microsoft Instructor-Led Courseware conducted by a Trainer at or through an Authorized Learning Center.
3. "Classroom Device" means one (1) dedicated, secure computer that an Authorized Learning Center owns or controls that is located at an Authorized Learning Center's training facilities that meets or exceeds the hardware level specified for the particular Microsoft Instructor-Led Courseware.
4. "End User" means an individual who is (i) duly enrolled in and attending an Authorized Training Session or Private Training Session, (ii) an employee of an MPN Member (defined below), or (iii) a Microsoft full-time employee, a Microsoft Imagine Academy (MSIA) Program Member, or a Microsoft Learn for Educators – Validated Educator.
5. "Licensed Content" means the content accompanying this agreement which may include the Microsoft Instructor-Led Courseware or Trainer Content.
6. "Microsoft Certified Trainer" or "MCT" means an individual who is (i) engaged to teach a training session to End Users on behalf of an Authorized Learning Center or MPN Member, and (ii) currently certified as a Microsoft Certified Trainer under the Microsoft Certification Program.
7. "Microsoft Instructor-Led Courseware" means the Microsoft-branded instructor-led training course that educates IT professionals, developers, students at an academic institution, and other learners on Microsoft technologies. A Microsoft Instructor-Led Courseware title may be branded as MOC, Microsoft Dynamics, or Microsoft Business Group courseware.
8. "Microsoft Imagine Academy (MSIA) Program Member" means an active member of the Microsoft Imagine Academy Program.
9. "Microsoft Learn for Educators – Validated Educator" means an educator who has been validated through the Microsoft Learn for Educators program as an active educator at a college, university, community college, polytechnic or K-12 institution.
10. "Microsoft Learning Competency Member" means an active member of the Microsoft Partner Network program in good standing that currently holds the Learning Competency status.
11. "MOC" means the "Official Microsoft Learning Product" instructor-led courseware known as Microsoft Official Course that educates IT professionals, developers, students at an academic institution, and other learners on Microsoft technologies.
12. "MPN Member" means an active Microsoft Partner Network program member in good standing.

13. "Personal Device" means one (1) personal computer, device, workstation or other digital electronic device that you personally own or control that meets or exceeds the hardware level specified for the particular Microsoft Instructor-Led Courseware.
 14. "Private Training Session" means the instructor-led training classes provided by MPN Members for corporate customers to teach a predefined learning objective using Microsoft Instructor-Led Courseware. These classes are not advertised or promoted to the general public and class attendance is restricted to individuals employed by or contracted by the corporate customer.
 15. "Trainer" means (i) an academically accredited educator engaged by a Microsoft Imagine Academy Program Member to teach an Authorized Training Session, (ii) an academically accredited educator validated as a Microsoft Learn for Educators – Validated Educator, and/or (iii) a MCT.
 16. "Trainer Content" means the trainer version of the Microsoft Instructor-Led Courseware and additional supplemental content designated solely for Trainers' use to teach a training session using the Microsoft Instructor-Led Courseware. Trainer Content may include Microsoft PowerPoint presentations, trainer preparation guide, train the trainer materials, Microsoft One Note packs, classroom setup guide and Pre-release course feedback form. To clarify, Trainer Content does not include any software, virtual hard disks or virtual machines.
2. **USE RIGHTS.** The Licensed Content is licensed, not sold. The Licensed Content is licensed on a **one copy per user basis**, such that you must acquire a license for each individual that accesses or uses the Licensed Content.
- 2.1 Below are five separate sets of use rights. Only one set of rights apply to you.
 1. **If you are a Microsoft Imagine Academy (MSIA) Program Member:**
 1. Each license acquired on behalf of yourself may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.
 2. For each license you acquire on behalf of an End User or Trainer, you may either:
 1. distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User who is enrolled in the Authorized Training Session, and only immediately prior to the commencement of the Authorized Training Session that is the subject matter of the Microsoft Instructor-Led Courseware being provided, **or**
 2. provide one (1) End User with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**
 3. provide one (1) Trainer with the unique redemption code and instructions on how they can access one (1) Trainer Content.
 3. For each license you acquire, you must comply with the following:
 1. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,
 2. you will ensure each End User attending an Authorized Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Authorized Training Session,
 3. you will ensure that each End User provided with the hard-copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End

User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,

4. you will ensure that each Trainer teaching an Authorized Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Authorized Training Session,
5. you will only use qualified Trainers who have in-depth knowledge of and experience with the Microsoft technology that is the subject of the Microsoft Instructor-Led Courseware being taught for all your Authorized Training Sessions,
6. you will only deliver a maximum of 15 hours of training per week for each Authorized Training Session that uses a MOC title, and
7. you acknowledge that Trainers that are not MCTs will not have access to all of the trainer resources for the Microsoft Instructor-Led Courseware.

2. If you are a Microsoft Learning Competency Member:

1. Each license acquire may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.
2. For each license you acquire on behalf of an End User or MCT, you may either:
 1. distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User attending the Authorized Training Session and only immediately prior to the commencement of the Authorized Training Session that is the subject matter of the Microsoft Instructor-Led Courseware provided, **or**
 2. provide one (1) End User attending the Authorized Training Session with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**
 3. you will provide one (1) MCT with the unique redemption code and instructions on how they can access one (1) Trainer Content.
3. For each license you acquire, you must comply with the following:
 1. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,
 2. you will ensure that each End User attending an Authorized Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Authorized Training Session,
 3. you will ensure that each End User provided with a hard-copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,

4. you will ensure that each MCT teaching an Authorized Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Authorized Training Session,
5. you will only use qualified MCTs who also hold the applicable Microsoft Certification credential that is the subject of the MOC title being taught for all your Authorized Training Sessions using MOC,
6. you will only provide access to the Microsoft Instructor-Led Courseware to End Users, and
7. you will only provide access to the Trainer Content to MCTs.

3. If you are a MPN Member:

1. Each license acquired on behalf of yourself may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.
2. For each license you acquire on behalf of an End User or Trainer, you may either:
 1. distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User attending the Private Training Session, and only immediately prior to the commencement of the Private Training Session that is the subject matter of the Microsoft Instructor-Led Courseware being provided, **or**
 2. provide one (1) End User who is attending the Private Training Session with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**
 3. you will provide one (1) Trainer who is teaching the Private Training Session with the unique redemption code and instructions on how they can access one (1) Trainer Content.
3. For each license you acquire, you must comply with the following:
 1. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,
 2. you will ensure that each End User attending an Private Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Private Training Session,
 3. you will ensure that each End User provided with a hard copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,
 4. you will ensure that each Trainer teaching an Private Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Private Training Session,

5. you will only use qualified Trainers who hold the applicable Microsoft Certification credential that is the subject of the Microsoft Instructor-Led Courseware being taught for all your Private Training Sessions,
6. you will only use qualified MCTs who hold the applicable Microsoft Certification credential that is the subject of the MOC title being taught for all your Private Training Sessions using MOC,
7. you will only provide access to the Microsoft Instructor-Led Courseware to End Users, and
8. you will only provide access to the Trainer Content to Trainers.

4. If you are an End User:

For each license you acquire, you may use the Microsoft Instructor-Led Courseware solely for your personal training use. If the Microsoft Instructor-Led Courseware is in digital format, you may access the Microsoft Instructor-Led Courseware online using the unique redemption code provided to you by the training provider and install and use one (1) copy of the Microsoft Instructor-Led Courseware on up to three (3) Personal Devices. You may also print one (1) copy of the Microsoft Instructor-Led Courseware. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.

5. If you are a Trainer.

1. For each license you acquire, you may install and use one (1) copy of the Trainer Content in the form provided to you on one (1) Personal Device solely to prepare and deliver an Authorized Training Session or Private Training Session, and install one (1) additional copy on another Personal Device as a backup copy, which may be used only to reinstall the Trainer Content. You may not install or use a copy of the Trainer Content on a device you do not own or control. You may also print one (1) copy of the Trainer Content solely to prepare for and deliver an Authorized Training Session or Private Training Session.

2. If you are an MCT, you may customize the written portions of the Trainer Content that are logically associated with instruction of a training session in accordance with the most recent version of the MCT agreement.
3. If you elect to exercise the foregoing rights, you agree to comply with the following: (i) customizations may only be used for teaching Authorized Training Sessions and Private Training Sessions, and (ii) all customizations will comply with this agreement. For clarity, any use of "customize" refers only to changing the order of slides and content, and/or not using all the slides or content, it does not mean changing or modifying any slide or content.

- 2.2 **Separation of Components.** The Licensed Content is licensed as a single unit and you may not separate their components and install them on different devices.
- 2.3 **Redistribution of Licensed Content.** Except as expressly provided in the use rights above, you may not distribute any Licensed Content or any portion thereof (including any permitted modifications) to any third parties without the express written permission of Microsoft.
- 2.4 **Third Party Notices.** The Licensed Content may include third party code that Microsoft, not the third party, licenses to you under this agreement. Notices, if any, for the third party code are included for your information only.
- 2.5 **Additional Terms.** Some Licensed Content may contain components with additional terms, conditions, and licenses regarding its use. Any non-conflicting terms in those conditions and licenses also apply to your use of that respective component and supplements the terms described in this agreement.

3. **LICENSED CONTENT BASED ON PRE-RELEASE TECHNOLOGY.** If the Licensed Content's subject matter is based on a pre-release version of Microsoft technology ("Pre-release"), then in addition to the other provisions in this agreement, these terms also apply:
 1. **Pre-Release Licensed Content.** This Licensed Content subject matter is on the Pre-release version of the Microsoft technology. The technology may not work the way a final version of the technology will and we may change the technology for the final version. We also may not release a final version. Licensed Content based on the final version of the technology may not contain the same information as the Licensed Content based on the Pre-release version. Microsoft is under no obligation to provide you with any further content, including any Licensed Content based on the final version of the technology.
 2. **Feedback.** If you agree to give feedback about the Licensed Content to Microsoft, either directly or through its third party designee, you give to Microsoft without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft technology, Microsoft product, or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its technology, technologies, or products to third parties because we include your feedback in them. These rights survive this agreement.
 3. **Pre-release Term.** If you are an Microsoft Imagine Academy Program Member, Microsoft Learning Competency Member, MPN Member, Microsoft Learn for Educators – Validated Educator, or Trainer, you will cease using all copies of the Licensed Content on the Pre-release technology upon (i) the date which Microsoft informs you is the end date for using the Licensed Content on the Pre-release technology, or (ii) sixty (60) days after the commercial release of the technology that is the subject of the Licensed Content, whichever is earliest ("Pre-release term"). Upon expiration or termination of the Pre-release term, you will irretrievably delete and destroy all copies of the Licensed Content in your possession or under your control.
 4. **SCOPE OF LICENSE.** The Licensed Content is licensed, not sold. This agreement only gives you some rights to use the Licensed Content. Microsoft reserves all other rights. Unless applicable law gives you more rights despite this limitation, you may use the Licensed Content only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the Licensed Content that only allows you to use it in certain ways. Except as expressly permitted in this agreement, you may not:
 - access or allow any individual to access the Licensed Content if they have not acquired a valid license for the Licensed Content,
 - alter, remove or obscure any copyright or other protective notices (including watermarks), branding or identifications contained in the Licensed Content,
 - modify or create a derivative work of any Licensed Content,
 - publicly display, or make the Licensed Content available for others to access or use,
 - copy, print, install, sell, publish, transmit, lend, adapt, reuse, link to or post, make available or distribute the Licensed Content to any third party,
 - work around any technical limitations in the Licensed Content, or
 - reverse engineer, decompile, remove or otherwise thwart any protections or disassemble the Licensed Content except and only to the extent that applicable law expressly permits, despite this limitation.
 5. **RESERVATION OF RIGHTS AND OWNERSHIP.** Microsoft reserves all rights not expressly granted to you in this agreement. The Licensed Content is protected by copyright and other intellectual property

laws and treaties. Microsoft or its suppliers own the title, copyright, and other intellectual property rights in the Licensed Content.

6. **EXPORT RESTRICTIONS.** The Licensed Content is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the Licensed Content. These laws include restrictions on destinations, end users and end use. For additional information, see www.microsoft.com/exporting.
7. **SUPPORT SERVICES.** Because the Licensed Content is provided "as is", we are not obligated to provide support services for it.
8. **TERMINATION.** Without prejudice to any other rights, Microsoft may terminate this agreement if you fail to comply with the terms and conditions of this agreement. Upon termination of this agreement for any reason, you will immediately stop all use of and delete and destroy all copies of the Licensed Content in your possession or under your control.
9. **LINKS TO THIRD PARTY SITES.** You may link to third party sites through the use of the Licensed Content. The third party sites are not under the control of Microsoft, and Microsoft is not responsible for the contents of any third party sites, any links contained in third party sites, or any changes or updates to third party sites. Microsoft is not responsible for webcasting or any other form of transmission received from any third party sites. Microsoft is providing these links to third party sites to you only as a convenience, and the inclusion of any link does not imply an endorsement by Microsoft of the third party site.
10. **ENTIRE AGREEMENT.** This agreement, and any additional terms for the Trainer Content, updates and supplements are the entire agreement for the Licensed Content, updates and supplements.
11. **APPLICABLE LAW.**
 1. United States. If you acquired the Licensed Content in the United States, Washington state law governs the interpretation of this agreement and applies to claims for breach of it, regardless of conflict of laws principles. The laws of the state where you live govern all other claims, including claims under state consumer protection laws, unfair competition laws, and in tort.
 2. Outside the United States. If you acquired the Licensed Content in any other country, the laws of that country apply.
12. **LEGAL EFFECT.** This agreement describes certain legal rights. You may have other rights under the laws of your country. You may also have rights with respect to the party from whom you acquired the Licensed Content. This agreement does not change your rights under the laws of your country if the laws of your country do not permit it to do so.
13. **DISCLAIMER OF WARRANTY. THE LICENSED CONTENT IS LICENSED "AS-IS" AND "AS AVAILABLE." YOU BEAR THE RISK OF USING IT. MICROSOFT AND ITS RESPECTIVE AFFILIATES GIVES NO EXPRESS WARRANTIES, GUARANTEES, OR CONDITIONS. YOU MAY HAVE ADDITIONAL CONSUMER RIGHTS UNDER YOUR LOCAL LAWS WHICH THIS AGREEMENT CANNOT CHANGE. TO THE EXTENT PERMITTED UNDER YOUR LOCAL LAWS, MICROSOFT AND ITS RESPECTIVE AFFILIATES EXCLUDES ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.**
14. **LIMITATION ON AND EXCLUSION OF REMEDIES AND DAMAGES. YOU CAN RECOVER FROM MICROSOFT, ITS RESPECTIVE AFFILIATES AND ITS SUPPLIERS ONLY DIRECT DAMAGES UP TO US\$5.00. YOU CANNOT RECOVER ANY OTHER DAMAGES, INCLUDING CONSEQUENTIAL, LOST PROFITS, SPECIAL, INDIRECT OR INCIDENTAL DAMAGES.**

This limitation applies to

- anything related to the Licensed Content, services, content (including code) on third party Internet sites or third-party programs; and
- claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.

It also applies even if Microsoft knew or should have known about the possibility of the damages. The above limitation or exclusion may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential, or other damages.

Please note: As this Licensed Content is distributed in Quebec, Canada, some of the clauses in this agreement are provided below in French.

Remarque : Ce le contenu sous licence étant distribué au Québec, Canada, certaines des clauses dans ce contrat sont fournies ci-dessous en français.

EXONÉRATION DE GARANTIE. Le contenu sous licence visé par une licence est offert « tel quel ». Toute utilisation de ce contenu sous licence est à votre seule risque et péril. Microsoft n'accorde aucune autre garantie expresse. Vous pouvez bénéficier de droits additionnels en vertu du droit local sur la protection dues consommateurs, que ce contrat ne peut modifier. La ou elles sont permises par le droit locale, les garanties implicites de qualité marchande, d'adéquation à un usage particulier et d'absence de contrefaçon sont exclues.

LIMITATION DES DOMMAGES-INTÉRÊTS ET EXCLUSION DE RESPONSABILITÉ POUR LES DOMMAGES. Vous pouvez obtenir de Microsoft et de ses fournisseurs une indemnisation en cas de dommages directs uniquement à hauteur de 5,00 \$ US. Vous ne pouvez prétendre à aucune indemnisation pour les autres dommages, y compris les dommages spéciaux, indirects ou accessoires et pertes de bénéfices.

Cette limitation concerne:

- tout ce qui est relié au le contenu sous licence, aux services ou au contenu (y compris le code) figurant sur des sites Internet tiers ou dans des programmes tiers; et.
- les réclamations au titre de violation de contrat ou de garantie, ou au titre de responsabilité stricte, de négligence ou d'une autre faute dans la limite autorisée par la loi en vigueur.

Elle s'applique également, même si Microsoft connaissait ou devrait connaître l'éventualité d'un tel dommage. Si votre pays n'autorise pas l'exclusion ou la limitation de responsabilité pour les dommages indirects, accessoires ou de quelque nature que ce soit, il se peut que la limitation ou l'exclusion ci-dessus ne s'appliquera pas à votre égard.

EFFET JURIDIQUE. Le présent contrat décrit certains droits juridiques. Vous pourriez avoir d'autres droits prévus par les lois de votre pays. Le présent contrat ne modifie pas les droits que vous confèrent les lois de votre pays si celles-ci ne le permettent pas.

Revised April 2019



Contents

■	Module 0 Course Introduction	1
	About this course	1
■	Module 1 Implement an Identity Management Solution	5
	Learning Objectives	5
	Implement the Initial Configuration of Azure AD	7
	Create, Configure, and Manage Identities	31
	Implement and Manage External Identities	36
	Implement and Manage Hybrid Identity	64
	Module 1 Review Questions	109
	Module 1 Hands-on Exercises	112
	Module 1 Summary	118
■	Module 2 Implement Authentication and Access Solution	125
	Learning Objectives	125
	Plan and Implement Azure Multifactor Authentication (MFA)	127
	Manage User Authentication	137
	Plan, Implement, and Administer Conditional Access	156
	Manage Azure AD Identity Protection	186
	Module 2 Review Questions	199
	Module 2 Hands-on Exercises	200
	Module 2 Summary	205
■	Module 3 Implement Access Management for Apps	209
	Learning Objectives	209
	Plan and Design Integration of Enterprise Apps for SSO	211
	Implement and Monitor integration of Enterprise Apps for SSO	224
	Implement App Registration	238
	Module 3 Review Questions	250
	Module 3 Hands-on Exercises	252
	Module 3 Summary	255
■	Module 4 Plan and Implement an Identity Governance Strategy	261
	Learning Objectives	261
	Plan and Implement Entitlement Management	263
	Plan, Implement, and Manage Access Reviews	270
	Plan and Implement Privileged Access	292

Monitor and Maintain Azure Active Directory	310
Module 4 Review Questions	335
Module 4 Hands-on Exercises	338
Module 4 Summary	343

Module 0 Course Introduction

About this course

About this course

Course Description

This course provides knowledge and skills on identity and access management in the cloud using Azure Active Directory and its related components in Azure.

Level

Intermediate

Audience

This course is targeted to those looking to familiarize themselves with the skills and knowledge aligned to a Microsoft Identity and Access Administrator using Azure Active Directory and other across cloud-based and related Microsoft services.

This is a broad audience that may include cloud administrators, security engineers, new or existing IT professionals, or students that have an interest in identity and access management solutions.

The person taking this content should be familiar with Microsoft Azure and some knowledge of Microsoft 365 and wants to understand how Microsoft implements identity and access solutions in the cloud.

The content for this course aligns to the SC-300 exam objective domain.

Prerequisites

- General understanding of Azure, Azure AD, and cloud computing concepts.
- IT knowledge or experience working in an IT environment.

- General understanding of security concepts like Zero Trust, Defense in Depth, and Least Privileged Access.

Expected learning

- Implement an identity management solution based on Azure and Azure AD
- Implement an authentication and access management solution based on Azure and Azure AD
- Implement access management for apps within Azure
- Plan and implement an identity governance strategy based on Azure and Azure AD

Course Syllabus

Course Outline

Modules and lesson of the training are roughly aligned to the SC-300 Exam Skills Outline, available on the **certification page¹**.

Module 1 - Implement an Identity Management Solution

- Learning Objectives
- Implement the Initial Configuration of Azure AD
- Create, Configure, and Manage Identities
- Implement and Manage External Identities
- Implement and Manage Hybrid Identity
- Module 1 Review Questions
- Module 1 Summary

Module 2 - Implement Authentication and Access Solution

- Learning Objectives
- Plan and Implement Azure Multifactor Authentication (MFA)
- Manage User Authentication
- Plan, Implement, and Administer Conditional Access
- Manage Azure AD Identity Protection
- Module 2 Review Questions
- Module 2 Summary

Module 3 - Implement Access Management for Apps

- Learning Objectives
- Plan and Design the Integration of Enterprise Apps for SSO

¹ <https://docs.microsoft.com/learn/certifications/exams/sc-300>

-
- Implement and Monitor the Integration of Enterprise Apps for SSO
 - Implement App Registration
 - Module 3 Review Questions
 - Module 3 Summary

Module 4 - Plan and Implement and Identity Governance Strategy

- Learning Objectives
- Plan and Implement Entitlement Management
- Plan, Implement and Manage Access Reviews
- Plan and Implement Privileged Access
- Monitor and Maintain Azure Active Directory
- Module 4 Review Questions
- Module 4 Summary

Microsoft Learn SC-300 Self Study

If students or instructors want to refresh their skills and knowledge of identity and access management with online training, here are the links:

- **SC-300 part 1: Implement an identity management solution²**
- **SC-300 part 2: Implement an authentication and access management solution³**
- **SC-300 part 3: Implement access management for apps⁴**
- **SC-300 part 4: Plan and implement an identity governance strategy⁵**

SC-900 Certification Exam

The SC-300 **Microsoft Identity and Access Administrator certification exam⁶** is designed for candidates looking to demonstrate skills and knowledge of identity and access management across Azure, Azure AD and related Microsoft services.

The Microsoft Identity and Access Administrator designs, implements, and operates an organization's identity and access management systems by using Azure Active Directory (Azure AD). They manage tasks such as providing secure authentication and authorization access to enterprise applications. The administrator provides seamless experiences and self-service management capabilities for all users. Adaptive access and governance are core elements to the role. This role is also responsible for troubleshooting, monitoring, and reporting for the identity and access environment.

The Identity and Access Administrator may be a single individual or a member of a larger team. This role collaborates with many other roles in the organization to drive strategic identity projects to modernize identity solutions, to implement hybrid identity solutions, and to implement identity governance.

² <https://docs.microsoft.com/learn/paths/implement-identity-management-solution/>

³ <https://docs.microsoft.com/learn/paths/implement-authentication-access-management-solution/>

⁴ <https://docs.microsoft.com/learn/paths/implement-access-management-for-apps/>

⁵ <https://docs.microsoft.com/learn/paths/plan-implement-identity-governance-strategy/>

⁶ <https://docs.microsoft.com/learn/certifications/exams/sc-300>

The exam includes four study areas. The percentages indicate the relative weight of each area on the exam. The higher the percentage, the more questions the exam will contain. Be sure to read the exam page for specifics about what skills are covered in each area.

SC-300 Study Areas

Skills Measured	Weights
Implement an identity management solution	25-30%
Implement an authentication and access management solution	25-30%
Implement access management for apps	10-15%
Plan and implement an identity governance strategy	25-30%

Module 1 Implement an Identity Management Solution

Learning Objectives

Learning Objectives

After completing this module, you'll be able to:

- Implement initial configuration of Azure Active Directory
 - configure and manage Azure AD directory roles
 - configure and manage custom domains
 - configure and manage device registration options
 - configure delegation by using administrative units
 - configure tenant-wide settings
- Create, configure, and manage identities
 - create, configure, and manage users
 - create, configure, and manage groups
 - manage licenses
 - Implement and manage external identities
 - manage external collaboration settings in Azure Active Directory
 - invite external users (individually or in bulk)
 - manage external user accounts in Azure Active Directory
 - configure identity providers (social and SAML/WS-fed)
- Implement and manage hybrid identity
 - implement and manage Azure Active Directory Connect (AADC)

- implement and manage Password Hash Synchronization (PHS)
- implement and manage Pass-Through Authentication (PTA)
- implement and manage seamless Single Sign-On (SSO)
- implement and manage Federation excluding manual ADFS deployments
- implement and manage Azure Active Directory Connect Health
- troubleshoot synchronization errors

Implement the Initial Configuration of Azure AD

Introduction

In this module you will learn how to configure and manage Azure Active Directory (Azure AD) roles, custom domains, and device registration options. In addition, you will learn how to configure delegation by using administrative units and configure tenant-wide settings.

Learning objectives

In this module, you will:

- Configure and manage Azure Active Directory roles.
- Configure and manage custom domains.
- Configure and manage device registration options.
- Configure delegation by using administrative units.
- Configure tenant-wide settings

Prerequisites

None

Configure and Manage Roles

Azure Active Directory (Azure AD) is Microsoft's cloud-based identity and access management service, which helps your employees sign in and access resources in:

- **External resources**, such as Microsoft 365, the Azure portal, and thousands of other SaaS applications.
- **Internal resources**, such as apps on your corporate network and intranet, along with any cloud apps developed by your own organization.

Who uses Azure AD?

Azure AD is intended for:

- **IT admins** - As an IT admin, you can use Azure AD to control access to your apps and your app resources, based on your business requirements. For example, you can use Azure AD to require multi-factor authentication when accessing important organizational resources. Additionally, you can use Azure AD to automate user provisioning between your existing Windows Server AD and your cloud apps, including Microsoft 365. Finally, Azure AD gives you powerful tools to automatically help protect user identities and credentials and to meet your access governance requirements.
- **App developers** - As an app developer, you can use Azure AD as a standards-based approach for adding single sign-on (SSO) to your app, allowing it to work with a user's pre-existing credentials. Azure AD also provides APIs that can help you build personalized app experiences using existing organizational data.

- **Microsoft 365, Office 365, Azure, or Dynamics CRM Online subscribers** - As a subscriber, you're already using Azure AD. Each Microsoft 365, Office 365, Azure, and Dynamics CRM Online tenant is automatically an Azure AD tenant. You can immediately start to manage access to your integrated cloud apps.

In Azure AD, if one of your users needs permission to manage Azure AD resources, you must assign them to a role that provides the permissions they need.

If you are new to Azure, you may find it a little challenging to understand all the different roles in Azure. The following section helps explain the following roles and provides additional information on Azure roles and Azure AD roles:

- Classic subscription administrator roles
- Azure roles
- Azure AD roles

Azure AD roles

Azure AD roles are used to manage Azure AD resources in a directory such as create or edit users, assign administrative roles to others, reset user passwords, manage user licenses, and manage domains. The following table describes a few of the more important Azure AD roles.

Azure AD role	Permissions	Notes
Global Administrator	- Manage access to all administrative features in Azure Active Directory, as well as services that federate to Azure Active Directory - Assign administrator roles to others - Reset the password for any user and all other administrators	The person who signs up for the Azure Active Directory tenant becomes a Global Administrator.
User Administrator	- Create and manage all aspects of users and groups - Manage support tickets - Monitor service health - Change passwords for users, Helpdesk administrators, and other User Administrators	
Billing Administrator	- Make purchases - Manage subscriptions - Manage support tickets - Monitors service health	

In the Azure portal, you can see the list of Azure AD roles on the **Roles and administrators** blade.

Default Directory - Roles and administrators

Get just-in-time access to a role when you need it using PIM. Learn more about PIM →

Your Role: Global administrator and 2 other roles

ROLE	DESCRIPTION	...
Application administrator	Can create and manage all aspects of app registrations and enterprise...	...
Application developer	Can create application registrations independent of the 'Users can reg...	...
Billing administrator	Can perform common billing related tasks like updating payment info...	...
Cloud application administrator	Can create and manage all aspects of app registrations and enterprise...	...
Cloud device administrator	Full access to manage devices in Azure AD.	...
Compliance administrator	Can read and manage compliance configuration and reports in Azure
Conditional Access administrator	Can manage conditional access capabilities.	...
Customer LockBox access approver	Can approve Microsoft support requests to access customer organizatio...	...
Desktop Analytics administrator	Can access and manage Desktop management tools and services.	...
Dynamics 365 administrator	Can manage all aspects of the Dynamics 365 product.	...
Exchange administrator	Can manage all aspects of the Exchange product.	...
Global administrator	Can manage all aspects of Azure AD and Microsoft services that use A...	...
Guest inviter	Can invite guest users independent of the 'members can invite guests...	...
Information Protection administrator	Can manage all aspects of the Azure Information Protection product.	...

Differences between Azure roles and Azure AD roles

At a high level, Azure roles control permissions to manage Azure resources, while Azure AD roles control permissions to manage Azure AD resources. The following table compares some of the differences.

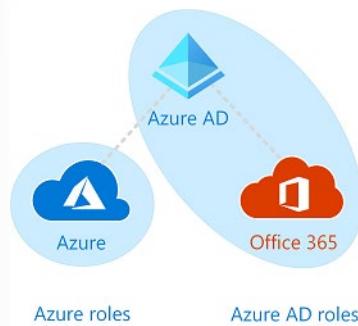
Azure roles	Azure AD roles
Manage access to Azure resources	Manage access to Azure AD resources
Supports custom roles	Supports custom roles
Scope can be specified at multiple levels (management group, subscription, resource group, resource)	Scope is at the tenant level
Role information can be accessed in Azure portal, Azure CLI, Azure PowerShell, Azure Resource Manager templates, REST API	Role information can be accessed in Azure admin portal, Microsoft 365 admin center, Microsoft Graph, Azure AD PowerShell

Do Azure roles and Azure AD roles overlap?

By default, Azure roles and Azure AD roles do not span Azure and Azure AD. However, if a Global Administrator elevates their access by choosing the **Access management for Azure resources** switch in the Azure portal, the Global Administrator will be granted the User Access Administrator role (an Azure role) on all subscriptions for a particular tenant. The User Access Administrator role enables the user to grant other users access to Azure resources. This switch can be helpful to regain access to a subscription.

Several Azure AD roles span Azure AD and Microsoft 365, such as the Global Administrator and User Administrator roles. For example, if you are a member of the Global Administrator role, you have global administrator capabilities in Azure AD and Microsoft 365, such as making changes to Microsoft Exchange

and Microsoft SharePoint. However, by default, the Global Administrator doesn't have access to Azure resources.



Assign roles

There are multiple ways to assign roles within Azure AD. You need to pick the one the best meets your needs. Note that the user interview might be slightly different for most of the configuration data is the same. Methods for assigning roles include:

- Assign a role to a user or group
 - **Azure AD** → **Roles and administration** → **Select a role** → **+ Add Assignment**
 - Assign a user or group to a role
 - **Azure AD** → **Open Users** (or Groups) → Select an **User** (or group) → **Assigned roles** → **+ Add Assignment**
 - Assign a role to a broad-scope, like a Subscription, Resource Group, or Management Group
 - Done via the **Access control (IAM)** within each blade
 - Assign a role using PowerShell or Microsoft Graph API
 - Assign a role using Privileged Identity Management (PIM)

Each of these methods can be used, but care must be taken as there are no built in restrictions. You could good to assign an administrative role to a group with guest users. Proper identity governance is the key.

Example - using PIM to assign a role

A common way to assign Azure AD roles to a user is on the Assigned roles page for a user. You can also configure the user eligibility to be elevated just-in-time into a role using **Privileged Identity Management (PIM)**.

Note - If you have an Azure AD Premium P2 license plan and already use PIM, all role management tasks are performed in the Privileged Identity Management experience. This feature is currently limited to assigning only one role at a time. You can't currently select multiple roles and assign them to a user all at once.

Name	Principal name	Type	Scope	Membership	Start time	End time	Action
Lisha Daher	lisha@fimdev.net	User	Directory	Direct	11/18/2019, 7:07:32 PM	Permanent	Remove Update Extend
Irene van der Merwe	irene@fimdev.net	User	Directory	Direct	2/13/2020, 1:34:33 PM	Permanent	Remove Update Extend
Rabeh Zaher	rabeh_microsoft_com#EXT#@fi	User	Directory	Direct	5/6/2020, 1:53:03 PM	Permanent	Remove Update Extend
Helene Botha	helene@fimdev.net	User	Directory	Direct	1/6/2020, 4:45:51 PM	Permanent	Remove Update Extend
Dritan Kodra	dritan@fimdev.net	User	Directory	Direct	2/13/2020, 1:34:46 PM	Permanent	Remove Update Extend

Create and assign a custom role in Azure Active Directory

This section describes how to create new custom roles in Azure AD. For the basics of custom roles, see the [custom roles overview](#)¹. The role can be assigned either at the directory-level scope or an app registration resource scope only.

Custom roles can be created in the **Roles and administrators**² tab on the Azure AD overview page.

Create a new custom role to grant access to manage app registrations

1. Sign in to the [Azure AD admin center](#)³ with Privileged role administrator or Global administrator permissions in the Azure AD organization.
2. Select **Azure Active Directory > Roles and administrators > New custom role**.

¹ <https://docs.microsoft.com/azure/active-directory/roles/custom-overview>

² <https://portal.azure.com/>

³ <https://aad.portal.azure.com/>

The screenshot shows the Microsoft Azure 'Roles and administrators (Preview)' page. At the top, there's a search bar and a 'New custom role' button, which is highlighted with a red box. Below the header, there's a message about PIM (just-in-time access) and a section titled 'Your Role: Global administrator and 2 other roles'. The main area is titled 'Administrative roles' with a sub-note about granting access to Azure AD and other services. A table lists various administrative roles with their descriptions:

Role	Description
<input type="checkbox"/> App_access_manager	Can manage API
<input type="checkbox"/> Application administrator	Can create and
<input type="checkbox"/> Application developer	Can create appli
<input type="checkbox"/> Application Support Administrator	
<input type="checkbox"/> Authentication administrator	Has access to vi
<input type="checkbox"/> Azure DevOps administrator	Can manage Azi

3. On the **Basics** tab, provide a name and description for the role and then click **Next**.

The screenshot shows the Microsoft Azure portal interface for creating a new custom role. The left sidebar contains various service icons. The main header says "Microsoft Azure" and "Search resources, services, and docs". The breadcrumb navigation shows "Home > Contoso - Roles and administrators > New custom role". The title "New custom role" is displayed above a feedback message: "Got a second? We would love your feedback on role creation →". Below this are three tabs: "Basics" (highlighted with a red box), "Permissions", and "Review + create". A note states: "Roles created here will be available for assignment on other resources as well. [Learn more](#)". The "Name" field is marked with an asterisk and has a placeholder "Name". The "Description" field is empty. Under "Baseline permissions", there are two radio buttons: "Start from scratch" (selected) and "Clone from a custom role". At the bottom, a "Next" button is highlighted with a red box.

4. On the **Permissions** tab, select the permissions necessary to manage basic properties and credential properties of app registrations.
5. First, enter “credentials” in the search bar and select the `microsoft.directory/applications/credentials/update` permission.

The screenshot shows the Microsoft Azure portal interface for creating a new custom role. The left sidebar has a dark theme with various icons. The main header says "Microsoft Azure" and "Search resources, services, and docs". The breadcrumb navigation shows "Home > Contoso - Roles and administrators > New custom role". The title is "New custom role" with a "All roles" link. A feedback message says "Got a second? We would love your feedback on role creation →". Below it, tabs for "Basics", "Permissions", and "Review + create" are shown, with "Permissions" being the active tab. A note says "Add permissions that should be included in the role. Permissions grant the ability to perform specific tasks. Not all Azure AD tasks are supported yet. You must add at least one permission." A search bar shows "credentials". A table lists permissions under the "PERMISSION" column and descriptions in the "DESCRIPTION" column. One row is selected: "microsoft.directory/applications/credentials/update" with the description "Update the certificates and client secrets on single-directory applications." Another row is shown: "microsoft.directory/applications/credentials/update" with the description "Update applications.credentials property in Azure Active Directory." At the bottom, there are "Previous" and "Next" buttons, with "Next" being highlighted by a red box.

6. Next, enter “basic” in the search bar, select the `microsoft.directory/applications/basic/update` permission, and then click **Next**.
7. On the **Review + create** tab, review the permissions and select **Create**.

Your custom role will show up in the list of available roles to assign.

Configure and Manage Custom Domains

A domain name is a part of the identifier for many Azure Active Directory (Azure AD) resources: it's part of a user name or email address for a user, part of the address for a group, and is sometimes part of the app ID URI for an application. A resource in Azure AD can include a domain name that's owned by the organization that contains the resource. Only a Global Administrator can manage domains in Azure AD.

Set the primary domain name for your Azure AD organization

When your organization is created, the initial domain name, such as **contoso.onmicrosoft.com**, is also the primary domain name. The primary domain is the default domain name for a new user when you create a new user. Setting a primary domain name streamlines the process for an administrator to create new users in the portal. To change the primary domain name:

1. Sign in to the **Azure portal**⁴ with an account that's a Global Administrator for the organization.
2. Select **Azure Active Directory**.
3. Select **Custom domain names**.

⁴ <https://portal.azure.com/>

The screenshot shows the 'fourcoffee - Custom domain names' page in the Azure Active Directory. The left sidebar lists various management options like Overview, Quick start, and Enterprise applications. The 'Custom domain names' option is highlighted with a red box. At the top, there's a 'Add custom domain' button, which is also highlighted with a red box. The main area displays two custom domains: 'fourcoffee.com' (Status: Unverified) and 'fourcoffee.onmicrosoft.com' (Status: Available).

4. Select the name of the domain that you want to be the primary domain.
5. Select the **Make primary** command. Confirm your choice when prompted.

The screenshot shows the 'fourcoffee.com - Custom domain names' page in the Microsoft Azure portal. The 'Make primary' checkbox is checked and highlighted with a red box. A confirmation dialog box is open, asking 'Do you want to make fourcoffee.com your primary domain?'. Below the dialog, there are two buttons: 'Yes' and 'No'. A tooltip provides instructions for configuring federated sign-on. The page also shows other details like 'PRIMARY DOMAIN' and 'IN USE'.

You can change the primary domain name for your organization to be any verified custom domain that isn't federated. Changing the primary domain for your organization won't change the user name for any existing users.

Add custom domain names to your Azure AD organization

You can add up to 900 managed domain names. If you're configuring all your domains for federation with on-premises Active Directory, you can add up to 450 domain names in each organization.

Add subdomains of a custom domain

If you want to add a subdomain name such as **europe.contoso.com** to your organization, you should first add and verify the root domain, such as contoso.com. The subdomain is automatically verified by Azure AD. To see that the subdomain you added is verified, refresh the domain list in the browser.

If you have already added a contoso.com domain to one Azure AD organization, you can also verify the subdomain europe.contoso.com in a different Azure AD organization. When adding the subdomain, you are prompted to add a TXT record in the DNS hosting provider.

What to do if you change the DNS registrar for your custom domain name

If you change the DNS registrars, there are no additional configuration tasks in Azure AD. You can continue using the domain name with Azure AD without interruption. If you use your custom domain name with Microsoft 365, Intune, or other services that rely on custom domain names in Azure AD, see the documentation for those services.

Delete a custom domain name

You can delete a custom domain name from your Azure AD if your organization no longer uses that domain name, or if you need to use that domain name with another Azure AD.

To delete a custom domain name, you must first ensure that no resources in your organization rely on the domain name. You can't delete a domain name from your organization if:

- Any user has a user name, email address, or proxy address that includes the domain name.
- Any group has an email address or proxy address that includes the domain name.
- Any application in your Azure AD has an app ID URI that includes the domain name.

You must change or delete any such resource in your Azure AD organization before you can delete the custom domain name.

ForceDelete option

You can **ForceDelete** a domain name in the Azure AD Admin Center or using Microsoft Graph API. These options use an asynchronous operation and update all references from the custom domain name like **user@contoso.com** to the initial default domain name such as **user@contoso.onmicrosoft.com**.

To call **ForceDelete** in the Azure portal, you must ensure that there are fewer than 1000 references to the domain name, and any references where Exchange is the provisioning service must be updated or removed in the Exchange Admin Center. This includes Exchange Mail-Enabled Security Groups and distributed lists. Also, the **ForceDelete** operation won't succeed if either of the following is true:

- You purchased a domain via Microsoft 365 domain subscription services
- You are a partner administering on behalf of another customer organization

The following actions are performed as part of the **ForceDelete** operation:

- Renames the UPN, EmailAddress, and ProxyAddress of users with references to the custom domain name to the initial default domain name.
- Renames the EmailAddress of groups with references to the custom domain name to the initial default domain name.

- Renames the identifierUris of applications with references to the custom domain name to the initial default domain name.

An error is returned when:

- The number of objects to be renamed is greater than 1000
- One of the applications to be renamed is a multi-tenant app

Add your custom domain name with the Azure Active Directory portal

Every new Azure AD tenant comes with an initial domain name, *domainname.onmicrosoft.com*. You can't change or delete the initial domain name, but you can add your organization's names. Adding custom domain names helps you to create user names that are familiar to your users, such as *chrisg@contoso.com*.

Before you begin

Before you can add a custom domain name, create your domain name with a domain registrar. For an accredited domain registrar, see **ICANN-Accredited Registrars**⁵.

Create your directory in Azure AD

After you get your domain name, you can create your first Azure AD directory. Sign into the Azure portal for your directory, using an account with the **Owner** role for the subscription, to create your new directory.

Important - The person who creates the tenant is automatically the Global administrator for that tenant. The Global administrator can add additional administrators to the tenant.

Tip - If you plan to federate your on-premises Windows Server AD with Azure AD, then you need to select I plan to configure this domain for single sign-on with my local Active Directory when you run the Azure AD Connect tool to synchronize your directories.

You also need to register the same domain name you select for federating with your on-premises directory in the Azure AD Domain step in the wizard. To see what that setup looks like, see **Verify the Azure AD domain selected for federation**⁶. If you don't have the Azure AD Connect tool, you can [download it here](#)⁷.

Configure and Manage Device Registration

With the proliferation of devices of all shapes and sizes and the bring your own device (BYOD) concept, IT professionals are faced with two somewhat opposing goals:

- Allow end users to be productive wherever and whenever
- Protect the organization's assets

To protect these assets, IT staff need to first manage the device identities. IT staff can build on the device identity with tools like Microsoft Intune to ensure standards for security and compliance are met. Azure

⁵ <https://www.icann.org/registrar-reports/accredited-list.html>

⁶ <https://docs.microsoft.com/azure/active-directory/hybrid/how-to-connect-install-custom>

⁷ <https://go.microsoft.com/fwlink/?LinkId=615771>

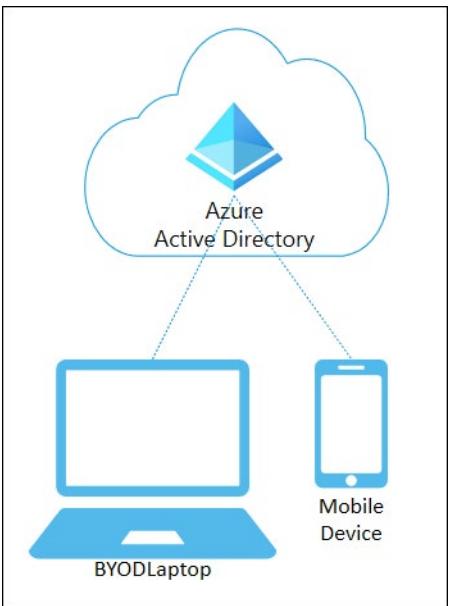
Active Directory (Azure AD) enables single sign-on to devices, apps, and services from anywhere through these devices.

- Your users get access to your organization's assets they need.
- Your IT staff get the controls they need to secure your organization.

Azure AD registered devices

The goal of Azure AD registered devices is to provide your users with support for the BYOD or mobile device scenarios. In these scenarios, a user can access your organization's Azure Active Directory controlled resources using a personal device.

Azure AD registered	Description
Definition	Registered to Azure AD without requiring organizational account to sign in to the device
Primary audience	Applicable to all users with the following criteria: Bring your own device (BYOD) -and- Mobile devices
Device ownership	User or Organization
Operating systems	Windows 10, iOS, Android, and macOS
Provisioning	Windows 10 / Settings -or- iOS/Android Company Portal -or- Microsoft Authenticator app -or- macOS Company Portal
Device sign in options	End-user local credentials -and- Password -and- Windows Hello -and- PIN -and- Biometrics or Pattern for other devices
Device management	Mobile Device Management (example: Microsoft Intune) -and- Mobile Application Management
Key capabilities	SSO to cloud resources -and- Conditional Access when enrolled into Intune -and- Conditional Access via App protection policy -and- Enables Phone sign in with Microsoft Authenticator app



Azure AD registered devices are signed in to using a local account like a Microsoft account on a Windows 10 device, but additionally have an Azure AD account attached for access to organizational resources. Access to resources in the organization can be further limited based on that Azure AD account and Conditional Access policies applied to the device identity.

Administrators can secure and further control these Azure AD registered devices using Mobile Device Management (MDM) tools like Microsoft Intune. MDM provides a means to enforce organization-required configurations like requiring storage to be encrypted, password complexity, and security software kept updated.

Azure AD registration can be accomplished when accessing a work application for the first time or manually using the Windows 10 Settings menu.

Scenarios

A user in your organization wants to access tools for email, reporting time-off, and benefits enrollment from their home PC. Your organization has these tools behind a Conditional Access policy that requires access from an Intune compliant device. The user adds their organization account and registers their home PC with Azure AD and the required Intune policies are enforced giving the user access to their resources.

Another user wants to access their organizational email on their personal Android phone that has been rooted. Your company requires a compliant device and has created an Intune compliance policy to block any rooted devices. The employee is stopped from accessing organizational resources on this device.

Azure AD joined devices

Azure AD join is intended for organizations that want to be cloud-first or cloud-only. Any organization can deploy Azure AD joined devices no matter the size or industry. Azure AD join works even in a environment, enabling access to both cloud and on-premises apps and resources.

Azure AD joined	Description
Definition	Joined only to Azure AD requiring organizational account to sign in to the device
Primary audience	Suitable for both cloud-only and hybrid organizations -or- Applicable to all users in an organization
Device ownership	Organization
Operating systems	All Windows 10 devices except Windows 10 Home -or- Windows Server 2019 Virtual Machines running in Azure (Server core is not supported)
Provisioning	Self-service: Windows OOBE or Settings -or- Bulk enrollment -or- Windows Autopilot
Device sign in options	Organizational accounts using: Password -or- Windows Hello for Business -or- FIDO2.0 security keys (preview)
Device management	Mobile Device Management (example: Microsoft Intune) -or- Co-management with Microsoft Intune and Microsoft Endpoint Configuration Manager
Key capabilities	SSO to both cloud and on-premises resources -or- Conditional Access through MDM enrollment and MDM compliance evaluation -or- Self-service Password Reset and Windows Hello PIN reset on lock screen -or- Enterprise State Roaming across devices

Azure AD joined devices are signed in to using an organizational Azure AD account. Access to resources in the organization can be further limited based on that Azure AD account and Conditional Access policies applied to the device identity.

Administrators can secure and further control Azure AD joined devices using Mobile Device Management (MDM) tools like Microsoft Intune or in co-management scenarios using Microsoft Endpoint Configuration Manager. These tools provide a means to enforce organization-required configurations like requiring storage to be encrypted, password complexity, software installations, and software updates. Administrators can make organization applications available to Azure AD joined devices using Configuration Manager.

Azure AD join can be accomplished using self-service options like the Out of Box Experience (OOBE), bulk enrollment, or Windows Autopilot.

Azure AD joined devices can still maintain single sign-on access to on-premises resources when they are on the organization's network. Devices that are Azure AD joined can still authenticate to on-premises servers like file, print, and other applications.

Scenarios

Although Azure AD join is primarily intended for organizations that do not have an on-premises Windows Server Active Directory infrastructure, you can certainly use it in scenarios where:

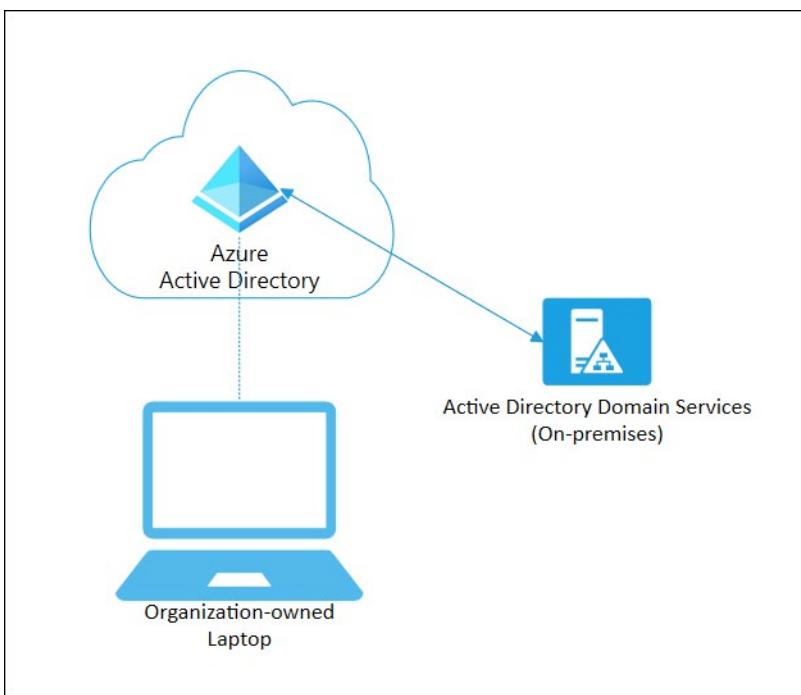
- You want to transition to cloud-based infrastructure using Azure AD and MDM like Intune.
- You can't use an on-premises domain join, for example, if you need to get mobile devices such as tablets and phones under control.

- Your users primarily need to access Microsoft 365 or other SaaS apps integrated with Azure AD.
- You want to manage a group of users in Azure AD instead of in Active Directory. This scenario can apply, for example, to seasonal workers, contractors, or students.
- You want to provide joining capabilities to workers in remote branch offices with limited on-premises infrastructure.

You can configure Azure AD joined devices for all Windows 10 devices with the exception of Windows 10 Home.

The goal of Azure AD joined devices is to simplify:

- Windows deployments of work-owned devices
- Access to organizational apps and resources from any Windows device
- Cloud-based management of work-owned devices
- Users to sign in to their devices with their Azure AD or synced Active Directory work or school accounts.



Azure AD Join can be deployed by using a number of different methods.

Hybrid Azure AD joined devices

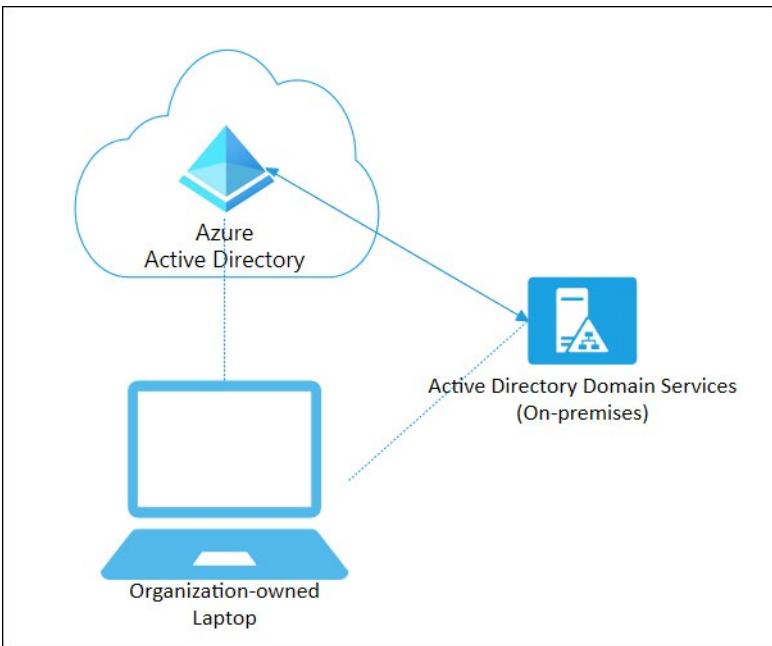
For more than a decade, many organizations have used the domain join to their on-premises Active Directory to enable:

- IT departments to manage work-owned devices from a central location.
- Users to sign in to their devices with their Active Directory work or school accounts.

Typically, organizations with an on-premises footprint rely on imaging methods to provision devices, and they often use **Configuration Manager** or **group policy (GP)** to manage them.

If your environment has an on-premises AD footprint and you also want benefit from the capabilities provided by Azure Active Directory, you can implement hybrid Azure AD joined devices. These devices are devices that are joined to your on-premises Active Directory and registered with your Azure Active Directory.

Hybrid Azure AD joined	Description
Definition	Joined to on-premises AD and Azure AD requiring organizational account to sign in to the device
Primary audience	Suitable for hybrid organizations with existing on-premises AD infrastructure -and- Applicable to all users in an organization
Device ownership	Organization
Operating systems	Windows 10, 8.1 and 7 -or- Windows Server 2008/R2, 2012/R2, 2016 and 2019
Provisioning	Windows 10, Windows Server 2016/2019 -or- Domain join by IT and autojoin via Azure AD Connect or ADFS config -or- Domain join by Windows Autopilot and autojoin via Azure AD Connect or ADFS config -or- Windows 8.1, Windows 7, Windows Server 2012 R2, Windows Server 2012, and Windows Server 2008 R2 - Require MSI
Device sign in options	Organizational accounts using: Password -and- Windows Hello for Business for Win10
Device management	Group Policy -and- Configuration Manager standalone -or- co-management with Microsoft Intune
Key capabilities	SSO to both cloud and on-premises resources -and- Conditional Access through Domain join or through Intune if co-managed -and- Self-service Password Reset and Windows Hello PIN reset on lock screen -and- Enterprise State Roaming across devices



Scenarios

Use Azure AD hybrid joined devices if:

- You have Win32 apps deployed to these devices that rely on Active Directory machine authentication.
- You want to continue to use Group Policy to manage device configuration.
- You want to continue to use existing imaging solutions to deploy and configure devices.
- You must support down-level Windows 7 and 8.1 devices in addition to Windows 10.

Configure Delegation by using Administrative Units

Administrative units are Azure Active Directory (Azure AD) resources that can be containers for other Azure AD resources. An administrative unit can contain only users and groups.

Administrative units restrict permissions in a role to any portion of your organization that you define. You could, for example, use administrative units to delegate the Helpdesk Administrator role to regional support specialists, so they can manage users only in the region that they support.

You can manage administrative units by using the Azure portal, PowerShell cmdlets and scripts, or Microsoft Graph.

Plan your administrative units

You can use administrative units to logically group Azure AD resources. An organization whose IT department is scattered globally might create administrative units that define relevant geographical boundaries. In another scenario, where a global organization has suborganizations that are semi-autonomous in their operations, administrative units could represent the suborganizations.

The criteria on which administrative units are created are guided by the unique requirements of an organization. Administrative units are a common way to define structure across Microsoft 365 services. We recommend that you prepare your administrative units with their use across Microsoft 365 services in mind. You can get maximum value out of administrative units when you can associate common resources across Microsoft 365 under an administrative unit.

You can expect the creation of administrative units in the organization to go through the following stages:

1. **Initial adoption:** Your organization will start creating administrative units based on initial criteria, and the number of administrative units will increase as the criteria are refined.
2. **Pruning:** After the criteria are defined, administrative units that are no longer required will be deleted.
3. **Stabilization:** Your organizational structure is defined, and the number of administrative units isn't going to change significantly in the short term.

Delegate administration in Azure Active Directory

With organizational growth comes complexity. One common response is to reduce some of the workload of access management with Azure Active Directory (AD) admin roles. You can assign the least possible privilege to users to access their apps and perform their tasks. Even if you don't assign the Global Administrator role to every application owner, you're placing application management responsibilities on the existing Global Administrators. There are many reasons for an organization move toward a more decentralized administration.

In Azure AD, you can delegate Application creation and management permissions in the following ways:

- Restricting who can create applications and manage the applications they create. By default in Azure AD, all users can register application registrations and manage all aspects of applications they create. This can be restricted to only allow selected people that permission.
- Assigning one or more owners to an application. This is a simple way to grant someone the ability to manage all aspects of Azure AD configuration for a specific application.
- Assigning a built-in administrative role that grants access to manage configuration in Azure AD for all applications. This is the recommended way to grant IT experts access to manage broad application configuration permissions without granting access to manage other parts of Azure AD not related to application configuration.
- Creating a custom role defining very specific permissions and assigning it to someone either to the scope of a single application as a limited owner, or at the directory scope (all applications) as a limited administrator.

When granting access use one of the above methods for two reasons. First, delegating the ability to perform administrative tasks reduces global administrator overhead. Second, using limited permissions improves your security posture and reduces the potential for unauthorized access.

Plan for Delegation

It's work to develop a delegation model that fits your needs. Developing a delegation model is an iterative design process, and we suggest you follow these steps:

- Define the roles you need
- Delegate app administration
- Grant the ability to register applications

- Delegate app ownership
- Develop a security plan
- Establish emergency accounts
- Secure your administrator roles
- Make privileged elevation temporary

Define roles

Determine the Active Directory tasks that are carried out by administrators and how they map to roles. Each task should be evaluated for frequency, importance, and difficulty. These criteria are vital aspects of task definition because they govern whether a permission should be delegated:

- Tasks that you do routinely, have limited risk, and are trivial to complete are excellent candidates for delegation.
- Tasks that you do rarely but have great impact across the organization and require high skill levels should be considered very carefully before delegating. Instead, you can temporarily elevate an account to the required role or reassign the task.

Delegate app administration

The proliferation of apps within your organization can strain your delegation model. If it places the burden for application access management on the Global Administrator, it's likely that model increases its overhead as time goes on. If you have granted people the Global Administrator role for things like configuring enterprise applications, you can now offload them to the following less-privileged roles. Doing so helps to improve your security posture and reduces the potential for unfortunate mistakes. The most-privileged application administrator roles are:

- The **Application Administrator** role, which grants the ability to manage all applications in the directory, including registrations, single sign-on settings, user and group assignments and licensing, Application Proxy settings, and consent. It doesn't grant the ability to manage Conditional Access.
- The **Cloud Application Administrator** role, which grants all the abilities of the Application Administrator, except it doesn't grant access to Application Proxy settings (because it has no on-premises permission).

Delegate app registration

By default, all users can create application registrations. To selectively grant the ability to create application registrations:

- Set **Users can register applications** to No in **User settings**
- Assign the user to the Application Developer role

To selectively grant the ability to consent to allow an application to access data:

- Set **Users can consent to applications accessing company data on their behalf** To No in **User settings**
- Assign the user to the Application Developer role

When an Application Developer creates a new application registration, they are automatically added as the first owner.

Delegate app ownership

For even finer-grained app access delegation, you can assign ownership to individual enterprise applications. This complements the existing support for assigning application registration owners. Ownership is assigned on a per-enterprise application basis in the Enterprise Applications blade. The benefit is owners can manage only the enterprise applications they own. For example, you can assign an owner for the Salesforce application, and that owner can manage access to and configuration for Salesforce, and no other applications. An enterprise application can have many owners, and a user can be the owner for many enterprise applications. There are two app owner roles:

- The **Enterprise Application Owner** role grants the ability to manage the 'enterprise applications that the user owns, including single sign-on settings, user and group assignments, and adding additional owners. It doesn't grant the ability to manage Application Proxy settings or Conditional Access.
- The **Application Registration Owner** role grants the ability to manage application registrations for app that the user owns, including the application manifest and adding additional owners.

Develop a security plan

Azure AD provides an extensive guide to planning and executing a security plan on your Azure AD admin roles, **Securing privileged access for hybrid and cloud deployments⁸**.

Establish emergency accounts

To maintain access to your identity management store when issue arises, prepare emergency access accounts according to **Create emergency-access administrative accounts⁹**.

Secure your administrator roles

Attackers who get control of privileged accounts can do tremendous damage, so protect these accounts first, using the **baseline access policy¹⁰** that is available by default to all Azure AD organizations (in public preview). The policy enforces multi-factor authentication on privileged Azure AD accounts. The following Azure AD roles are covered by the Azure AD baseline policy:

- Global administrator
- SharePoint administrator
- Exchange administrator
- Conditional Access administrator
- Security administrator

Configure Tenant Wide Settings

Tenant-wide setting, are the configuration options that apply to all resources within your tenant as the name implies. These tenant wide options are set in specific places, to control the look, feel, and configuration of your tenant and its members.

⁸ <https://docs.microsoft.com/azure/active-directory/roles/security-planning>

⁹ <https://docs.microsoft.com/azure/active-directory/roles/security-emergency-access>

¹⁰ <https://cloudblogs.microsoft.com/enterprisemobility/2018/06/22/baseline-security-policy-for-azure-ad-admin-accounts-in-public-preview/>

Tenant-wide option

- **Tenant Properties**

- Azure AD → Properties
- Where you give the name of your directory and set values like the primary contact

- **User Settings**

- Azure AD → Users → User Settings
- Where you define what global rights your users have, like registering applications

- **External Collaboration Settings**

- Azure AD → Users → User Settings → Manage external collaboration
- Where you define what task an external guest user can perform like inviting more guest users

Configure tenant-wide user settings

The screenshot shows the 'Default Directory | User settings' page in the Azure Active Directory portal. The left sidebar has a list of options: External Identities, Roles and administrators, Administrative units, Enterprise applications, Devices, App registrations, Identity Governance, Application proxy, Licenses, Azure AD Connect, Custom domain names, Mobility (MDM and MAM), Password reset, Company branding, User settings (which is selected and highlighted in grey), Properties, and Security. The main content area has several sections: 'Enterprise applications' (Manage how end users launch and view their applications), 'App registrations' (Users can register applications with options Yes, Selected group, or No), 'Administration portal' (Restrict access to Azure AD administration portal with options Yes, Selected group, or No), 'LinkedIn account connections' (Allow users to connect their work or school account with LinkedIn. Data sharing between Microsoft and LinkedIn is not enabled until you Learn more about LinkedIn account connections with options Yes, Selected group, or No), and 'External users' (Manage external collaboration settings). There are 'Save' and 'Discard' buttons at the top.

In Azure Active Directory (Azure AD), all users are granted a set of default permissions. A user's access consists of the type of user, their role assignments, and their ownership of individual objects. The default user permissions can be changed only in user settings in Azure AD.

Member and guest users

The set of default permissions received depends on whether the user is a native member of the tenant (member user) or if the user is brought over from another directory as a B2B collaboration guest (guest user).

- Member users can register applications, manage their own profile photo and mobile phone number, change their own password, and invite B2B guests. In addition, users can read all directory information (with a few exceptions).

- Guest users have restricted directory permissions. They can manage their own profile, change their own password, and retrieve some information about other users, groups, and apps; however, they cannot read all directory information. For example, guest users cannot enumerate the list of all users, groups, and other directory objects. Guests can be added to administrator roles, which grant them full read and write permissions contained in the role. Guests can also invite other guests.

The following default permissions for member users can be restricted in the following ways:

Permission	Setting explanation
Users can register application	By default, member users can register applications. Setting this option to No prevents users from creating application registrations. The ability can then be granted back to specific individuals by adding them to the Application Developer role.
Restrict access to Azure AD administration portal	Setting this option to No lets non-administrators use the Azure AD administration portal to read and manage Azure AD resources. Yet, restricts all non-administrators from accessing any Azure AD data in the administration portal. This setting does not restrict access to Azure AD data using PowerShell or other clients such as Visual Studio. When set to Yes, to grant a specific non-admin user the ability to use the Azure AD administration portal assign any administrative role such as the Directory Readers role. This role allows reading basic directory information, which member users have by default (guests and service principals do not).

Sign in with LinkedIn

With more than 500 million members worldwide, LinkedIn is the largest and most trusted source of professional identities. Leverage this power to enhance the sign-in experience of your sites and applications.

Use sign in with LinkedIn to:

- Reduce friction and obtain more sign-ups by allowing members to Sign In with LinkedIn, without having the need to create a new account.
- Minimize the costs and time associated with implementing your own login, identity, profile management, and password management.
- Personalize your sites and applications with the latest member profiles.

Manage security defaults

Managing security can be difficult with common identity-related attacks like password spray, replay, and phishing becoming more and more popular. Security defaults make it easier to help protect your organization from these attacks with preconfigured security settings:

- Requiring all users to register for Azure AD Multi-Factor Authentication.
- Requiring administrators to perform multi-factor authentication.
- Blocking legacy authentication protocols.

- Requiring users to perform multi-factor authentication when necessary.
- Protecting privileged activities like access to the Azure portal.

Availability

Microsoft is making security defaults available to everyone. The goal is to ensure that all organizations have a basic level of security enabled at no extra cost.

Configure the external user options

Home > Default Directory >

External collaboration settings ...

Save Discard

>Email one-time passcode for guests has been moved to All Identity Providers. →

Guest user access

Guest user access restrictions ⓘ

Learn more

Guest users have the same access as members (most inclusive)

Guest users have limited access to properties and memberships of directory objects

Guest user access is restricted to properties and memberships of their own directory objects (most restrictive)

Guest invite settings

Guest invite restrictions ⓘ

Learn more

Anyone in the organization can invite guest users including guests and non-admins (most inclusive)

Member users and users assigned to specific admin roles can invite guest users including guests with member permissions

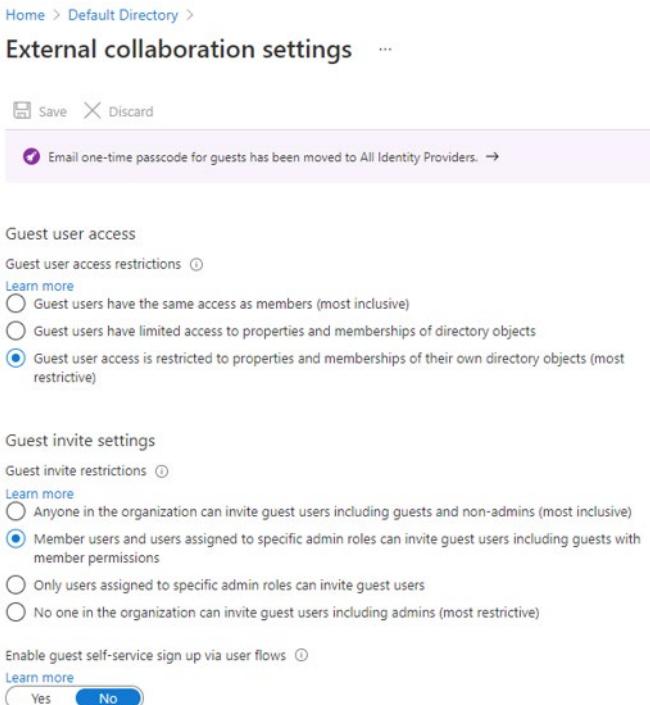
Only users assigned to specific admin roles can invite guest users

No one in the organization can invite guest users including admins (most restrictive)

Enable guest self-service sign up via user flows ⓘ

Learn more

Yes No



Here you configure the actions that external users can take while using the cloud resources of your tenant.

- **Guest user access** - Guest users can be given rights to where they operate almost as a full user, to restriction where they can only look at their own content.
- **Guest invite settings** - Who can invite guests to join the organization; from guest themselves to only admins.
- **Guest self-service up** - Allow guest to partake in self-service options for users.

Configure tenant properties for the directory

Home > Default Directory

Default Directory | Properties ...

Azure Active Directory

External Identities
Roles and administrators
Administrative units
Enterprise applications
Devices
App registrations
Identity Governance
Application proxy
Licenses
Azure AD Connect
Custom domain names
Mobility (MDM and MAM)
Password reset
Company branding
User settings
Properties
Security

Save Discard

Tenant properties

Name *
Default Directory *

Country or region
United States

Location
United States datacenters

Notification language
English

Tenant ID
[empty]

Technical contact
[empty]

Global privacy contact
[empty]

Privacy statement URL
[empty]

Set the basic values the define the look at feel of your tenant within Azure AD.

- **Name** - friendly name for your tenant, for use in the Azure portal
- **Country or region** - location of your primary company and the Azure datacenters being used
- **Notification language** - language used for sending notifications and alerts
- **Tenant ID** - unique identifier for your tenant to be used programmatically, as needed
- **Technical contact** - primary contact for the tenant, defaults to the user who created the tenant
- **Global privacy contact** - user or alias to content for privacy topics
- **Privacy statement URL** - link to a PDF or webpage containing the privacy rules for your cloud solutions

Create, Configure, and Manage Identities

Introduction

Transitioning workloads to the cloud involves more than just moving servers, websites, and data. Companies need to think about how to secure those resources, by defining authorized users. Next, companies need to ensure that users only have access to data that they need, that they only create services they are authorized to create, and that they only perform operations they are authorized to perform. Access to cloud-based workloads is controlled centrally in two ways. First by providing a definitive identity for each user that they use for every service. Then second by ensuring employees and vendors have enough access to do their jobs.

Azure helps to make these sorts of challenges easier to solve with Azure Active Directory (Azure AD), the Microsoft cloud-based identity and access management service. Azure AD provides single sign-on and multi-factor authentication to help protect your users and your data. In this module, you will learn the basics of creating, configuring, and managing users and groups of users. You will also learn how to manage licenses.

Learning objectives

In this module, you will:

- Create, configure, and manage users
- Create, configure, and manage groups
- Manage licenses

Prerequisites

None

Users

Every user who needs access to Azure resources needs an Azure user account. A user account contains all the information needed to authenticate the user during the sign-on process. Once authenticated, Azure AD builds an access token to authorize the user and determine what resources they can access and what they can do with those resources.

You use the **Azure Active Directory** dashboard in the Azure portal to work with user objects. Keep in mind that you can only work with a single directory at a time - but you can use the **Directory + Subscription** panel to switch directories. The dashboard also has a **Switch directory** button in the toolbar which makes it easy to switch to another available directory.

View users

To view the Azure AD users, select the **Users** entry under the **Manage** group - this will open the **All Users** view. Take a minute to access the portal and view your users. Notice the **USER TYPE** and **SOURCE** columns, as the following figure depicts.

NAME	USER NAME	USER TYPE	SOURCE
CO Contoso	@microsoft.onmicrosoft.com	Member	Azure Active Directory
CO Contoso	@microsoft.onmicrosoft.com	Member	Azure Active Directory
CO Contoso	@microsoft.onmicrosoft.com	Member	Azure Active Directory
CO Contoso	@microsoft.onmicrosoft.com	Member	Azure Active Directory
CO Contoso	@microsoft.onmicrosoft.com	Member	Azure Active Directory
CO Contoso	@microsoft.onmicrosoft.com	Member	Azure Active Directory
Contoso		Guest	External Azure Active Directory
contoso-user		Guest	Invited user
CA Contoso Audit	@microsoft.com	Member	Windows Server AD

Typically, Azure AD defines users in three ways:

- **Cloud identities** - These users exist only in Azure AD. Examples are administrator accounts and users that you manage yourself. Their source is **Azure Active Directory** or **External Azure Active Directory** if the user is defined in another Azure AD instance but needs access to subscription resources controlled by this directory. When these accounts are removed from the primary directory, they are deleted.
- **Directory-synchronized identities** - These users exist in an on-premises Active Directory. A synchronization activity that occurs via **Azure AD Connect** brings these users in to Azure. Their source is **Windows Server AD**.
- **Guest users** - These users exist outside Azure. Examples are accounts from other cloud providers and Microsoft accounts such as an Xbox LIVE account. Their source is **Invited user**. This type of account is useful when external vendors or contractors need access to your Azure resources. Once their help is no longer necessary, you can remove the account and all of their access.

Groups

An Azure Active Directory (Azure AD) group helps organize users, which makes it easier to manage permissions. Using groups lets the resource owner (or Azure AD directory owner), assign a set of access permissions to all the members of the group, instead of having to provide the rights one-by-one. Groups allow us to define a security boundary and then add and remove specific users to grant or deny access with a minimum amount of effort. Even better, Azure AD supports the ability to define membership based on rules - such as what department a user works in, or the job title they have.

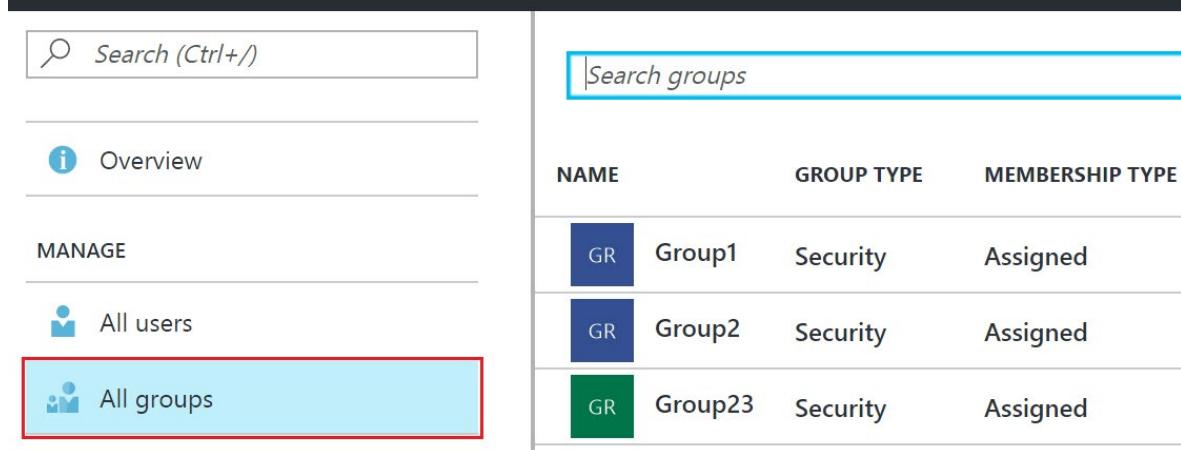
Azure AD allows you to define two different types of groups.

- **Security groups.** These are the most common and are used to manage member and computer access to shared resources for a group of users. For example, you can create a security group for a specific security policy. By doing it this way, you can give a set of permissions to all the members at once, instead of having to add permissions to each member individually. This option requires an Azure AD administrator.
- **Microsoft 365 groups.** These groups provide collaboration opportunities by giving members access to a shared mailbox, calendar, files, SharePoint site, and more. This option also lets you give people outside of your organization access to the group. This option is available to users as well as admins.

View available groups

You can view all groups through the **Groups** item under the **Manage** group from the Azure AD dashboard. A new Azure AD install won't have any groups defined.

Users and groups - All groups

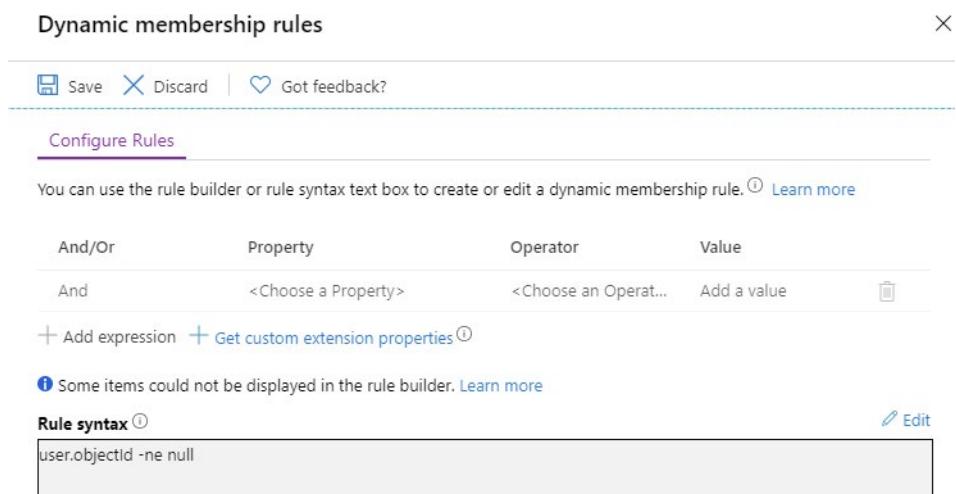


The screenshot shows the 'Users and groups' blade in the Azure portal. On the left, there's a search bar and a sidebar with 'Overview' and 'MANAGE' sections. Under 'MANAGE', there are links for 'All users' and 'All groups'. The 'All groups' link is highlighted with a red box. On the right, there's a table titled 'Search groups' with columns 'NAME', 'GROUP TYPE', and 'MEMBERSHIP TYPE'. Three groups are listed: 'Group1' (Security, Assigned), 'Group2' (Security, Assigned), and 'Group23' (Security, Assigned). The 'NAME' column uses blue squares for Group1 and Group2, and a green square for Group23.

NAME	GROUP TYPE	MEMBERSHIP TYPE
GR Group1	Security	Assigned
GR Group2	Security	Assigned
GR Group23	Security	Assigned

Dynamic groups

The final type of group is a dynamic group, which the name implies, the membership is generated by a formula each time the group is used. A dynamic distribution group includes any recipient in Active Directory with attribute values that match its filter. If a recipient's properties are modified to match the filter, the recipient could inadvertently become a group member and start receiving messages that are sent to the group. Well-defined, consistent account provisioning processes will reduce the chances of this issue occurring.



The screenshot shows the 'Dynamic membership rules' configuration page. It has a header with 'Dynamic membership rules' and a close button. Below the header are 'Save', 'Discard', and 'Got feedback?' buttons. A 'Configure Rules' section is shown with a note: 'You can use the rule builder or rule syntax text box to create or edit a dynamic membership rule.' There's a 'Learn more' link. The main area shows a table for building rules:

And/Or	Property	Operator	Value
And	<Choose a Property>	<Choose an Operator...>	Add a value <input type="button" value="Edit"/>

Below the table are buttons for '+ Add expression' and '+ Get custom extension properties'. A note says: 'Some items could not be displayed in the rule builder.' There's also a 'Rule syntax' section with a text input containing 'user.ObjectId -ne null' and an 'Edit' button.

This dynamic group would consist of all valid members of the Azure AD.

Manage Licenses

Microsoft paid cloud services, such as Microsoft 365, Enterprise Mobility + Security, Dynamics 365, and other similar products, require licenses. These licenses are assigned to each user who needs access to these services. To manage licenses, administrators use one of the management portals (Office or Azure) and PowerShell cmdlets. Azure Active Directory (Azure AD) is the underlying infrastructure that supports identity management for all Microsoft cloud services. Azure AD stores information about license assignment states for users.

Until now, licenses could only be assigned at the individual user level, which can make large-scale management difficult. For example, to add or remove user licenses based on organizational changes, such as users joining or leaving the organization or a department, an administrator often must write a complex PowerShell script. This script makes individual calls to the cloud service.

To address those challenges, Azure AD now includes group-based licensing. You can assign one or more product licenses to a group. Azure AD ensures that the licenses are assigned to all members of the group. Any new members who join the group are assigned the appropriate licenses. When they leave the group, those licenses are removed. This licensing management eliminates the need for automating license management via PowerShell to reflect changes in the organization and departmental structure on a per-user basis.

License requirements

You must have one of the following licenses to use group-based licensing:

- Paid or trial subscription for Azure AD Premium P1 and above
- Paid or trial edition of Office 365 Enterprise E3 or Office 365 A3 or Office 365 GCC G3 or Office 365 E3 for GCCH or Office 365 E3 for DOD and above

Required number of licenses

For any groups assigned a license, you must also have a license for each unique member. While you don't have to assign each member of the group a license, you must have at least enough licenses to include all of the members. For example, if you have 1,000 unique members who are part of licensed groups in your tenant, you must have at least 1,000 licenses to meet the licensing agreement.

Features

Here are the main features of group-based licensing:

- Licenses can be assigned to any security group in Azure AD. Security groups can be synced from on-premises, by using Azure AD Connect. You can also create security groups directly in Azure AD (also called cloud-only groups), or automatically via the Azure AD dynamic group feature.
- When a product license is assigned to a group, the administrator can disable one or more service plans in the product. Typically, this assignment is done when the organization is not yet ready to start using a service included in a product. For example, the administrator might assign Microsoft 365 to a department, but temporarily disable the Yammer service.
- All Microsoft cloud services that require user-level licensing are supported. This support includes all Microsoft 365 products, Enterprise Mobility + Security, and Dynamics 365.

- Group-based licensing is currently available only through the **Azure portal¹¹**.
- Azure AD automatically manages license modifications that result from group membership changes. Typically, license modifications are effective within minutes of a membership change.
- A user can be a member of multiple groups with license policies specified. A user can also have some licenses that were directly assigned, outside of any groups. The resulting user state is a combination of all assigned product and service licenses. If a user is assigned same license from multiple sources, the license will be consumed only once.
- In some cases, licenses cannot be assigned to a user. For example, there might not be enough available licenses in the tenant, or conflicting services might have been assigned at the same time. Administrators have access to information about users for whom Azure AD could not fully process group licenses. They can then take corrective action based on that information.

Some Microsoft services are not available in all locations. Before assigning a license to a user, the administrator should specify usage location in the User Profile.

For group license assignment, any users without a usage location specified inherit the location of the directory. If you have users in multiple locations, we recommend that you always set usage location as part of your user creation flow in Azure AD (for example, via AAD Connect configuration) - that ensures the result of license assignment is always correct and users do not receive services in locations that are not allowed.

¹¹ <https://portal.azure.com/>

Implement and Manage External Identities

Introduction

Being able to invite external users to use your Azure resources is a great benefit, but it needs to be done in a secure way. This module is designed to help you understand how to enable secure B2B collaboration scenarios with users outside your organization, including managing external collaboration settings in Azure Active Directory (Azure AD) and inviting users individually or in bulk. You will also learn about managing external user accounts and configuring identity providers.

Learning objectives

In this module, you will:

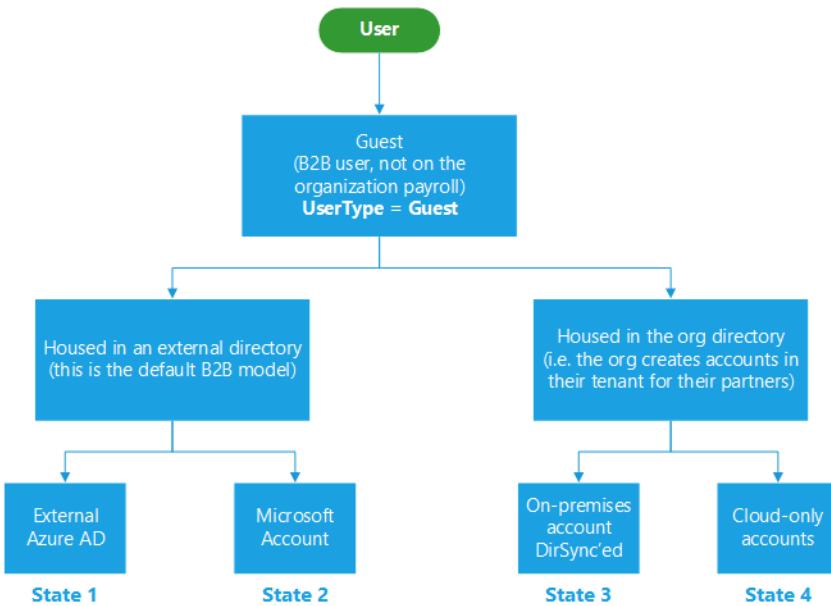
- Manage external collaboration settings in Azure AD
- Invite external users (individually or in bulk)
- Manage external user accounts in Azure AD
- Configure identity providers (social and SAML/WS-fed)

Prerequisites

None

Collaboration

Azure AD External Identities is a feature that makes it possible for you to allow people outside your organization to access your apps and resources, while letting them sign in using whatever identity they prefer. Your partners, distributors, suppliers, vendors, and other guest users can “bring their own identities.” Whether they have a corporate or government-issued digital identity, or an unmanaged social identity like Google or Facebook, they can use their own credentials to sign in. The external user’s identity provider manages their identity, and you manage access to your apps with Azure AD to keep your resources protected.



Depending on the inviting organization's needs, an Azure AD B2B collaboration user can be in one of the following account states:

- State 1: Homed in an external instance of Azure AD and represented as a guest user in the inviting organization. In this case, the B2B user signs in by using an Azure AD account that belongs to the invited tenant. If the partner organization doesn't use Azure AD, the guest user in Azure AD is still created. The requirements are that they redeem their invitation and Azure AD verifies their email address. This arrangement is also called a just-in-time (JIT) tenancy or a "viral" tenancy.
- State 2: Homed in a Microsoft or other account and represented as a guest user in the host organization. In this case, the guest user signs in with a Microsoft account or a social account. The invited user's identity is created as a Microsoft account in the inviting organization's directory during offer redemption.
- State 3: Homed in the host organization's on-premises Active Directory and synced with the host organization's Azure AD. You can use Azure AD Connect to sync the partner accounts to the cloud as Azure AD B2B users with UserType = Guest. See Grant locally-managed partner accounts access to cloud resources.
- State 4: Homed in the host organization's Azure AD with UserType = Guest and credentials that the host organization manages.

External identities scenarios

Azure AD External Identities focuses less on a user's relationship to your organization and more on how the user wants to sign in to your apps and resources. Within this framework, Azure AD supports a variety of scenarios.

A B2B collaboration scenario allows you to invite external users into your own tenant as "guest" users that you can assign permissions to (for authorization) while letting them use their existing credentials (for authentication). Users sign in to the shared resources using a simple invitation and redemption process with their work, school, or other email account. You can also use Azure AD entitlement management to configure policies that manage access for external users. And now with the availability of self-service sign-up user flows (preview), you can allow external users to sign up for applications themselves. The

experience can be customized to allow sign-up with a work, school, or social identity (such as Google or Facebook). You can also collect information about the user during the sign-up process.

The following list identifies an example B2B collaboration scenario and details some of the capabilities it provides:

- **Primary scenario** - Collaboration using Microsoft applications (Microsoft 365, Teams, and so on) or your own applications (SaaS apps, custom-developed apps, and so on).
- **Intended for** - Collaborating with business partners from external organizations like suppliers, partners, vendors. Users appear as guest users in your directory. These users may or may not have managed IT.
- **Identity providers supported** - External users can collaborate using work accounts, school accounts, any email address, SAML and WS-Fed based identity providers, Gmail, and Facebook.
- **External user management** - External users are managed in the same directory as employees, but are typically annotated as guest users. Guest users can be managed the same way as employees, added to the same groups, and so on.
- **Single sign-on (SSO)** - SSO to all Azure AD-connected apps is supported. For example, you can provide access to Microsoft 365 or on-premises apps, and to other SaaS apps such as Salesforce or Workday.
- **Security policy and compliance** - Managed by the host/inviting organization (for example, with Conditional Access policies).
- **Branding** - Host/inviting organization's brand is used.

Manage external collaboration settings in Azure Active Directory

This unit describes how to enable Azure Active Directory (Azure AD) B2B collaboration, designate who can invite guests, and determine the permissions that guest users have in your Azure AD.

By default, all users and guests in your directory can invite guests even if they're not assigned to an admin role. External collaboration settings let you turn guest invitations on or off for different types of users in your organization. You can also delegate invitations to individual users by assigning roles that allow them to invite guests.

Azure AD allows you to restrict what external guest users can see in your Azure AD directory. By default, guest users are set to a limited permission level that blocks them from enumerating users, groups, or other directory resources, but lets them see membership of non-hidden groups. A new preview setting lets you restrict guest access even further, so that guests can only view their own profile information. For details, see [Restrict guest access permissions \(preview\)](#)¹².

Configure business-to-business external collaboration settings

With Azure AD B2B (Business to Business) collaboration, a tenant admin can set the following invitation policies:

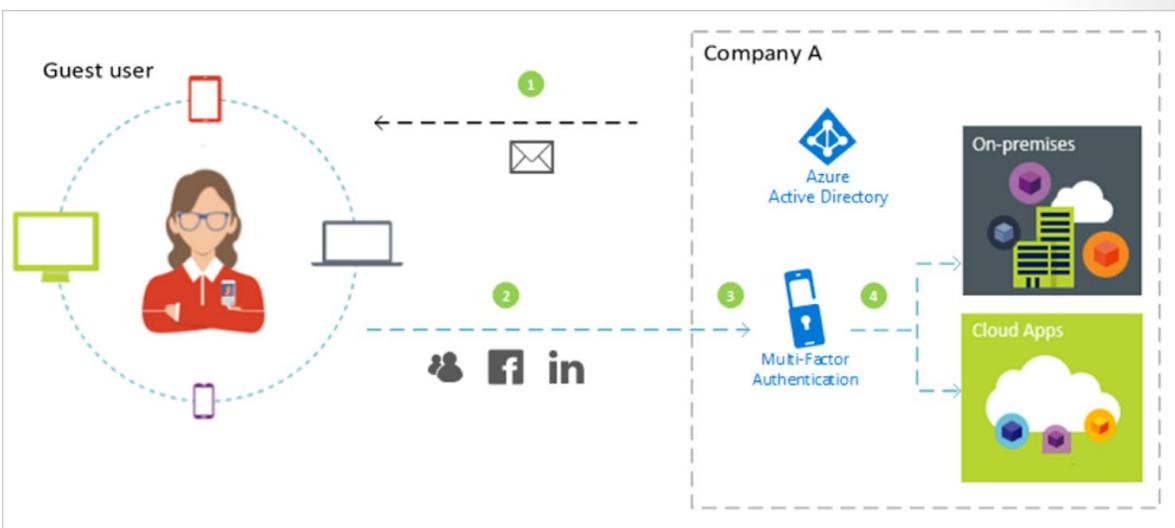
- Turn off invitations
- Only admins and users in the Guest Inviter role can invite

¹² <https://docs.microsoft.com/azure/active-directory/enterprise-users/users-restrict-guest-permissions>

- Admins, the Guest Inviter role, and members can invite
 - All users, including guests, can invite
- By default, all users, including guests, can invite guest users.

Invite External Users both Individually and in Bulk

As a user who is assigned any of the limited administrator directory roles, you can use the Azure portal to invite B2B collaboration users. You can invite guest users to the directory, to a group, or to an application. After you invite a user through any of these methods, the invited user's account is added to Azure Active Directory (Azure AD), with a user type of *Guest*. The guest user must then redeem their invitation to access resources. An invitation of a user does not expire.



After you add a guest user to the directory, you can either send the guest user a direct link to a shared app, or the guest user can click the redemption URL in the invitation email. Make sure your organization's external collaboration settings are configured such that you're allowed to invite guests. By default, all users and admins can invite guests. But your organization's external collaboration policies might be configured to prevent certain types of users or admins from inviting guests.

How users in your organization can invite guest users to an app

After a guest user has been added to the directory in Azure AD, an application owner can send the guest user a direct link to the app they want to share. Azure AD admins can also set up self-service management for gallery or SAML-based apps in their Azure AD tenant. This way, application owners can manage their own guest users, even if the guest users haven't been added to the directory yet. When an app is configured for self-service, the application owner uses their Access Panel to invite a guest user to an app or add a guest user to a group that has access to the app. Self-service app management for gallery and SAML-based apps requires some initial setup by an admin, which can be summarized as follows:

- Enable self-service group management for your tenant
- Create a group to assign to the app and make the user an owner
- Configure the app for self-service and assign the group to the app

How to bulk invite Azure AD B2B collaboration users

If you use Azure Active Directory (Azure AD) B2B collaboration to work with external partners, you can invite multiple guest users to your organization at the same time. Specifically, you do the following:

- Use **Bulk invite users** to prepare a comma-separated value (.csv) file with the user information and invitation preferences
- Upload the .csv file to Azure AD
- Verify the users were added to the directory

Understand the CSV template

Download and fill in the bulk upload CSV template to help you successfully invite Azure AD guest users in bulk. The CSV template you download might look like this example:

Row 1 must be preserved as-is, and the version number is always required.		
1	A	B
version:v1.0		
2	Email address to invite [inviteeEmail] Required	Redirection url [inviteRedirectURL] Re
3	Example: lstokes@fabrikam.com	https://myapps.azure.com
4		

Preserve the column headings as-is in row 2. Column headings indicate acceptable values and whether they're required. Don't add additional columns.

Use the entries in row 3 as examples. Remove the row's contents and replace the examples with your entries.

CSV template structure

The rows in a downloaded CSV template are as follows:

- **Version number:** The first row containing the version number must be included in the upload CSV.
- **Column headings:** The format of the column headings is `<Item name> [PropertyName] <Required or blank>`. For example, `Email address to invite [inviteeEmail] Required`. Some older versions of the template might have slight variations.
- **Examples row:** We have included in the template a row of examples of acceptable values for each column. You must remove the examples row and replace it with your own entries.

Additional guidance

- The first two rows of the upload template must not be removed or modified, or the upload can't be processed.
- The required columns are listed first.
- We don't recommend adding new columns to the template. Any additional columns you add are ignored and not processed.
- We recommend that you download the latest version of the CSV template as often as possible.

Click-Through Demo - Invite Guest User to use an application

Launch the click through demo (https://mslearn.microsoft.com/en-us/guides/Manage%20Guest%20User%20Access%20in%20Azure%20AD%20for%20B2B%20Collaboration)	In this interactive guide, you'll learn how to manage guest user access in Azure Active Directory for business-to-business (B2B) collaboration. You'll see how to invite external users to collaborate, assign resources to guest users, and create conditional access policies to keep data secure.
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

User Accounts in Azure Active Directory

Azure Active Directory (Azure AD) B2B collaboration users are added as guest users to the directory, and guest permissions in the directory are restricted by default. Your business may need some guest users to fill higher-privilege roles in your organization. To support defining higher-privilege roles, guest users can be added to any roles you desire, based on your organization's needs.

Add a B2B user to a role

Microsoft recommends that organizations use the rule of least privilege. You can use Privileged Identity Management (PIM) to grant access for B2B/guest users.

Default role

Sam Oogle - Directory role

User - PREVIEW

Save Discard

Search (Ctrl+ /)

Overview

MANAGE

Profile

Directory role

Groups

Licenses

Devices

Azure resources

ACTIVITY

Sign-ins

Audit logs

Directory role ●

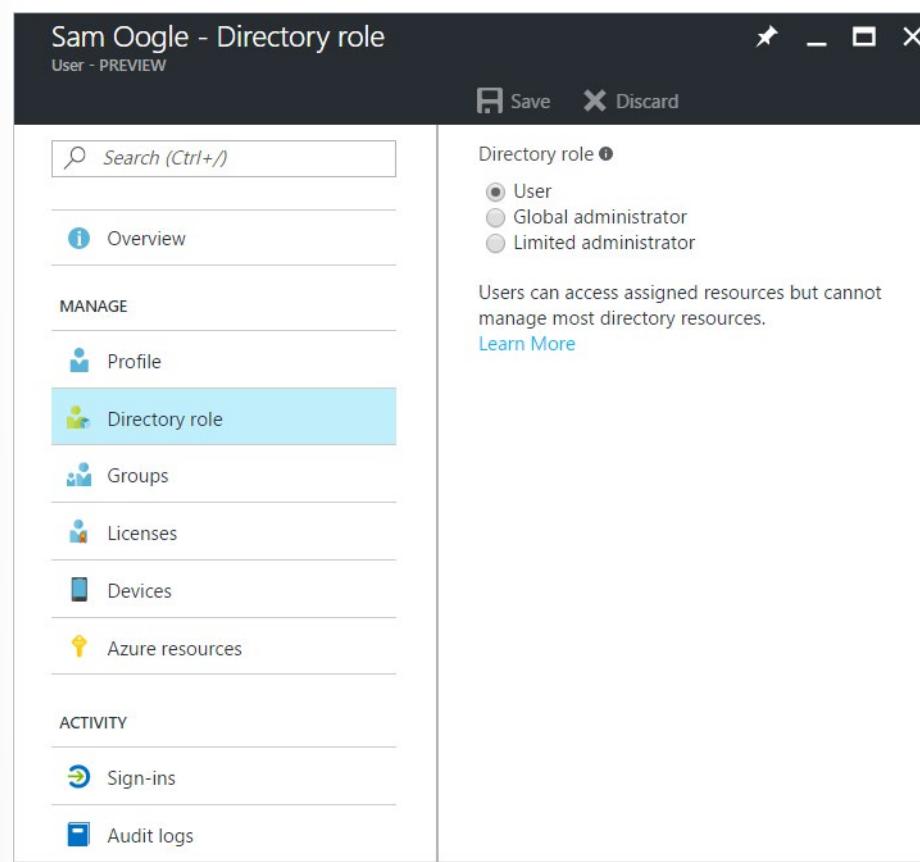
User

Global administrator

Limited administrator

Users can access assigned resources but cannot manage most directory resources.

[Learn More](#)



Global Administrator role

Sam Oogle - Directory role
User - PREVIEW

Save Discard

Search (Ctrl+ /)

Overview

MANAGE

Profile

Directory role

Groups

Licenses

Devices

Azure resources

ACTIVITY

Sign-ins

Audit logs

Directory role i

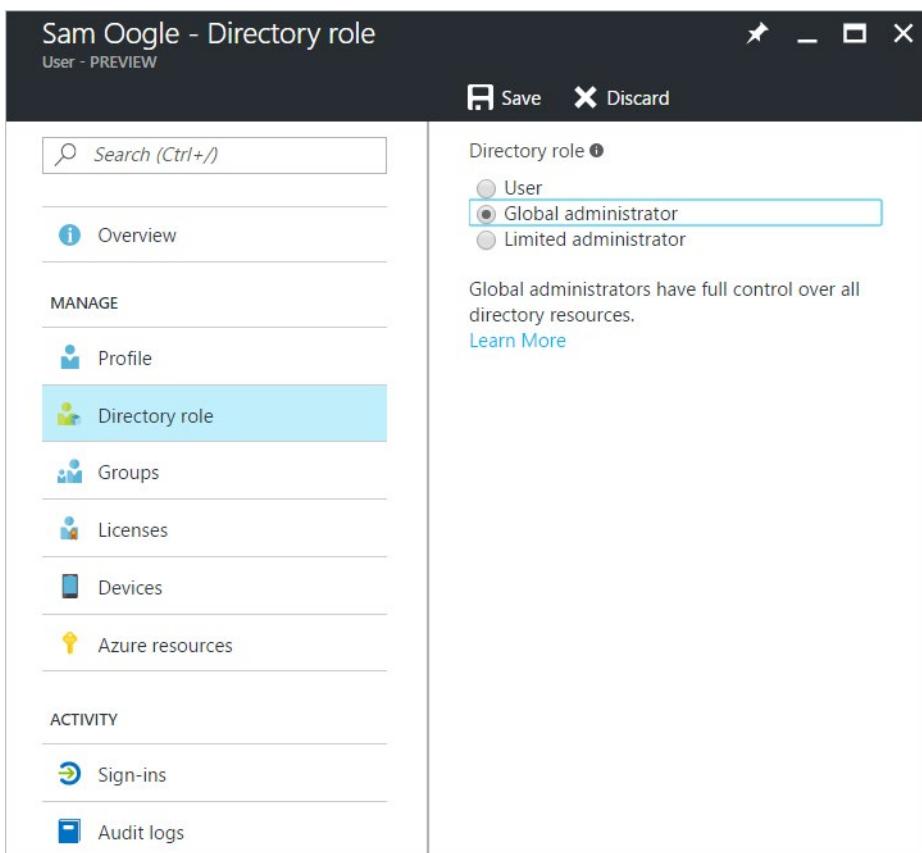
User

Global administrator

Limited administrator

Global administrators have full control over all directory resources.

[Learn More](#)



Limited Administrator role

The screenshot shows the Azure portal interface for managing a user's administrative roles. The top navigation bar indicates the user is being previewed. The left sidebar has sections for Overview, Manage (Profile, Groups, Licenses, Devices, Azure resources), and Activity (Sign-ins, Audit logs). The main content area is titled 'Sam Oogle - Directory role' and shows the 'Directory role' configuration. Under 'Manage directory role', the 'Limited administrator' option is selected. A note says to 'Select the administrative role or roles for this user.' Below this, a list of other administrative roles is shown, each with a detailed description and a 'More' link:

- >Password administrator
- Service administrator
- Billing administrator
- Exchange administrator
- Skype for Business administrator
- User administrator
- SharePoint administrator
- Compliance administrator
- Security reader
- Security administrator
- Privileged role administrator
- Intune Service administrator
- Guest inviter

Understand the B2B user

B2B guest user objects in Azure AD have properties and states before and after invitation redemption. An Azure AD business-to-business (B2B) collaboration user is a user with UserType = Guest. This guest user typically is from a partner organization and has limited privileges in the inviting directory, by default.

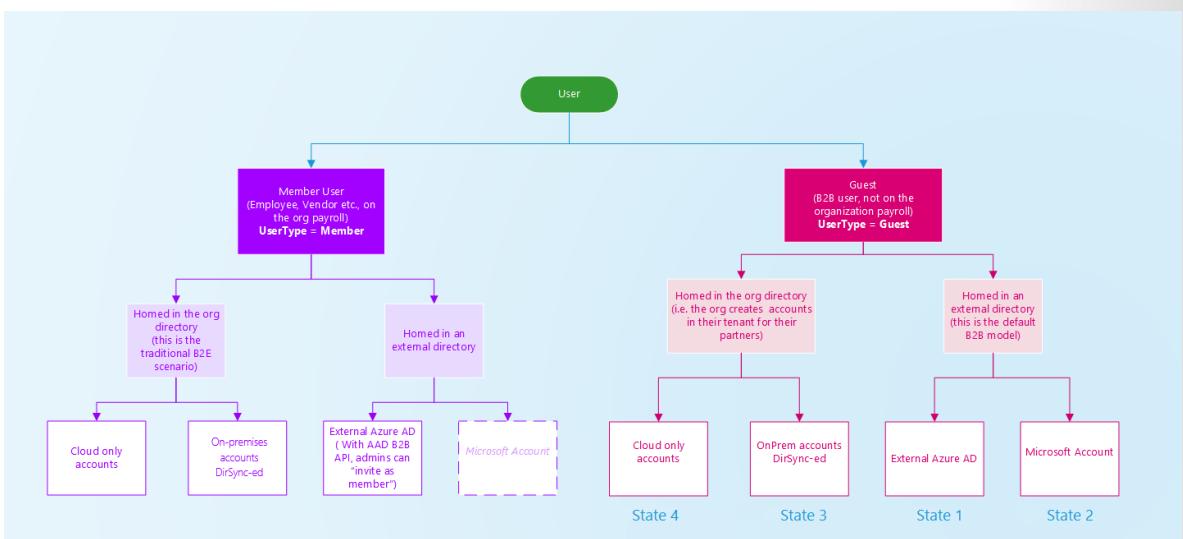
Depending on the inviting organization's needs, an Azure AD B2B collaboration user can be in one of the following account states:

- State 1: Homed in an external instance of Azure AD and represented as a guest user in the inviting organization. In this case, the B2B user signs in by using an Azure AD account that belongs to the invited tenant. If the partner organization doesn't use Azure AD, the guest user in Azure AD is still created. The requirements are that they redeem their invitation and Azure AD verifies their email address. This arrangement is also called a just-in-time (JIT) tenancy or a **viral** tenancy.

Important - Starting March 31, 2021, Microsoft will no longer support the redemption of invitations by creating unmanaged Azure AD accounts and tenants for B2B collaboration scenarios. In prepara-

tion, we encourage customers to opt into **email one-time passcode authentication**¹³. We welcome your feedback on this public preview feature and are excited to create even more ways to collaborate.

- State 2: Homed in a Microsoft or other account and represented as a guest user in the host organization. In this case, the guest user signs in with a Microsoft account or a social account (google.com or similar). The invited user's identity is created as a Microsoft account in the inviting organization's directory during offer redemption.
- State 3: Homed in the host organization's on-premises Active Directory and synced with the host organization's Azure AD. You can use Azure AD Connect to sync the partner accounts to the cloud as Azure AD B2B users with UserType = Guest.
- State 4: Homed in the host organization's Azure AD with UserType = Guest and credentials that the host organization manages.



Now, let's see what an Azure AD B2B collaboration user looks like in Azure AD.

Before invitation redemption

State 1 and State 2 accounts are the result of inviting guest users to collaborate by using the guest users' own credentials. When the invitation is initially sent to the guest user, an account is created in your directory. This account doesn't have any credentials associated with it because authentication is performed by the guest user's identity provider. The **Source** property for the guest user account in your directory is set to **Invited user**.

¹³ <https://docs.microsoft.com/azure/active-directory/external-identities/one-time-passcode>

The screenshot shows the Azure Active Directory User Profile page for a user named gsamoogle. The left sidebar has a 'Manage' section with 'Profile' selected, and other options like 'Directory role', 'Groups', 'Applications', 'Licenses', 'Devices', and 'Azure resources'. Below that is an 'Activity' section with 'Sign-ins' and 'Audit logs'. At the bottom is a 'Troubleshooting + Support' section with 'Troubleshoot' and 'New support request'. The main content area shows the user's profile picture (a globe icon), their name 'gsamoogle', and email 'gsamoogle@gmail.com'. It also displays 'User Sign-ins' (100), 'Group memberships' (0), and a timeline showing 50 sign-ins in December. Under the 'Identity' tab, the 'Source' field is set to 'Invited user'. A 'Resend invitation' button is visible. The 'Job info' tab is also present.

After invitation redemption

After the guest user accepts the invitation, the **Source** property is updated based on the guest user's identity provider.

For guest users in State 1, the **Source** is **External Azure Active Directory**.

Guest User1 - Profile

User Sign-ins

Date	Sign-ins
December	2

Group memberships

Group	Members
None	0

Identity edit

Name	First name	Last name
Guest User1	---	---
User name	User type	Invitation accepted
guestuser1@fabrikam.com	Guest	Yes
Object ID	Source	
462cd2ac-018a-4194...	External Azure Active Directory	

Job info edit

Job title	Department	Manager
-----------	------------	---------

For guest users in State 2, the **Source** is **Microsoft Account**.

gsamoogle - Profile

User Sign-ins

Date	Sign-ins
December	2

Group memberships

Group	Members
None	0

Identity edit

Name	First name	Last name
gsamoogle	---	---
User name	User type	Invitation accepted
gsamoogle@gmail.com	Guest	Yes
Object ID	Source	
8ea9cea8-1594-4...	Microsoft Account	

Job info edit

Job title	Department	Manager
-----------	------------	---------

For guest users in State 3 and State 4, the **Source** property is set to **Azure Active Directory** or **Windows Server Active Directory**, as described in the next section.

Key properties of the Azure AD B2B collaboration user

UserType

This property indicates the relationship of the user to the host tenancy. This property can have two values:

- Member: This value indicates an employee of the host organization and a user in the organization's payroll. For example, this user expects to have access to internal-only sites. This user is not considered an external collaborator.
- Guest: This value indicates a user who isn't considered internal to the company, such as an external collaborator, partner, or customer. Such a user isn't expected to receive a CEO's internal memo or receive company benefits, for example.

Note - The UserType has no relation to how the user signs in, the directory role of the user, and so on. This property simply indicates the user's relationship to the host organization and allows the organization to enforce policies that depend on this property.

Source

This property indicates how the user signs in.

- Invited User: This user has been invited but has not yet redeemed an invitation.
- External Azure Active Directory: This user is homed in an external organization and authenticates by using an Azure AD account that belongs to the other organization. This type of sign-in corresponds to State 1.
- Microsoft account: This user is homed in a Microsoft account and authenticates by using a Microsoft account. This type of sign-in corresponds to State 2.
- Windows Server Active Directory: This user is signed in from on-premises Active Directory that belongs to this organization. This type of sign-in corresponds to State 3.
- Azure Active Directory: This user authenticates by using an Azure AD account that belongs to this organization. This type of sign-in corresponds to State 4.

Note - Source and UserType are independent properties. A value of Source does not imply a particular value for UserType.

Can Azure AD B2B users be added as members instead of guests?

Typically, an Azure AD B2B user and guest user are synonymous. Therefore, an Azure AD B2B collaboration user is added as a user with UserType = Guest by default. However, in some cases, the partner organization is a member of a larger organization to which the host organization also belongs. If so, the host organization might want to treat users in the partner organization as members instead of guests. Use the Azure AD B2B Invitation Manager APIs to add or invite a user from the partner organization to the host organization as a member.

Filter for guest users in the directory

The screenshot shows the 'Users - All users' page in Microsoft Azure Active Directory. On the left, there's a navigation sidebar with links like 'All users', 'Deleted users', 'Password reset', 'User settings', 'Activity', 'Sign-ins', 'Audit logs', 'Troubleshooting + Support', 'Troubleshoot', and 'New support request'. The main area has a search bar for 'Name' and a dropdown for 'Show' set to 'Guest users only'. A table lists six guest users:

NAME	USER NAME	USER TYPE	SOURCE
stdealer	stdealer@comcast.net	Guest	Microsoft Account
bryce	bryce@litwarecorp.com	Guest	Invited user
basarajesh	basarajesh@yahoo.com	Guest	Microsoft Account
Sanda	sanda@contoso.com	Guest	Azure Active Directory
Sarat Subramaniam	sarat@fabrikam.com	Guest	External Azure Active Directory
tjb2b	tjb2b@live.com	Guest	Invited user

Convert UserType

It's possible to convert UserType from Member to Guest and vice-versa by using PowerShell. However, the UserType property represents the user's relationship to the organization. Therefore, you should change this property only if the relationship of the user to the organization changes. If the relationship of the user changes, should the user principal name (UPN) change? Should the user continue to have access to the same resources? Should a mailbox be assigned? We don't recommend changing the UserType by using PowerShell as an atomic activity. Also, in case this property becomes immutable by using PowerShell, we don't recommend taking a dependency on this value.

The screenshot shows the 'Microsoft - User settings' page. The left sidebar includes links for 'App registrations', 'Application proxy', 'Licenses', 'Azure AD Connect', 'Custom domain names', 'Mobility (MDM and MAM)', 'Password reset', 'Company branding', 'User settings' (which is selected), 'Properties', and 'Notifications settings'. The right pane contains several configuration sections:

- Enterprise applications**: Manage how end users launch and view their applications.
- App registrations**: Users can register applications. A 'Yes' button is highlighted.
- Administration portal**: Restrict access to Azure AD administration portal. A 'No' button is highlighted.
- External users**: Manage external collaboration settings.
- Access panel**: Manage settings for access panel preview features.

Remove guest user limitations

There may be cases where you want to give your guest users higher privileges. You can add a guest user to any role and even remove the default guest user restrictions in the directory to give a user the same privileges as members.

It's possible to turn off the default limitations so that a guest user in the company directory has the same permissions as a member user.

The screenshot shows the 'Microsoft - User settings' page in the Azure Active Directory portal. The left sidebar lists various options: App registrations, App registrations (Preview), Application proxy, Licenses, Azure AD Connect, Custom domain names, Mobility (MDM and MAM), Password reset, Company branding, **User settings** (which is selected), Properties, Notifications settings, Security, Identity Secure Score (Preview), and Conditional Access. The main pane contains sections for Enterprise applications (Manage how end users launch and view their applications), App registrations (Users can register applications, with 'Yes' selected), Administration portal (Restrict access to Azure AD administration portal, with 'No' selected), External users (Manage external collaboration settings), and Access panel (Manage settings for access panel preview features).

Dynamic groups and Azure Active Directory B2B collaboration

What are dynamic groups?

Dynamic configuration of security group membership for Azure AD is available in the [Azure portal](#)¹⁴. Administrators can set rules to populate groups that are created in Azure AD based on user attributes (such as userType, department, or country/region). Members can be automatically added to or removed from a security group based on their attributes. These groups can provide access to applications or cloud resources (SharePoint sites, documents) and to assign licenses to members. The appropriate Azure AD Premium P1 or P2 licensing is required to create and use dynamic groups.

¹⁴ <https://portal.azure.com/>

Configure Identity Providers

You can set up direct federation with any organization whose identity provider (IdP) supports the Security Assertion Markup Language (SAML) 2.0 or WS-Federation (WS-Fed) protocol. When you set up direct federation with a partner's IdP, new guest users from that domain can use their own IdP-managed organizational account to sign in to your Azure Active Directory (Azure AD) tenant and start collaborating with you. There's no need for the guest user to create a separate Azure AD account.

Note - Direct federation guest users must sign in using a link that includes the tenant context (for example, <https://myapps.microsoft.com/?tenantid=tenant id> or <https://portal.azure.com/tenant id>, or in the case of a verified domain, <https://myapps.microsoft.com/verified domain.onmicrosoft.com>). Direct links to applications and resources also work as long as they include the tenant context. Direct federation users are currently unable to sign in using common endpoints that have no tenant context. For example, using <https://myapps.microsoft.com>, <https://portal.azure.com>, or <https://teams.microsoft.com> will result in an error.

When is a guest user authenticated with direct federation?

After you set up direct federation with an organization, any new guest users you invite will be authenticated using direct federation. Note that setting up direct federation doesn't change the authentication method for guest users who have already redeemed an invitation from you. Here are some examples:

- If guest users have already redeemed invitations from you, and you subsequently set up direct federation with their organization, those guest users will continue to use the same authentication method they used before you set up direct federation.
- If you set up direct federation with a partner organization and invite guest users, and then the partner organization later moves to Azure AD, the guest users who have already redeemed invitations will continue to use direct federation, as long as the direct federation policy in your tenant exists.
- If you delete direct federation with a partner organization, any guest users currently using direct federation will be unable to sign in.

In any of these scenarios, you can update a guest user's authentication method by deleting the guest user account from your directory and reinviting them.

Direct federation is tied to domain namespaces, such as contoso.com and fabrikam.com. When establishing a direct federation configuration with AD FS or a third-party IdP, organizations associate one or more domain namespaces to these IdPs.

End-user experience

With direct federation, guest users sign into your Azure AD tenant using their own organizational account. When they are accessing shared resources and are prompted for sign-in, direct federation users are redirected to their IdP. After successful sign-in, they are returned to Azure AD to access resources. Direct federation users' refresh tokens are valid for 12 hours, the default length for passthrough refresh token in Azure AD. If the federated IdP has SSO enabled, the user will experience SSO and will not see any sign-in prompt after initial authentication.

Limitations

Direct federation limitations include those described in the following table.

Limitation	Description
DNS-verified domains in Azure AD	The domain you want to federate with must not be DNS-verified in Azure AD. You're allowed to set up direct federation with unmanaged (email-verified or "viral") Azure AD tenants because they aren't DNS-verified.
Authentication URL	Direct federation is only allowed for policies where the authentication URL's domain matches the target domain, or where the authentication URL is a specified allowed identity provider. Current providers include: accounts.google.com -and- pingidentity.com -and- okta.com -and- federation.exostar.com (This list is subject to change.) For example, when setting up direct federation for fabrikam.com , the authentication URL <code>https://fabrikam.com/adfs</code> will pass the validation. A host in the same domain will also pass, for example <code>https://sts.fabrikam.com/adfs</code> . However, the authentication URL <code>https://fabrikamconglomerate.com/adfs</code> or <code>https://fabrikam.com.uk/adfs</code> for the same domain won't pass.
Signing certificate renewal	If you specify the metadata URL in the identity provider settings, Azure AD will automatically renew the signing certificate when it expires. However, if the certificate is rotated for any reason before the expiration time, or if you don't provide a metadata URL, Azure AD will be unable to renew it. In this case, you'll need to update the signing certificate manually.
Limit on federation relationships	Currently, a maximum of 1,000 federation relationships is supported. This limit includes both internal federations and direct federations.
Limit on multiple domains	Microsoft doesn't currently support direct federation with multiple domains from the same tenant.

Security Assertion Markup Language 2.0 configuration

Azure AD B2B can be configured to federate with identity providers that use the SAML protocol with specific requirements listed below.

Note - The target domain for direct federation must not be DNS-verified on Azure AD. The authentication URL domain must match the target domain or it must be the domain of an allowed identity provider.

Required Security Assertion Markup Language 2.0 attributes and claims

The following tables show requirements for specific attributes and claims that must be configured at the third-party identity provider. To set up direct federation, the following attributes must be received in the

SAML 2.0 response from the identity provider. These attributes can be configured by linking to the online security token service XML file or by entering them manually.

Required attributes for the SAML 2.0 response from the IdP:

Attribute	Value
AssertionConsumerService	https://login.microsoftonline.com/login.srf
Audience	urn:federation:MicrosoftOnline
Issuer	The issuer URI of the partner IdP, for example http://www.example.com/exk10l-6w90DHM0yi...

Required claims for the SAML 2.0 token issued by the IdP:

Attribute	Value
NameID Format	urn:oasis:names:tc:SAML-2.0:nameid-format:persistent
emailaddress	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress

WS-Federation configuration

Azure AD B2B can be configured to federate with identity providers that use the WS-Fed protocol with some specific requirements as listed below. Currently, the two WS-Fed providers have been tested for compatibility with Azure AD include AD FS and Shibboleth.

The target domain for direct federation must not be DNS-verified on Azure AD. The authentication URL domain must match either the target domain or the domain of an allowed identity provider.

Required WS-Federation attributes and claims

The following tables show requirements for specific attributes and claims that must be configured at the third-party WS-Fed identity provider. To set up direct federation, the following attributes must be received in the WS-Fed message from the identity provider. These attributes can be configured by linking to the online security token service XML file or by entering them manually.

Required attributes in the WS-Fed message from the IdP:

Attribute	Value
PassiveRequestorEndpoint	https://login.microsoftonline.com/login.srf
Audience	urn:federation:MicrosoftOnline
Issuer	The issuer URI of the partner IdP, for example http://www.example.com/exk10l-6w90DHM0yi...

Required claims for the WS-Fed token issued by the IdP:

Attribute	Value
ImmutableID	http://schemas.microsoft.com/LiveID/Federation/2008/05/ImmutableID

Attribute	Value
emailaddress	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress

Add Google as an identity provider for B2B guest users

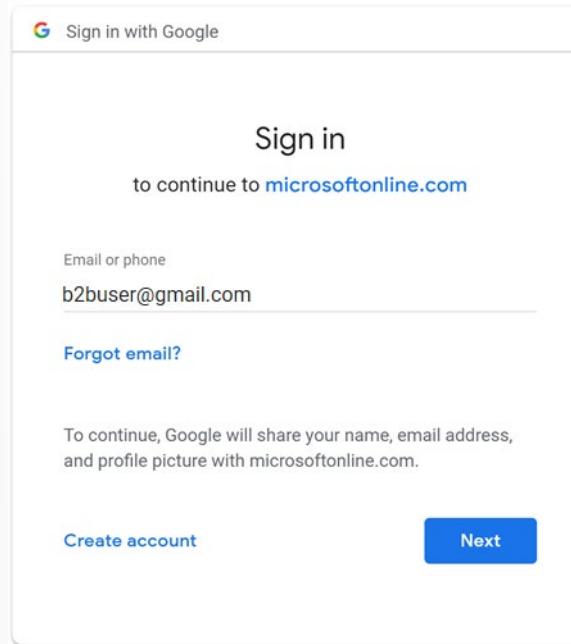
By setting up federation with Google, you can allow invited users to sign in to your shared apps and resources with their own Gmail accounts, without having to create Microsoft accounts.

Note - Google federation is designed specifically for Gmail users. To federate with G Suite domains, use [direct federation](#)¹⁵.

What is the experience for the Google user?

When you send an invitation to Google Gmail users, the guest users should access your shared apps or resources by using a link that includes the tenant context. Their experience varies depending on whether they're already signed in to Google:

- Guest users who aren't signed in to Google will be prompted to do so.
- Guest users who are already signed in to Google will be prompted to choose the account they want to use. They must choose the account you used to invite them.
- Guest users who see a "header too long" error can clear their cookies or open a private or incognito window and try to sign in again.



Deprecation of WebView sign-in support

Starting January 4, 2021, Google is deprecating embedded WebView sign-in support. If you're using Google federation or self-service sign-up with Gmail, you should test your line-of-business native

¹⁵ <https://docs.microsoft.com/azure/active-directory/external-identities/direct-federation>

applications for compatibility. If your apps include WebView content that requires authentication, Google Gmail users won't be able to authenticate. The following are known scenarios that will impact Gmail users:

- Windows apps that use embedded WebView or the WebAccountManager (WAM) on older versions of Windows.
- Other native apps you've developed that use an embedded browser framework for authentication.

This change does not affect:

- Windows apps that use embedded WebView or the WebAccountManager (WAM) on the latest versions of Windows
- Microsoft iOS apps
- G Suite identities, for example when you're using SAML-based direct federation with G Suite

We're continuing to test various platforms and scenarios, and will update published information accordingly.

To test your apps for compatibility

1. Follow **Google's guidance¹⁶** to determine if your apps are affected.
2. Using Fiddler or another testing tool, inject a header during sign-in and use a Google external identity to test sign-in:
 1. Add Google-Accounts-Check-OAuth-Login:true to your HTTP request headers when the requests are sent to accounts.google.com.
 2. Attempt to sign in to the app by entering a Gmail address in the accounts.google.com sign-in page.
 3. If sign-in fails and you see an error such as "This browser or app may not be secure," your Google external identities will be blocked from signing in.
3. Resolve the issue by doing one of the following:
 - If your Windows app uses embedded WebView or the WebAccountManager (WAM) on an older version of Windows, update to the latest version of Windows.
 - Modify your apps to use the system browser for sign-in. For details, see **Embedded vs System Web UI¹⁷** in the MSAL.NET documentation.

Sign-in endpoints

Teams fully supports Google guest users on all devices. Google users can sign in to Teams from a common endpoint like <https://teams.microsoft.com>.

Other applications' common endpoints might not support Google users. Google guest users must sign in by using a link that includes your tenant information. Following are examples:

- <https://myapps.microsoft.com/?tenantid=your tenant ID>
- <https://portal.azure.com/your tenant ID>
- <https://myapps.microsoft.com/your verified domain.onmicrosoft.com>

¹⁶ <https://developers.googleblog.com/2020/08/guidance-for-our-effort-to-block-less-secure-browser-and-apps.html>

¹⁷ <https://docs.microsoft.com/azure/active-directory/develop/msal-net-web-browsers>

If Google guest users try to use a link like `https://myapps.microsoft.com` or `https://portal.azure.com`, they'll get an error.

You can also give Google guest users a direct link to an application or resource, as long as the link includes your tenant information. For example, `https://myapps.microsoft.com/signin/Twitter/application ID?tenantId=your tenant ID`.

Step 1: Configure a Google developer project

First, create a new project in the Google Developers Console to obtain a client ID and a client secret that you can later add to Azure Active Directory (Azure AD).

1. Go to the Google APIs at <https://console.developers.google.com>¹⁸, and sign in with your Google account. We recommend that you use a shared team Google account.
2. Accept the terms of service if you're prompted to do so.
3. Create a new project: On the dashboard, select **Create Project**, give the project a name (for example, **Azure AD B2B**), and then select **Create**:

The screenshot shows the 'New Project' screen in the Google Developers Console. At the top, it says 'New Project'. Below that, there's a message: 'You have 11 projects remaining in your quota. Request an increase or delete projects.' with a 'Learn more' link and a 'MANAGE QUOTAS' button. The 'Project Name *' field is filled with 'MyB2BApp' and has a red border around it. Below it, 'Project ID: myb2bapp' is shown with an 'EDIT' link. Under 'Location *', 'No organization' is selected with a 'BROWSE' button. At the bottom, there are 'CREATE' and 'CANCEL' buttons, both of which are highlighted with red boxes.

4. On the **APIs & Services** page, select **View** under your new project.
5. Select **Go to APIs overview** on the APIs card. Select **OAuth consent screen**.
6. Select **External**, and then select **Create**.
7. On the **OAuth consent screen**, enter an **Application name**:

¹⁸ <https://console.developers.google.com/>

OAuth consent screen

Before your users authenticate, this consent screen will allow them to choose whether they want to grant access to their private data, as well as give them a link to your terms of service and privacy policy. This page configures the consent screen for all applications in this project.

Verification status
Not published

Application name  The name of the app asking for consent

Application logo  An image on the consent screen that will help users recognize your app



8. Scroll to the **Authorized domains** section and enter **microsoftonline.com**:

Credentials

Add scopes

Authorized domains  To protect you and your users, Google only allows applications that authenticate using OAuth to use Authorized Domains. Your applications' links must be hosted on Authorized Domains. [Learn more](#)



9. Select **Save**.

10. Select **Credentials**. On the **Create credentials** menu, select **OAuth client ID**:

APIs

Credentials

You need credentials to access APIs. [Enable the APIs you plan to use](#) and then create the credentials they require. Depending on the API, you need an API key, a service account, or an OAuth 2.0 client ID. Refer to the [API documentation](#) for details.

Create credentials ▾

- API key**
Identifies your project using a simple API key to check quota and access
- OAuth client ID**
Requests user consent so your app can access the user's data
- Service account key**
Enables server-to-server, app-level authentication using robot accounts
- Help me choose**
Asks a few questions to help you decide which type of credential to use

11. Under **Application type**, select **Web application**. Give the application a suitable name, like **Azure AD B2B**. Under **Authorized redirect URIs**, enter the following URLs:

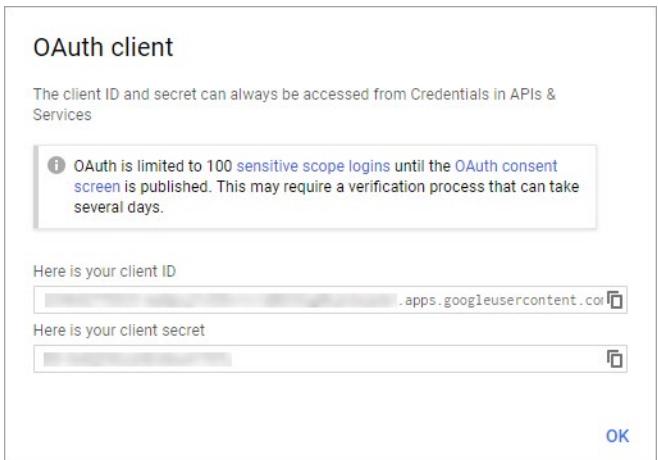
- <https://login.microsoftonline.com>
- <https://login.microsoftonline.com/te/{tenant ID}/oauth2/authresp> (where **tenant ID** is your tenant ID)

Note - To find your tenant ID, go to the [Azure portal¹⁹](#). Under Azure Active Directory, select Properties and copy the Tenant ID.

The screenshot shows the 'Create OAuth client ID' page. At the top, there's a back arrow and the title 'Create OAuth client ID'. Below the title, a note says: 'For applications that use the OAuth 2.0 protocol to call Google APIs, you can use an OAuth 2.0 client ID to generate an access token. The token contains a unique identifier. See [Setting up OAuth 2.0](#) for more information.' The 'Application type' section has a radio button for 'Web application' which is selected (indicated by a red box). Other options include 'Android', 'Chrome App', 'iOS', and 'Other'. The 'Name' field contains 'AAD B2B Web App'. The 'Restrictions' section notes that origins and redirect domains must be added to the list of Authorized Domains in the OAuth consent settings. The 'Authorized JavaScript origins' field contains 'https://www.example.com'. The 'Authorized redirect URIs' field contains two entries: 'https://login.microsoftonline.com' and 'https://login.microsoftonline.com/te/{tenant ID}/oauth2/authresp', with the second entry highlighted by a red box. At the bottom are 'Create' and 'Cancel' buttons.

12. Select **Create**. Copy the client ID and client secret. You'll use them when you add the identity provider in the Azure portal.

¹⁹ <https://portal.azure.com/>



Step 2: Configure Google federation in Azure AD

You'll now set the Google client ID and client secret. You can use the Azure portal or PowerShell to do so. Be sure to test your Google federation configuration by inviting yourself. Use a Gmail address and try to redeem the invitation with your invited Google account.

To configure Google federation in the Azure portal

1. Go to the [Azure portal²⁰](#). On the left pane, select **Azure Active Directory**.
2. Select **External Identities**.
3. Select **All identity providers**, and then select the **Google** button.
4. Enter the client ID and client secret you obtained earlier. Select **Save**:

²⁰ <https://portal.azure.com/>

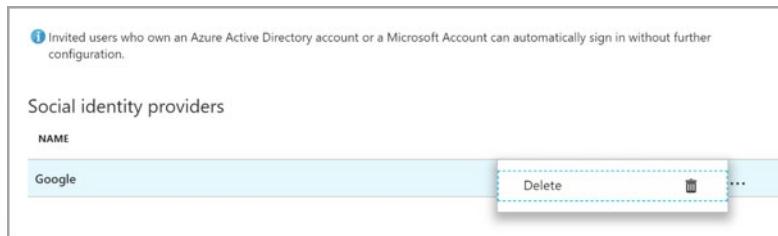
The screenshot shows a dialog box titled 'Add Google identity provider'. It contains a message: 'You must configure credentials at Google APIs first to get the client ID and client secret.' with a 'Learn more' link. There are three input fields: 'Name' (Google), 'Client ID' (Client ID), and 'Client secret' (Client secret). A 'Save' button is at the bottom.

How do I remove Google federation?

You can delete your Google federation setup. If you do so, Google guest users who have already redeemed their invitation won't be able to sign in. But you can give them access to your resources again by deleting them from the directory and reinviting them.

To delete Google federation in the Azure AD portal

1. Go to the [Azure portal²¹](#). On the left pane, select **Azure Active Directory**.
2. Select **External Identities**.
3. Select **All identity providers**.
4. On the **Google** line, select the ellipsis button (...) and then select **Delete**.



5. Select **Yes** to confirm the deletion.

²¹ <https://portal.azure.com/>

Add Facebook as an identity provider for external identities

You can add Facebook to your self-service sign-up user flows (Preview) so that users can sign in to your applications using their own Facebook accounts. To allow users to sign in using Facebook, you'll first need to enable self-service sign-up for your tenant. After you add Facebook as an identity provider, set up a user flow for the application and select Facebook as one of the sign-in options.

Note - Users can only use their Facebook accounts to sign up through apps using self-service sign-up and user flows. Users cannot be invited and redeem their invitation using a Facebook account.

Create an app in the Facebook developers console

To use a Facebook account as an identity provider, you need to create an application in the Facebook developers console. If you don't already have a Facebook account, you can sign up at <https://www.facebook.com/>.

Note - Use the following URLs in the steps 9 and 16 below.

- For **Site URL** enter the address of your application, such as `https://contoso.com`.
 - For **Valid OAuth redirect URIs**, enter `https://login.microsoftonline.com/{tenant-id}/oauth2/authresp`. You can find your **tenant-ID** in the Azure Active Directory Overview blade.
1. Sign in to [Facebook for developers](https://developers.facebook.com/)²² with your Facebook account credentials.
 2. If you have not already done so, you need to register as a Facebook developer. To do this, select **Get Started** on the upper-right corner of the page, accept Facebook's policies, and complete the registration steps.
 3. Select **My Apps** and then **Create App**.
 4. Enter a **Display Name** and a valid **Contact Email**.
 5. Select **Create App ID**. This may require you to accept Facebook platform policies and complete an online security check.
 6. Select **Settings > Basic**.
 7. Choose a **Category**, for example Business and Pages. This value is required by Facebook, but not used for Azure AD.
 8. At the bottom of the page, select **Add Platform**, and then select **Website**.
 9. In **Site URL**, enter the appropriate URL (noted above).
 10. In **Privacy Policy URL**, enter the URL for the page where you maintain privacy information for your application, for example `http://www.contoso.com`.
 11. Select **Save Changes**.
 12. At the top of the page, copy the value of **App ID**.
 13. Select **Show** and copy the value of **App Secret**. You use both of them to configure Facebook as an identity provider in your tenant. **App Secret** is an essential security credential.
 14. Select the plus sign next to **PRODUCTS**, and then select **Set up** under **Facebook Login**.
 15. Under **Facebook Login**, select **Settings**.

²² <https://developers.facebook.com/>

16. In **Valid OAuth redirect URLs**, enter the appropriate URL (noted above).
17. Select **Save Changes** at the bottom of the page.
18. To make your Facebook application available to Azure AD, select the Status selector at the top right of the page and turn it **On** to make the Application public, and then select **Switch Mode**. At this point the Status should change from **Development** to **Live**.

Configure a Facebook account as an identity provider

Now you'll set the Facebook client ID and client secret, either by entering it in the Azure AD portal or by using PowerShell. You can test your Facebook configuration by signing up via a user flow on an app enabled for self-service sign-up.

To configure Facebook federation in the Azure AD portal

1. Sign in to the **Azure portal**²³ as the global administrator of your Azure AD tenant.
2. Under **Azure services**, select **Azure Active Directory**.
3. In the left menu, select **External Identities**.
4. Select **All identity providers**, then select **Facebook**.
5. For the **Client ID**, enter the **App ID** of the Facebook application that you created earlier.
6. For the **Client secret**, enter the **App Secret** that you recorded.

The screenshot shows a modal dialog titled 'Add social identity provider'. It has a note: 'You must configure your Facebook Developer account first to get a client ID and client secret.' with a 'Learn more' link. There are three input fields: 'Name' (Facebook), 'Client ID *' (Client ID), and 'Client secret *' (Client secret). The 'Client ID *' field is highlighted with a red border.

7. Select **Save**.

How do I remove Facebook federation?

You can delete your Facebook federation setup. If you do so, any users who have signed up through user flows with their Facebook accounts will no longer be able to log in.

To delete Facebook federation in the Azure AD portal:

1. Go to the **Azure portal**²⁴. In the left pane, select **Azure Active Directory**.
2. Select **External Identities**.

²³ <https://portal.azure.com/>

²⁴ <https://portal.azure.com/>

3. Select **All identity providers**.
4. On the **Facebook** line, select the context menu (...) and then select **Delete**.
5. Select **Yes** to confirm deletion.

Implement and Manage Hybrid Identity

Introduction

Hybrid identity allows corporations to have identity solutions that span on-premises and cloud-based solutions. This capability provides unified authentication and authorization capabilities to resources regardless of their location.

Organizations today are adding cloud application to their existing on-premises apps, which makes them hybrid companies. They need to have identity solutions that authenticate and authorize users to access applications and the underlying data in a secure way. An on-premises Active Directory solution is not enough; extending to the cloud with Azure Active Directory (Azure AD) is necessary to provide a hybrid identity solution.

In this module, you will implement and manage a hybrid identity solution using Azure Active Directory and Azure AD Connect. You'll learn how to use the password hash synchronization (PHS) and pass-through authentication (PTA) to ensure you have the right authentication method for your needs. Then you will explore how single-sign-on (SSO) enables your users to access the apps they need while using secure access methods. Next, you will see how to connect to other existing external directories with Active Directory Federated Services (ADFS). Finally, you will learn how Azure AD Connect Health monitors the health of your identity solution and how to troubleshoot some common synchronization errors.

By the end of this module, you will be able to implement and manage a hybrid identity solution.

Learning objectives

In this module, you will:

- Plan, design, and implement Azure AD Connect
- Manage Azure AD Connect
- Implement and manage password hash synchronization (PHS)
- Implement and manage pass-through authentication (PTA)
- Implement and manage seamless single sign-on (Seamless SSO)
- Implement and manage federation excluding manual AD FS deployments
- Troubleshoot synchronization errors
- Implement and manage Azure AD Connect Health

Prerequisites

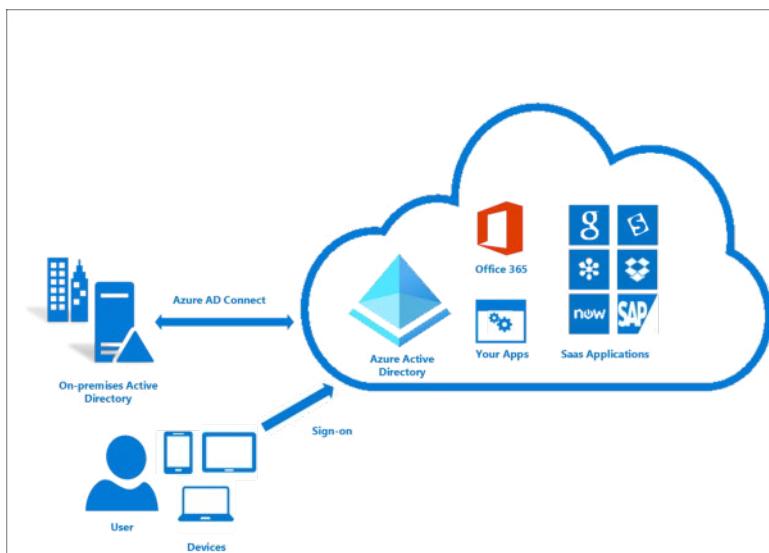
None

Plan, Design, and Implement Azure AD Connect

Azure AD Connect is a solution that bridges an organization's on-premises Active Directory with your cloud-based Azure Active Directory. This allows IT to synchronize identities from on-premises into Azure to ensure a consistent identity across both platforms. This connection enables services like password hash synchronization, pass-through authentication, and seamless single sign-on.

Azure AD Connect is the Microsoft tool designed to meet and accomplish your hybrid identity goals. It provides the following capabilities:

- Synchronization - Responsible for creating users, groups, and other objects. As well as, making sure identity information for your on-premises users and groups is matching the cloud. This synchronization also includes password hashes.
- Password hash synchronization - A sign-in method that synchronizes a hash of a user's on-premises AD password with Azure AD.
- Pass-through authentication - A sign-in method that allows users to use the same password on-premises and in the cloud, but doesn't require the additional infrastructure of a federated environment.
- Federation integration - Federation is an optional part of Azure AD Connect and can be used to configure a hybrid environment using an on-premises AD FS infrastructure. It also provides AD FS management capabilities such as certificate renewal and additional AD FS server deployments.
- Health monitoring - Azure AD Connect Health provides robust monitoring.



Why use Azure AD Connect?

Integrating your on-premises directories with Azure AD makes your users more productive by providing a common identity for accessing both cloud and on-premises resources. With Azure AD Connect, users can use a single identity to access on-premises applications and cloud services such as Microsoft 365. Additionally, organizations can provide an easy deployment experience for synchronization and sign-in using a single tool. Azure AD Connect replaces older versions of identity integration tools such as DirSync and Azure AD Sync; and is included in your Azure AD subscription.

Select an authentication method

Identity is the new control plane of IT security, so authentication is an organization's access guard to the new cloud world. Organizations need an identity control plane that strengthens their security and keeps their cloud apps safe from intruders. When the Azure AD hybrid identity solution is your new control plane, authentication is the foundation of cloud access. Choosing the correct authentication method is a crucial first decision in setting up an Azure AD hybrid identity solution. To choose an authentication

method, you need to consider the time, existing infrastructure, complexity, and cost of implementing your choice. These factors are different for every organization and might change over time.

Cloud authentication

When you choose this authentication method, Azure AD handles users' sign-in process. Coupled with seamless single sign-on (SSO), users can sign into cloud apps without having to reenter their credentials. With cloud authentication, you can choose from two options:

Azure AD password hash synchronization (PHS). The simplest way to enable authentication for on-premises directory objects in Azure AD. Users can use the same username and password that they use on-premises without having to deploy any additional infrastructure.

- **Effort.** Password hash synchronization requires the least effort regarding deployment, maintenance, and infrastructure. This level of effort typically applies to organizations that only need their users to sign in to Microsoft 365, SaaS apps, and other Azure AD-based resources. When turned on, password hash synchronization is part of the Azure AD Connect sync process and runs every two minutes.
- **User experience.** To improve users' sign-in experience, deploy seamless SSO with password hash synchronization. Seamless SSO eliminates unnecessary prompts when users are signed in.
- **Advanced scenarios.** If organizations choose to, it's possible to use insights from identities with Azure AD Identity Protection reports with Azure AD Premium P2. An example is the leaked credentials report. Windows Hello for Business has specific requirements when you use password hash synchronization. Azure AD Domain Services requires password hash synchronization to provision users with their corporate credentials in the managed domain.
- **Business continuity.** Using password hash synchronization with cloud authentication is highly available as a cloud service that scales to all Microsoft datacenters. To make sure password hash synchronization does not go down for extended periods, deploy a second Azure AD Connect server in staging mode in a standby configuration.
- **Considerations.** Currently, password hash synchronization doesn't immediately enforce changes in on-premises account states. In this situation, a user has access to cloud apps until the user account state is synchronized to Azure AD. Organizations might want to overcome this limitation by running a new synchronization cycle after administrators do bulk updates to on-premises user account states. An example is disabling accounts.

Azure AD pass-through authentication (PTA). Provides a simple password validation for Azure AD authentication services by using a software agent that runs on one or more on-premises servers. The servers validate the users directly with your on-premises Active Directory, which ensures that the password validation doesn't happen in the cloud. Companies with a security requirement to immediately enforce on-premises user account states, password policies, and sign-in hours might use this authentication method.

- **Effort.** For pass-through authentication, you need one or more (we recommend three) lightweight agents installed on existing servers. These agents must have access to your on-premises Active Directory Domain Services, including your on-premises AD domain controllers. They need outbound access to the Internet and access to your domain controllers. For this reason, it's not supported to deploy the agents in a perimeter network.
- **User experience.** To improve users' sign-in experience, deploy seamless SSO with pass-through authentication. Seamless SSO eliminates unnecessary prompts after users sign in.
- **Advanced scenarios.** Pass-through authentication enforces the on-premises account policy at the time of sign-in. For example, access is denied when an on-premises user's account state is disabled,

locked out, or their password expires or the logon attempt falls outside the hours when the user is allowed to sign in.

- **Business continuity.** We recommend that you deploy two extra pass-through authentication agents. These extras are in addition to the first agent on the Azure AD Connect server. This additional deployment ensures high availability of authentication requests. When you have three agents deployed, one agent can still fail when another agent is down for maintenance.
- **Considerations.** You can use password hash synchronization as a backup authentication method for pass-through authentication when the agents can't validate a user's credentials due to a significant on-premises failure. Fail over to password hash synchronization doesn't happen automatically and you must use Azure AD Connect to switch the sign-on method manually.

Federated authentication

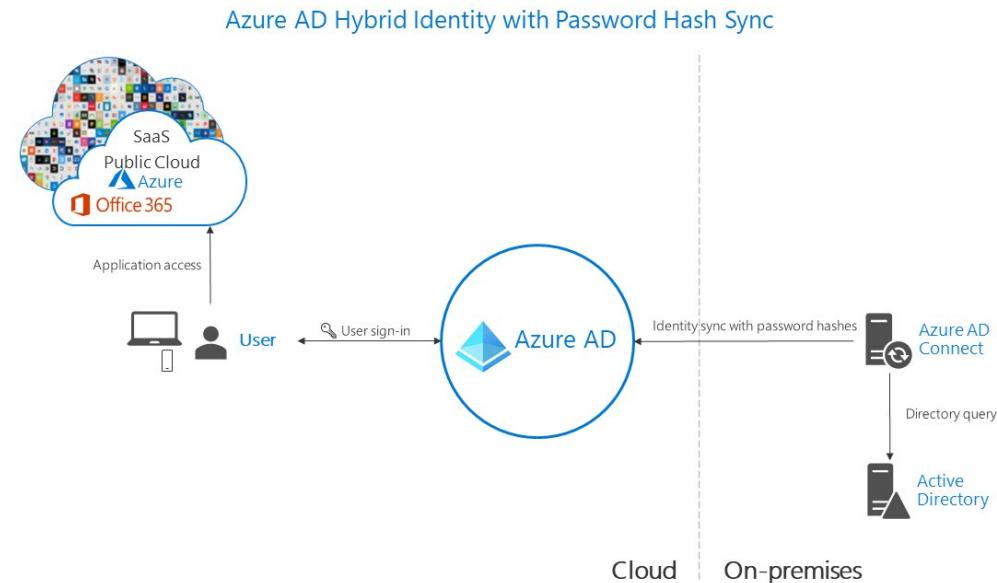
When you choose this authentication method, Azure AD hands off the authentication process to a separate trusted authentication system, such as on-premises Active Directory Federation Services (AD FS), to validate the user's password. The authentication system can provide additional advanced authentication requirements. Examples are smartcard-based authentication or third-party multifactor authentication.

- **Effort.** A federated authentication system relies on an external trusted system to authenticate users. Some companies want to reuse their existing federated system investment with their Azure AD hybrid identity solution. The maintenance and management of the federated system falls outside the control of Azure AD. It's up to the organization by using the federated system to make sure it's deployed securely and can handle the authentication load.
- **User experience.** The user experience of federated authentication depends on the implementation of the features, topology, and configuration of the federation farm. Some organizations need this flexibility to adapt and configure the access to the federation farm to suit their security requirements. For example, it's possible to configure internally connected users and devices to sign in users automatically, without prompting them for credentials. This configuration works because they already signed into their devices. If necessary, some advanced security features make users' sign-in process more difficult.
- **Advanced scenarios.** A federated authentication solution is required when customers have an authentication requirement that Azure AD doesn't support natively.
 - Authentication that requires smartcards or certificates.
 - On-premises MFA servers or third-party multifactor providers requiring a federated identity provider.
 - Authentication by using third-party authentication solutions.
 - Sign in that requires a sAMAccountName, for example DOMAIN\username, instead of a User Principal Name (UPN), for example, user@domain.com.
- **Business continuity.** Federated systems typically require a load-balanced array of servers, known as a farm. This farm is configured in an internal network and perimeter network topology to ensure high availability for authentication requests.
- **Considerations.** Federated systems typically require a more significant investment in on-premises infrastructure. Most organizations choose this option if they already have an on-premises federation investment. And if it's a strong business requirement to use a single-identity provider. Federation is more complex to operate and troubleshoot compared to cloud authentication solutions.

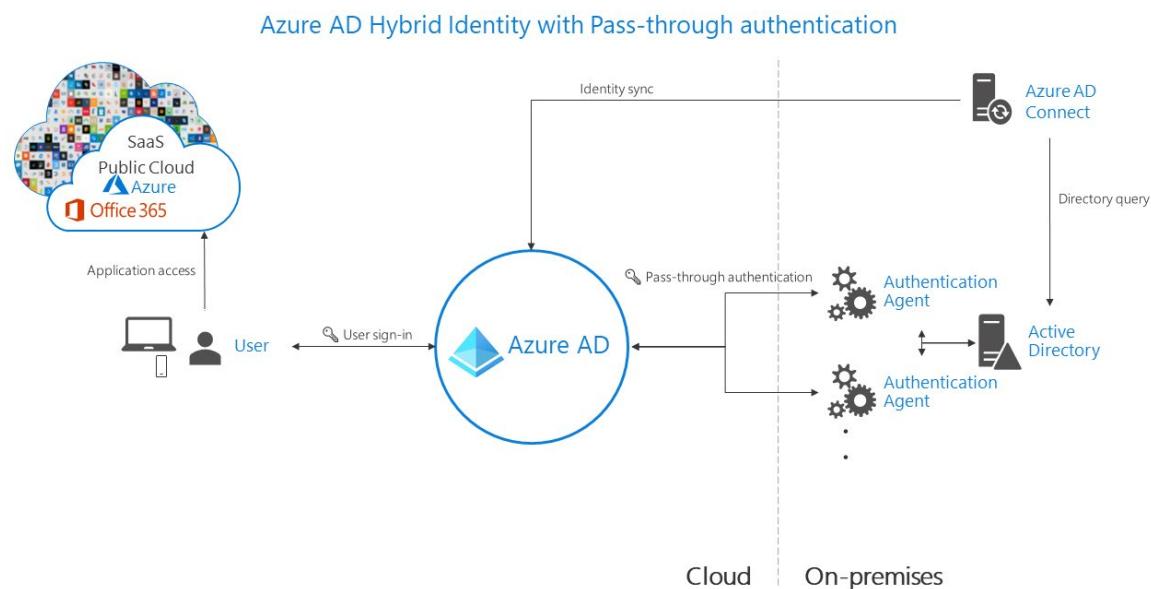
Architecture diagrams

The following diagrams outline the high-level architecture components required for each authentication method you can use with your Azure AD hybrid identity solution. They provide an overview to help you compare the differences between the solutions.

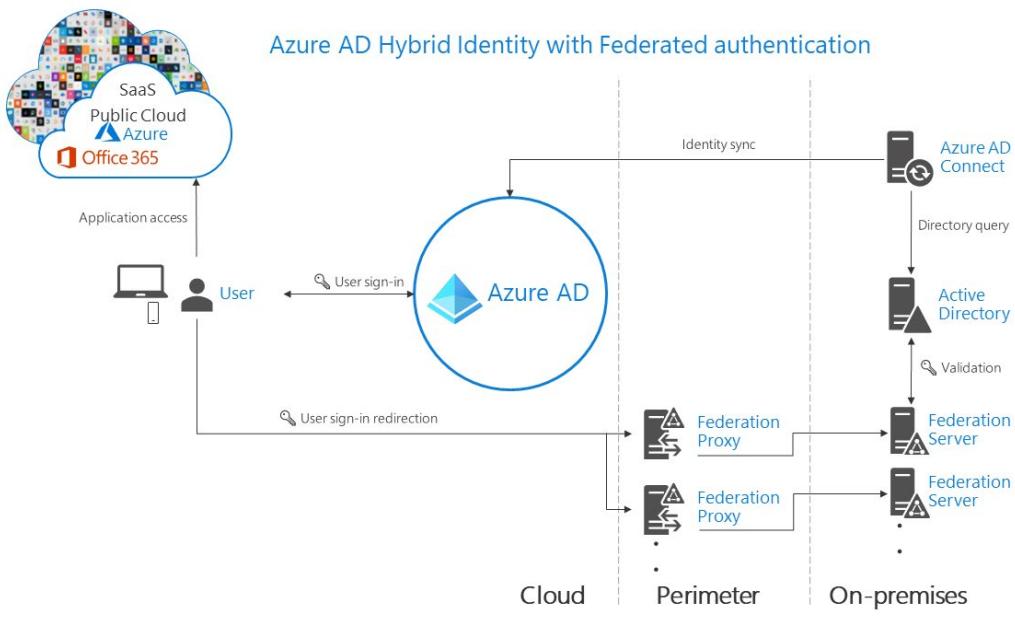
- Simplicity of a password hash synchronization solution:



- Agent requirements of pass-through authentication, using two agents for redundancy:



- Components required for federation in your perimeter and internal network of your organization:



Recommendations

Your identity system ensures your users' access to cloud apps and the line-of-business apps that you migrate and make available in the cloud. To keep authorized users productive and bad actors out of your organization's sensitive data, authentication controls access to apps.

Use or enable password hash synchronization for whichever authentication method you choose, for the following reasons:

- High availability and disaster recovery.** Pass-through authentication and federation rely on on-premises infrastructure. For pass-through authentication, the on-premises footprint includes the server hardware and networking the pass-through authentication agents require. For federation, the on-premises footprint is even larger. It requires servers in your perimeter network to proxy authentication requests and the internal federation servers.

To avoid single points of failure, deploy redundant servers. Then authentication requests will always be serviced if any component fails. Both pass-through authentication and federation also rely on domain controllers to respond to authentication requests, which can also fail. Many of these components need maintenance to stay healthy. Outages are more likely when maintenance isn't planned and implemented correctly. Avoid outages by using password hash synchronization because the Microsoft Azure AD cloud authentication service scales globally and is always available.

- On-premises outage survival.** The consequences of an on-premises outage due to a cyber-attack or disaster can be substantial, ranging from reputational brand damage to a paralyzed organization unable to deal with the attack. Recently, many organizations were victims of malware attacks, including targeted ransomware, which caused their on-premises servers to go down. When Microsoft helps customers deal with these kinds of attacks, it sees two categories of organizations:
 - Organizations that previously also turned on password hash synchronization on top of federated or pass-through authentication changed their primary authentication method to then use password hash synchronization. They were back online in a matter of hours. By using access to email via Microsoft 365, they worked to resolve issues and access other cloud-based workloads.

- Organizations that didn't previously enable password hash synchronization had to resort to untrusted external consumer email systems for communications to resolve issues. In those cases, it took them weeks to restore their on-premises identity infrastructure before users were able to sign in to cloud-based apps again.
3. **Identity protection.** One of the best ways to protect users in the cloud is Azure AD Identity Protection with Azure AD Premium P2. Microsoft continually scans the Internet for user and password lists that bad actors sell and make available on the dark web. Azure AD can use this information to verify if any of the usernames and passwords in your organization are compromised. Therefore, it's critical to enable password hash synchronization no matter which authentication method you use, whether it's federated or pass-through authentication. Leaked credentials are presented as a report. Use this information to block or force users to change their passwords when they try to sign in with leaked passwords.

Azure AD Connect design concepts

This section describes areas that must be thought through during the implementation design of Azure AD Connect. It is a deep dive on certain areas and these concepts are briefly described in other documents as well.

sourceAnchor

The sourceAnchor attribute is defined as *an attribute immutable during the lifetime of an object*. It uniquely identifies an object as being the same object on-premises and in Azure AD. The attribute is also called **immutableId** and the two names are used interchangeable. The attribute is used for the following scenarios:

- When a new sync engine server is built, or rebuilt after a disaster recovery scenario, this attribute links existing objects in Azure AD with objects on-premises.
- If you move from a cloud-only identity to a synchronized identity model, then this attribute allows objects to "hard match" existing objects in Azure AD with on-premises objects.
- If you use federation, then this attribute together with the **userPrincipalName** is used in the claim to uniquely identify a user.

The attribute value must follow the following rules:

- Fewer than 60 characters in length
 - Characters not being a-z, A-Z, or 0-9 are encoded and counted as 3 characters
 - Not contain a special character: \ ! # \$ % & * + / = ? ^ ` { } | ~ () ' ; : , [] " @ _
 - Must be globally unique
 - Must be either a string, integer, or binary
 - Should not be based on user's name because these can change
 - Should not be case-sensitive and avoid values that may vary by case
 - Should be assigned when the object is created

If you have a single forest on-premises, the attribute you should use is **objectGUID**. This is also the attribute used when you use express settings in Azure AD Connect and also the attribute used by DirSync. If you have multiple forests and do not move users between forests and domains, then **objectGUID** is a good attribute to use even in this case. Another solution is to pick an existing attribute you know does

not change. Commonly used attributes include **employeeID**. If you consider an attribute that contains letters, make sure there is no chance the case (upper case vs. lower case) can change for the attribute's value. Bad attributes that should not be used include those attributes with the name of the user. Once the sourceAnchor attribute is decided, the wizard stores the information in your Azure AD tenant. The information will be used by future installation of Azure AD Connect.

Azure AD sign-in

While integrating your on-premises directory with Azure AD, synchronization settings can affect the way user authenticates. Azure AD uses userPrincipalName (UPN) to authenticate the user. However, when you synchronize your users, you must choose the attribute to be used for value of userPrincipalName carefully. When you are selecting the attribute for providing the value of UPN to be used in Azure one should ensure

- The attribute values conform to the UPN syntax (RFC 822), that is it should be of the format user-name@domain
- The suffix in the values matches to one of the verified custom domains in Azure AD

In express settings, the assumed choice for the attribute is userPrincipalName. If the userPrincipalName attribute does not contain the value you want your users to sign in to Azure, then you must choose **Custom Installation**.

Custom domain state and User Principal Name

Ensure that there is a verified domain for the User Principal Name (UPN) suffix. John is a user in contoso.com. You want John to use the on-premises UPN john@contoso.com to sign in to Azure after you have synced users to your Azure AD directory contoso.onmicrosoft.com. To do so, you need to add and verify contoso.com as a custom domain in Azure AD before you can start syncing the users. If the UPN suffix of John, for example contoso.com, does not match a verified domain in Azure AD, then Azure AD replaces the UPN suffix with contoso.onmicrosoft.com.

Some organizations have non-routable domains, like contoso.local, or simple single label domains like contoso. You are not able to verify a non-routable domain in Azure AD. Azure AD Connect can sync to only a verified domain in Azure AD. When you create an Azure AD directory, it creates a routable domain that becomes default domain for your Azure AD for example, contoso.onmicrosoft.com. Therefore, it becomes necessary to verify any other routable domain in such a scenario in case you don't want to sync to the default onmicrosoft.com domain.

Azure AD Connect detects if you are running in a non-routable domain environment and would appropriately warn you from going ahead with express settings. If you are operating in a non-routable domain, then it is likely that the UPN, of the users, have non-routable suffixes too. For example, if you are running under contoso.local, Azure AD Connect suggests you use custom settings rather than using express settings. Using custom settings, you are able to specify the attribute that should be used as UPN to sign in to Azure after the users are synced to Azure AD.

Topologies for Azure AD Connect

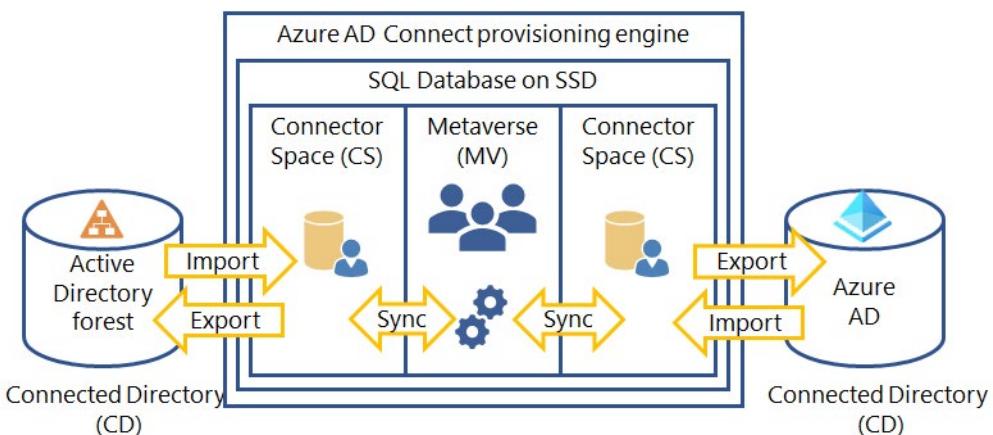
This section describes various on-premises and Azure Active Directory (Azure AD) topologies that use Azure AD Connect sync as the key integration solution; it includes both supported and unsupported configurations.

Common topology	Description
Single forest, single Azure AD tenant	The most common topology is a single on-premises forest, with one or multiple domains, and a single Azure AD tenant. For Azure AD authentication, password hash synchronization is used. The express installation of Azure AD Connect supports only this topology.
Multiple forests, single Azure AD tenant	Many organizations have environments with multiple on-premises Active Directory forests. There are various reasons for having more than one on-premises Active Directory forest. Typical examples are designs with account-resource forests and the result of a merger or acquisition. When you have multiple forests, all forests must be reachable by a single Azure AD Connect sync server. The server must be joined to a domain. If necessary to reach all forests, you can place the server in a perimeter network (also known as DMZ, demilitarized zone, and screened subnet).
Multiple forests, single sync server, users are represented in only one directory	In this environment, all on-premises forests are treated as separate entities. No user is present in any other forest. Each forest has its own Exchange organization, and there's no GALSync between the forests. This topology might be the situation after a merger/acquisition or in an organization where each business unit operates independently. These forests are in the same organization in Azure AD and appear with a unified GAL. In the preceding picture, each object in every forest is represented once in the metaverse and aggregated in the target Azure AD tenant.
Multiple forests: full mesh with optional GALSync	A full mesh topology allows users and resources to be located in any forest. Commonly, there are two-way trusts between the forests. If Exchange is present in more than one forest, there might be (optionally) an on-premises GALSync solution. Every user is then represented as a contact in all other forests. GALSync is commonly implemented through FIM 2010 or MIM 2016. Azure AD Connect cannot be used for on-premises GALSync.
Multiple forests: account-resource forest	In this scenario, one (or more) resource forest trusts all account forests. The resource forest typically has an extended Active Directory schema with Exchange and Lync. All Exchange and Lync services, along with other shared services, are located in this forest. Users have a disabled user account in this forest, and the mailbox is linked to the account forest.

Common topology	Description
Staging server	Azure AD Connect supports installing a second server in <i>staging mode</i> . A server in this mode reads data from all connected directories but does not write anything to connected directories. It uses the normal synchronization cycle and therefore has an updated copy of the identity data.
Multiple Azure AD tenants	There's a 1:1 relationship between an Azure AD Connect sync server and an Azure AD tenant. For each Azure AD tenant, you need one Azure AD Connect sync server installation. The Azure AD tenant instances are isolated by design. That is, users in one tenant can't see users in the other tenant. If you want this separation, this is a supported configuration. Otherwise, you should use the single Azure AD tenant model.
Each object only once in an Azure AD tenant	In this topology, one Azure AD Connect sync server is connected to each Azure AD tenant. The Azure AD Connect sync servers must be configured for filtering so that each has a mutually exclusive set of objects to operate on. You can, for example, scope each server to a particular domain or organizational unit.

Azure AD Connect component factors

The following diagram shows a high-level architecture of provisioning engine connecting to a single forest, although multiple forests are supported. This architecture shows how the various components interact with each other.



The provisioning engine connects to each Active Directory forest and to Azure AD. The process of reading information from each directory is called Import. Export refers to updating the directories from the provisioning engine. Sync evaluates the rules of how the objects will flow inside the provisioning engine.

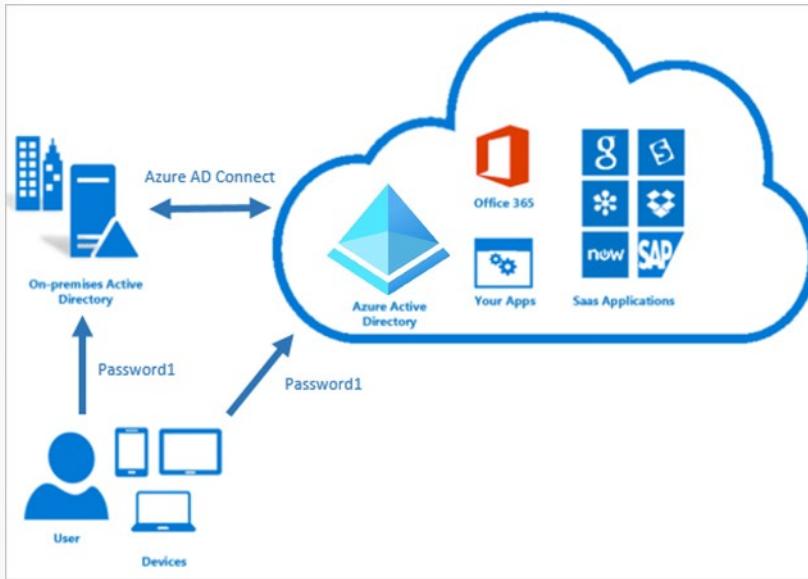
Azure AD Connect uses the following staging areas, rules, and processes to allow the sync from Active Directory to Azure AD:

- **Connector Space (CS)** - Objects from each connected directory (CD), the actual directories, are staged here first before they can be processed by the provisioning engine. Azure AD has its own CS and each forest you connect to has its own CS.
- **Metaverse (MV)** - Objects that need to be synced are created here based on the sync rules. Objects must exist in the MV before they can populate objects and attributes to the other connected directories. There's only one MV.
- **Sync rules** - They decide which objects will be created (projected) or connected (joined) to objects in the MV. The sync rules also decide which attribute values will be copied or transformed to and from the directories.
- **Run profiles** - Bundles the process steps of copying objects and their attribute values according to the sync rules between the staging areas and connected directories.

Password Hash Synchronization

How password hash synchronization works

Password hash synchronization is one of the sign-in methods used to accomplish hybrid identity. Azure AD Connect synchronizes a hash, of the hash, of a user's password from an on-premises Active Directory instance to a cloud-based Azure AD instance.



Active Directory Domain Services stores passwords in the form of a hash value representation of the actual user password. A hash value is a result of a one-way mathematical function (the hashing algorithm). There is no method to revert the result of a one-way function to the plain text version of a password. To synchronize your password, Azure AD Connect sync extracts your password hash from the on-premises Active Directory instance. Extra security processing is applied to the password hash before it is synchronized to the Azure Active Directory authentication service. Passwords are synchronized on a per-user basis and in chronological order.

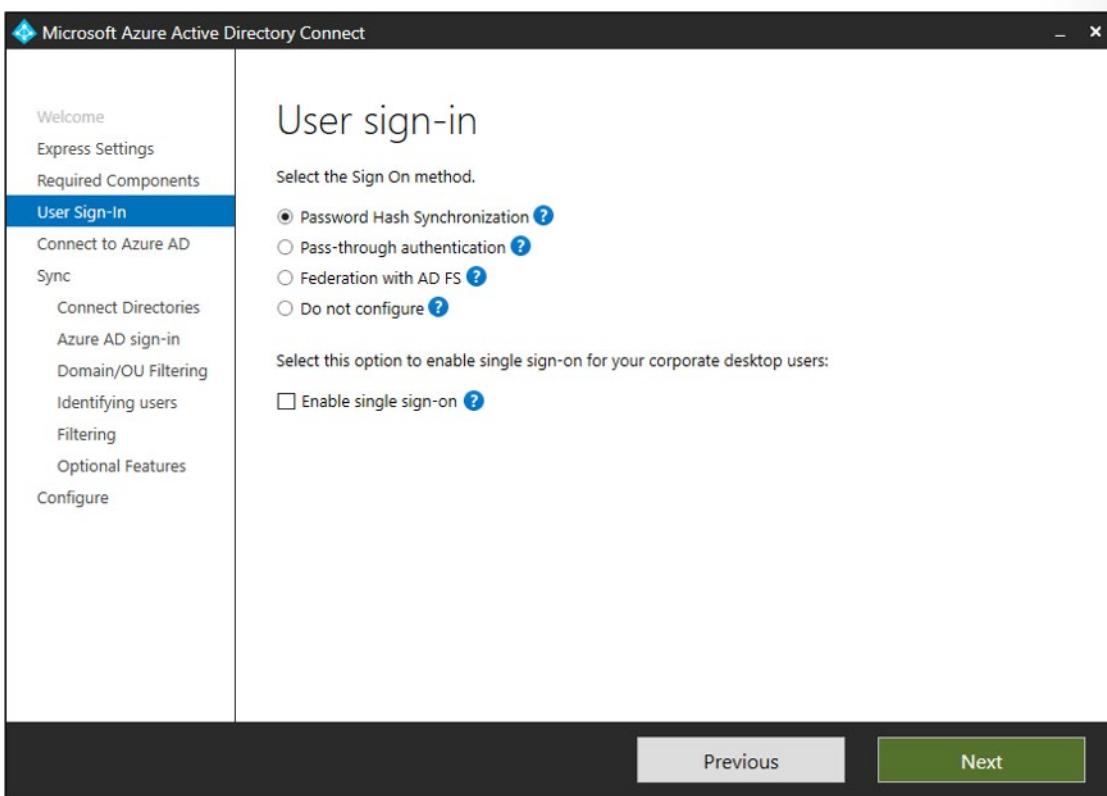
The actual data flow of the password hash synchronization process is similar to the synchronization of user data. However, passwords are synchronized more frequently than the standard directory synchroni-

zation window for other attributes. The password hash synchronization process runs every 2 minutes. You cannot modify the frequency of this process. When you synchronize a password, it overwrites the existing cloud password.

The first time you enable the password hash synchronization feature, it performs an initial synchronization of the passwords of all in-scope users. You cannot explicitly define a subset of user passwords that you want to synchronize. However, if there are multiple connectors, it is possible to disable password hash sync for some connectors but not others. When you change an on-premises password, the updated password is synchronized, most often in a matter of minutes. The password hash synchronization feature automatically retries failed synchronization attempts. If an error occurs during an attempt to synchronize a password, an error is logged in your event viewer.

Enable password hash synchronization

When you install Azure AD Connect by using the **Express Settings** option, password hash synchronization is automatically enabled. If you use custom settings when you install Azure AD Connect, password hash synchronization is available on the user sign-in page.



Password hash synchronization and Federal Information Processing standard

If your server has been locked down according to Federal Information Processing Standard (FIPS), then MD5 is disabled.

To enable MD5 for password hash synchronization, perform the following steps:

1. Go to %programfiles%\Azure AD Sync\Bin.

2. Open miiserver.exe.config.
3. Go to the configuration/runtime node at the end of the file.
4. Add the following node: <enforceFIPSPolicy enabled="false">
5. Save your changes.

For reference, this snippet is what it should look like:

```
<configuration>
    <runtime>
        <enforceFIPSPolicy enabled="false">
    </runtime>
</configuration>
```

Pass Through Authentication

Azure Active Directory (Azure AD) pass-through authentication allows your users to sign in to both on-premises and cloud-based applications by using the same passwords. Pass-through authentication signs users in by validating their passwords directly against on-premises Active Directory.

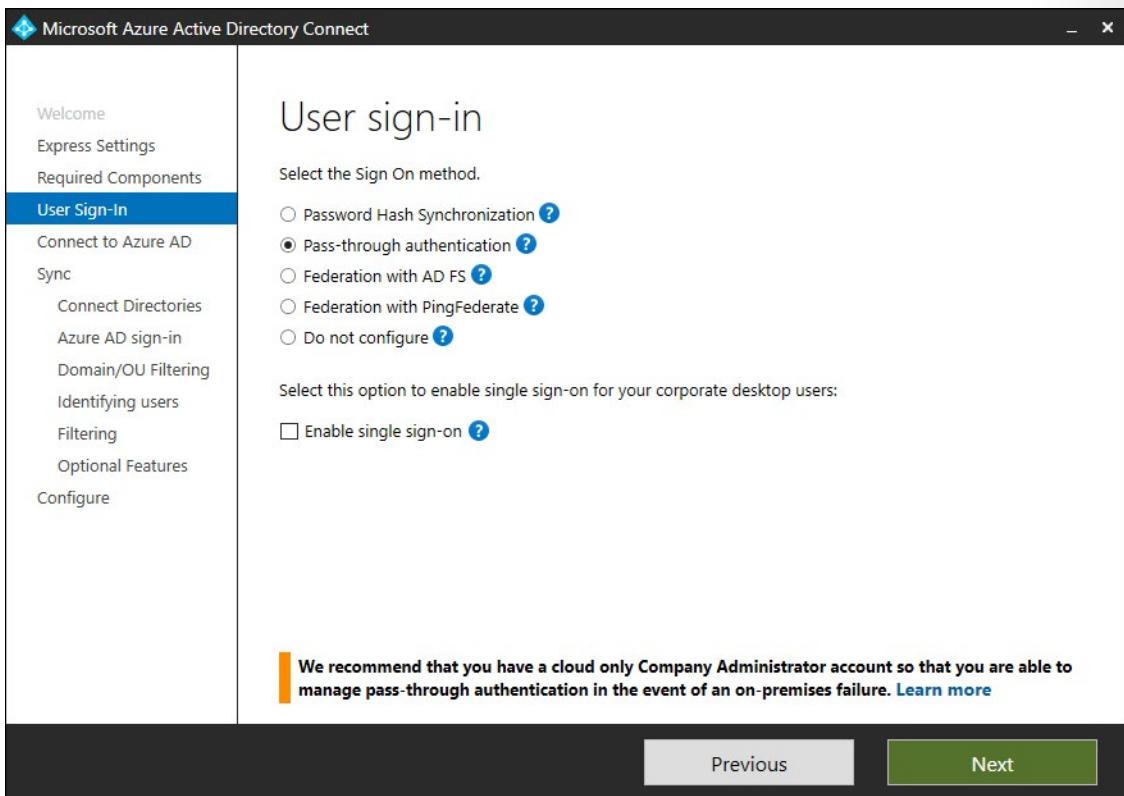
Enable the feature

Enable pass-through authentication through **Azure AD Connect²⁵**.

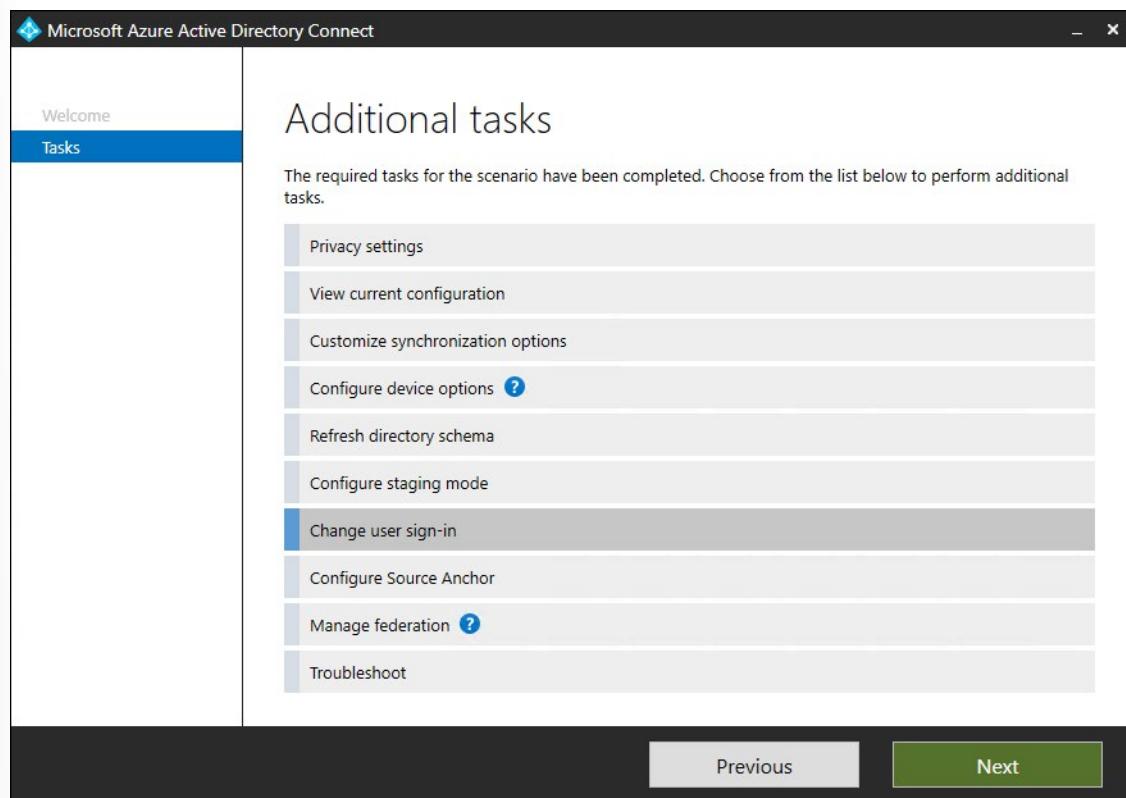
If you're installing Azure AD Connect for the first time, choose the **custom installation path²⁶**. At the **User sign-in** page, choose **Pass-through authentication** as the **Sign On method**. On successful completion, a pass-through authentication agent is installed on the same server as Azure AD Connect. In addition, the pass-through authentication feature is enabled on your tenant.

²⁵ <https://docs.microsoft.com/azure/active-directory/hybrid/whatis-hybrid-identity>

²⁶ <https://docs.microsoft.com/azure/active-directory/hybrid/how-to-connect-install-custom>



If you have already installed Azure AD Connect by using the express installation or the custom installation path, select the **Change user sign-in** task on Azure AD Connect, and then select **Next**. Then select **Pass-through authentication** as the sign-in method. On successful completion, a pass-through authentication agent is installed on the same server as Azure AD Connect and the feature is enabled on your tenant.



Important - Pass-through authentication is a tenant-level feature. Turning it on affects the sign-in for users across all the managed domains in your tenant. If you're switching from Active Directory Federation Services (AD FS) to Pass-through authentication, you should wait at least 12 hours before shutting down your AD FS infrastructure. This wait time is to ensure that users can keep signing in to Exchange ActiveSync during the transition.

Watch the Demo - Manage passthrough authentication for Seamless Single Sign-on

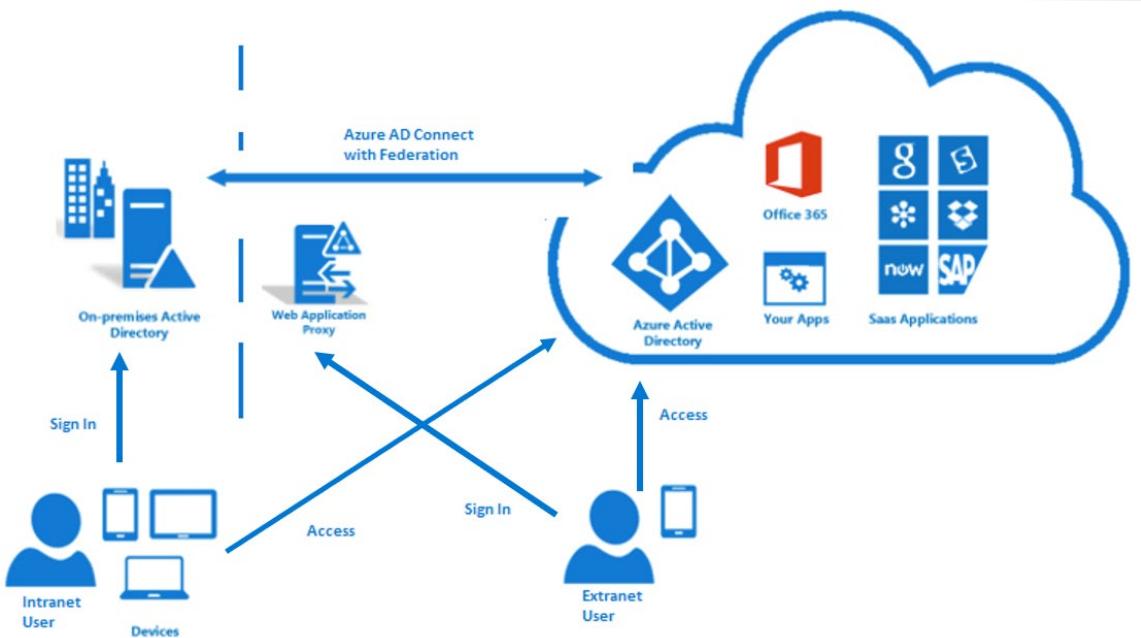
Azure AD seamless single sign-on (Seamless SSO) automatically signs in users from their network-connected corporate desktops. Seamless SSO provides your users with easy access to cloud-based applications without needing any other on-premises components.



<https://www.microsoft.com/videoplayer/embed/RE4Mzo1>

Federation

Federation can use a new or existing on-premises Active Directory farm in Windows Server 2012 R2 (or later), and Azure AD Connect enable users to log into Azure AD resources using their on-premises password.



Federation is a collection of domains that have established trust. The level of trust may vary, but typically includes authentication and almost always includes authorization. A typical federation might include a number of organizations that have established trust for shared access to a set of resources.

You can federate your on-premises environment with Azure AD and use this federation for authentication and authorization. This sign-in method ensures that all user authentication occurs on-premises. This method allows administrators to implement more rigorous levels of access control. Federation with AD FS and PingFederate is available.

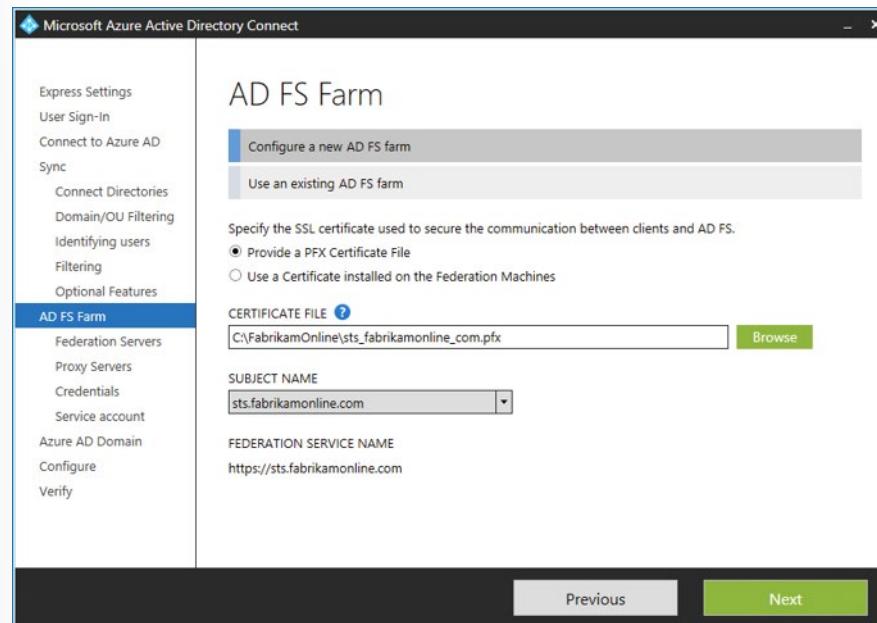
With federated sign-in, your users can sign in to Azure AD-based services with their on-premises passwords. While they're on the corporate network, they don't even have to enter their passwords. By using the federation option with AD FS, you can deploy a new or existing farm with AD FS in Windows Server 2012 R2. If you choose to specify an existing farm, Azure AD Connect configures the trust between your farm and Azure AD so that your users can sign in.

Requirement to deploy federation with AD FS and Azure AD Connect

Deploying to an AD FS farm, you need:

- Local administrator credentials on your federation servers.
- Local administrator credentials on any workgroup servers (not domain-joined) that you intend to deploy the Web Application Proxy role on.
- The machine that you run the wizard on to be able to connect to any other machines that you want to install AD FS or Web Application Proxy on by using Windows Remote Management.

Set up your federation using Azure AD Connect to connect to an AD FS farm



Specify the AD FS servers

Specify the servers where you want to install AD FS. You can add one or more servers, depending on your

capacity needs. Before you set up this configuration, join all AD FS servers to Active Directory. This step isn't required for the Web Application Proxy servers. Microsoft recommends installing a single AD FS server for test and pilot deployments. After the initial configuration, you can add and deploy more servers to meet your scaling needs by running Azure AD Connect again.

Specify the Web Application Proxy servers

Specify your Web Application Proxy servers. The Web Application Proxy server is deployed in your perimeter network, facing the extranet. It supports authentication requests from the extranet. You can add one or more servers, depending on your capacity needs. After the initial configuration, you can add and deploy more servers to meet your scaling needs by running Azure AD Connect again.

Specify the service account for the AD FS service

The AD FS service requires a domain service account to authenticate users and to look up user information in Active Directory. It can support two types of service accounts:

- Group managed service account
- Domain user account

Select the Azure AD domain that you want to federate

Use the Azure AD Domain page to set up the federation relationship between AD FS and Azure AD. Here, you configure AD FS to provide security tokens to Azure AD. You also configure Azure AD to trust the tokens from this AD FS instance. On this page, you can configure only a single domain in the initial installation. You can configure more domains later by running Azure AD Connect again.

Azure AD Connect tools to manage your federation

You can complete various AD FS-related tasks in Azure AD Connect with minimal user intervention by using the Azure AD Connect wizard. Even after you've finished installing Azure AD Connect by running the wizard, you can run the wizard again to do other tasks. For example, you can use the wizard to repair the trust with Microsoft 365, federate with Azure AD using alternate login ID, and add an AD FS Web Application Proxy (WAP) server.

Repair the trust

You can use Azure AD Connect to check the current health of the AD FS and Azure AD trust and take appropriate actions to repair the trust.

Federate with Azure AD using AlternateID

It is recommended that the on-premises User Principal Name(UPN) and the cloud User Principal Name are kept the same. If the on-premises UPN uses a non-routable domain (ex. Contoso.local) or cannot be changed due to local application dependencies, we recommend setting up alternate login ID. Alternate login ID allows you to configure a sign-in experience where users can sign in with an attribute other than their UPN, such as mail. The choice for User Principal Name in Azure AD Connect defaults to the userPrincipalName attribute in Active Directory. If you choose any other attribute for User Principal Name and are federating using AD FS, then Azure AD Connect will configure AD FS for alternate login ID.

Add a federated domain

It's easy to add a domain to be federated with Azure AD by using Azure AD Connect. Azure AD Connect adds the domain for federation and modifies the claim rules to correctly reflect the issuer when you have multiple domains federated with Azure AD.

Along with **Add and AD FS Server** and **Add an AD FS Web Application Proxy server**.

Device writeback

Device writeback is used to enable device-based conditional access for ADFS-protected devices. This conditional access provides extra security and assurance that access to applications is granted only to trusted devices. Device writeback enables this security by synchronizing all devices registered in Azure back to the on-premises Active Directory. When configured during setup, the following operations are performed to prepare the AD forest:

- If they do not exist already, create and configure new containers and objects under: CN=Device Registration Configuration, CN=Services, CN=Configuration, [forest dn].
- If they do not exist already, create and configure new containers and objects under: CN=RegisteredDevices, [domain-dn]. Device objects will be created in this container.
- Set necessary permissions on the Azure AD Connector account, to manage devices on your Active Directory.

The commands only needs to run on one forest, even if Azure AD Connect is being installed in multiple forests.

Troubleshoot Synchronization Errors

Errors could occur when identity data is synchronized from Windows Server Active Directory (AD DS) to Azure Active Directory (Azure AD). This section provides an overview of different types of sync errors, some of the possible scenarios that cause those errors and potential ways to fix the errors. This section includes the common error types and may not cover all the possible errors.

With the latest version of Azure AD Connect (August 2016 or higher), a report of Synchronization Errors is available in the [Azure portal](#)²⁷ as part of Azure AD Connect Health for sync.

Starting September 1, 2016 **Azure Active Directory Duplicate Attribute Resiliency**²⁸ feature will be enabled by default for all the *new* Azure Active Directory Tenants. This feature will be automatically enabled for existing tenants in the upcoming months.

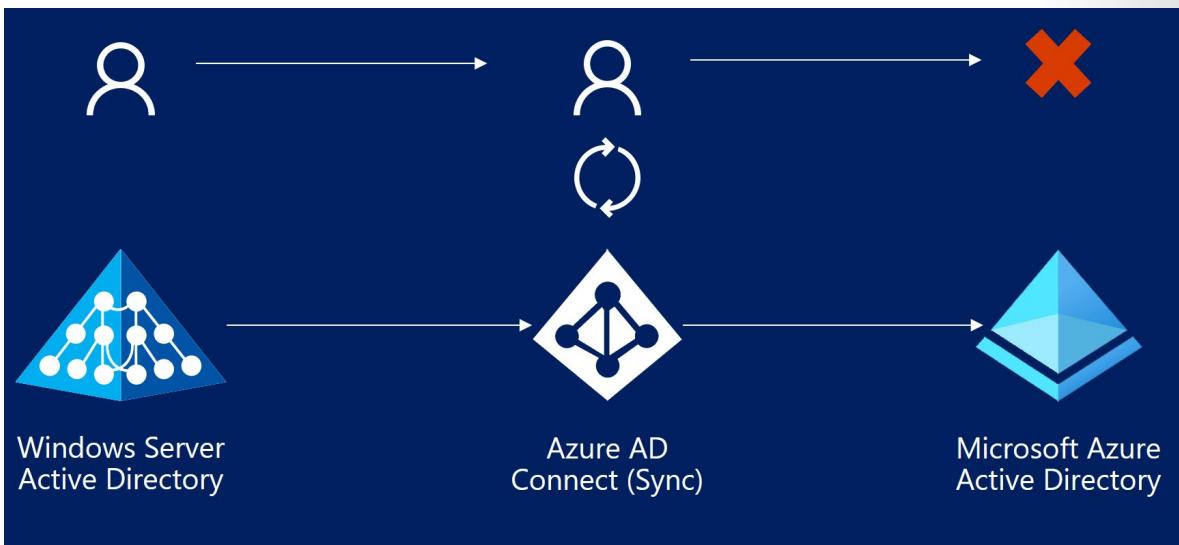
Azure AD Connect performs three types of operations from the directories it keeps in sync: Import, Synchronization, and Export. Errors can take place in all the operations. This section mainly focuses on errors during Export to Azure AD.

Errors during export to Azure AD

The following section describes different types of synchronization errors that can occur during the export operation to Azure AD using the Azure AD connector. This connector can be identified by the name format being contoso.onmicrosoft.com. Errors during export to Azure AD indicate that the operation (add, update, delete etc.) attempted by Azure AD Connect (Sync Engine) on Azure Active Directory failed.

²⁷ <https://aka.ms/aadconnecthealth>

²⁸ <https://docs.microsoft.com/azure/active-directory/hybrid/how-to-connect-sync-service-duplicate-attribute-resiliency>



Data mismatch errors

InvalidSoftMatch

Description

- When Azure AD Connect (sync engine) instructs Azure Active Directory to add or update objects, Azure AD matches the incoming object using the **sourceAnchor** attribute to the **immutableId** attribute of objects in Azure AD. This match is called a **Hard Match**.
- When Azure AD **does not find** any object that matches the **immutableId** attribute with the **sourceAnchor** attribute of the incoming object, before provisioning a new object, it falls back to use the **ProxyAddresses** and **UserPrincipalName** attributes to find a match. This match is called a **Soft Match**. The Soft Match is designed to match objects already present in Azure AD (that are sourced in Azure AD) with the new objects being added/updated during synchronization that represent the same entity (users, groups) on premises.
- InvalidSoftMatch** error occurs when the hard match does not find any matching object **AND** soft match finds a matching object but that object has a different value of *immutableId* than the incoming object's *SourceAnchor*, suggesting that the matching object was synchronized with another object from on premises Active Directory.

In other words, in order for the soft match to work, the object to be soft-matched with should not have any value for the *immutableId*. If any object with *immutableId* set with a value is failing the hard-match but satisfying the soft-match criteria, the operation would result in an InvalidSoftMatch synchronization error.

Azure Active Directory schema does not allow two or more objects to have the same value of the following attributes. (This is not an exhaustive list.)

- **ProxyAddresses**
- **UserPrincipalName**
- **onPremisesSecurityIdentifier**

- ObjectId

Azure AD Attribute Duplicate Attribute Resiliency²⁹ feature is also being rolled out as the default behavior of Azure Active Directory. This will reduce the number of synchronization errors seen by Azure AD Connect (as well as other sync clients) by making Azure AD more resilient in the way it handles duplicated ProxyAddresses and UserPrincipalName attributes present in on premises AD environments. This feature does not fix the duplication errors. So the data still needs to be fixed. But it allows provisioning of new objects which are otherwise blocked from being provisioned due to duplicated values in Azure AD. This will also reduce the number of synchronization errors returned to the synchronization client. If this feature is enabled for your Tenant, you will not see the InvalidSoftMatch synchronization errors seen during provisioning of new objects.

Example scenarios for InvalidSoftMatch

1. Two or more objects with the same value for the ProxyAddresses attribute exist in on-premises Active Directory. Only one is getting provisioned in Azure AD.
2. Two or more objects with the same value for the userPrincipalName attribute exists in on-premises Active Directory. Only one is getting provisioned in Azure AD.
3. An object was added in the on premises Active Directory with the same value of ProxyAddresses attribute as that of an existing object in Azure Active Directory. The object added on premises is not getting provisioned in Azure Active Directory.
4. An object was added in on premises Active Directory with the same value of userPrincipalName attribute as that of an account in Azure Active Directory. The object is not getting provisioned in Azure Active Directory.
5. A synced account was moved from Forest A to Forest B. Azure AD Connect (sync engine) was using ObjectGUID attribute to compute the SourceAnchor. After the forest move, the value of the SourceAnchor is different. The new object (from Forest B) is failing to sync with the existing object in Azure AD.
6. A synced object got accidentally deleted from on premises Active Directory and a new object was created in Active Directory for the same entity (such as user) without deleting the account in Azure Active Directory. The new account fails to sync with the existing Azure AD object.
7. Azure AD Connect was uninstalled and reinstalled. During the reinstallation, a different attribute was chosen as the SourceAnchor. All the objects that had previously synced stopped syncing with InvalidSoftMatch error.

Example case:

1. **Bob Smith** is a synced user in Azure Active Directory from on premises Active Directory of contoso.com
2. Bob Smith's **UserPrincipalName** is set as bobs@contoso.com.
3. "**abcdefghijklmnoprstuv==**" is the **SourceAnchor** calculated by Azure AD Connect using Bob Smith's objectGUID from on premises Active Directory, which is the **immutableId** for Bob Smith in Azure Active Directory.
4. Bob also has following values for the **proxyAddresses** attribute:

²⁹ <https://docs.microsoft.com/azure/active-directory/hybrid/how-to-connect-sync-service-duplicate-attribute-resiliency>

- smtp: bobs@contoso.com
 - smtp: bob.smith@contoso.com
 - smtp: bob@contoso.com
5. A new user, **Bob Taylor**, is added to the on premises Active Directory.
 6. Bob Taylor's **UserPrincipalName** is set as bobt@contoso.com.
 7. "**abcdefghijkl0123456789==**" is the **sourceAnchor** calculated by Azure AD Connect using Bob Taylor's **objectGUID** from on premises Active Directory. Bob Taylor's object has NOT synced to Azure Active Directory yet.
 8. Bob Taylor has the following values for the proxyAddresses attribute
 - smtp: bobt@contoso.com
 - smtp: bob.taylor@contoso.com
 - smtp: bob@contoso.com
 9. During sync, Azure AD Connect will recognize the addition of Bob Taylor in on premises Active Directory and ask Azure AD to make the same change.
 10. Azure AD will first perform hard match. That is, it will search if there is any object with the immutableId equal to "abcdefghijkl0123456789==". Hard Match will fail as no other object in Azure AD will have that immutableId.
 11. Azure AD will then attempt to soft-match Bob Taylor. That is, it will search if there is any object with proxyAddresses equal to the three values, including smtp: bob@contoso.com
 12. Azure AD will find Bob Smith's object to match the soft-match criteria. But this object has the value of immutableId = "abcdefghijklmnoprstuv==". which indicates this object was synced from another object from on premises Active Directory. Thus, Azure AD cannot soft-match these objects and results in an **InvalidSoftMatch** sync error.

How to fix InvalidSoftMatch error

The most common reason for the InvalidSoftMatch error is two objects with different SourceAnchor (immutableId) have the same value for the ProxyAddresses and/or UserPrincipalName attributes, which are used during the soft-match process on Azure AD. In order to fix the Invalid Soft Match

1. Identify the duplicated proxyAddresses, userPrincipalName, or other attribute value that's causing the error. Also identify which two (or more) objects are involved in the conflict. The report generated by **Azure AD Connect Health for sync**³⁰ can help you identify the two objects.
2. Identify which object should continue to have the duplicated value and which object should not.
3. Remove the duplicated value from the object that should NOT have that value. You should make the change in the directory where the object is sourced from. In some cases, you may need to delete one of the objects in conflict.
4. If you made the change in the on premises AD, let Azure AD Connect sync the change.

Sync error reports within Azure AD Connect Health for sync are updated every 30 minutes and include the errors from the latest synchronization attempt.

Note - ImmutableId, by definition, should not change in the lifetime of the object. If Azure AD Connect was not configured with some of the scenarios in mind from the above list, you could end up in a

³⁰ <https://docs.microsoft.com/azure/active-directory/hybrid/how-to-connect-health-sync>

situation where Azure AD Connect calculates a different value of the SourceAnchor for the AD object that represents the same entity (same user/group/contact etc) that has an existing Azure AD Object that you wish to continue using.

ObjectTypeMismatch

When Azure AD attempts to soft match two objects, it is possible that two objects of different "object type" (such as User, Group, Contact etc.) have the same values for the attributes used to perform the soft match. As duplication of these attributes is not permitted in Azure AD, the operation can result in "ObjectTypeMismatch" synchronization error.

Example scenarios for ObjectTypeMismatch error

- A mail enabled security group is created in Microsoft 365. Admin adds a new user or contact in on premises AD (that's not synchronized to Azure AD yet) with the same value for the ProxyAddresses attribute as that of the Microsoft 365 group.

Example case

1. Admin creates a new mail enabled security group in Microsoft 365 for the Tax department and provides an email address as `tax@contoso.com`. This group is assigned the ProxyAddresses attribute value of `smtp: tax@contoso.com`
2. A new user joins `Contoso.com` and an account is created for the user on premises with the proxyAddress as `smtp: tax@contoso.com`
3. When Azure AD Connect will sync the new user account, it will get the "ObjectTypeMismatch" error.

How to fix ObjectTypeMismatch error

The most common reason for the ObjectTypeMismatch error is two objects of different type (User, Group, Contact etc.) have the same value for the ProxyAddresses attribute. In order to fix the ObjectTypeMismatch:

1. Identify the duplicated proxyAddresses (or other attribute) value that's causing the error. Also identify which two (or more) objects are involved in the conflict. The report generated by **Azure AD Connect Health for sync³¹** can help you identify the two objects.
2. Identify which object should continue to have the duplicated value and which object should not.
3. Remove the duplicated value from the object that should NOT have that value. You should make the change in the directory where the object is sourced from. In some cases, you may need to delete one of the objects in conflict.
4. If you made the change in the on premises AD, let Azure AD Connect sync the change. Sync error report within Azure AD Connect Health for sync gets updated every 30 minutes and includes the errors from the latest synchronization attempt.

³¹ <https://docs.microsoft.com/azure/active-directory/hybrid/how-to-connect-health-sync>

Duplicate attributes

AttributeValueMustBeUnique

Description

Azure Active Directory schema does not allow two or more objects to have the same value of the following attributes. That is each object in Azure AD is forced to have a unique value of these attributes at a given instance.

- ProxyAddresses
- UserPrincipalName

If Azure AD Connect attempts to add a new object or update an existing object with a value for the above attributes that is already assigned to another object in Azure Active Directory, the operation results in the "AttributeValueMustBeUnique" sync error.

Possible scenarios:

Duplicate value is assigned to an already synced object, which conflicts with another synced object.

Example case:

1. **Bob Smith** is a synced user in Azure Active Directory from on premises Active Directory of contoso.com
2. Bob Smith's **UserPrincipalName** on premises is set as bobs@contoso.com.
3. Bob also has following values for the **proxyAddresses** attribute:
 - smtp: bobs@contoso.com
 - smtp: bob.smith@contoso.com
 - smtp: bob@contoso.com
4. A new user, **Bob Taylor**, is added to the on premises Active Directory.
5. Bob Taylor's **UserPrincipalName** is set as bobt@contoso.com.
6. **Bob Taylor** has the following values for the **ProxyAddresses** attribute i. smtp: bobt@contoso.com ii. smtp: bob.taylor@contoso.com
7. Bob Taylor's object is synchronized with Azure AD successfully.
8. Admin decided to update Bob Taylor's **ProxyAddresses** attribute with the following value: i. smtp: bob@contoso.com
9. Azure AD will attempt to update Bob Taylor's object in Azure AD with the above value, but that operation will fail as that ProxyAddresses value is already assigned to Bob Smith, resulting in "AttributeValueMustBeUnique" error.

How to fix AttributeValueMustBeUnique error

The most common reason for the AttributeValueMustBeUnique error is two objects with different SourceAnchor (immutableId) have the same value for the ProxyAddresses and/or UserPrincipalName attributes. In order to fix AttributeValueMustBeUnique error

1. Identify the duplicated proxyAddresses, userPrincipalName or other attribute value that's causing the error. Also identify which two (or more) objects are involved in the conflict. The report generated by **Azure AD Connect Health for sync³²** can help you identify the two objects.
2. Identify which object should continue to have the duplicated value and which object should not.
3. Remove the duplicated value from the object that should NOT have that value. You should make the change in the directory where the object is sourced from. In some cases, you may need to delete one of the objects in conflict.
4. If you made the change in the on premises AD, let Azure AD Connect sync the change for the error to get fixed.

Data validation failures

IdentityDataValidationFailed

Azure Active Directory enforces various restrictions on the data itself before allowing that data to be written into the directory. These restrictions are to ensure that end users get the best possible experiences while using the applications that depend on this data.

Scenarios

The UserPrincipalName attribute value has invalid/unsupported characters. b. The UserPrincipalName attribute does not follow the required format.

How to fix IdentityDataValidationFailed error

Ensure that the userPrincipalName attribute has supported characters and required format.

FederatedDomainChangeError

Description

This case results in a “**FederatedDomainChangeError**” sync error when the suffix of a user's UserPrincipalName is changed from one federated domain to another federated domain.

Scenarios

For a synchronized user, the UserPrincipalName suffix was changed from one federated domain to another federated domain on premises. For example, UserPrincipalName = bob@contoso.com was changed to UserPrincipalName = bob@fabrikam.com.

³² <https://docs.microsoft.com/azure/active-directory/hybrid/how-to-connect-health-sync>

Example

1. Bob Smith, an account for Contoso.com, gets added as a new user in Active Directory with the UserPrincipalName bob@contoso.com
2. Bob moves to a different division of Contoso.com called Fabrikam.com and their UserPrincipalName is changed to bob@fabrikam.com
3. Both contoso.com and fabrikam.com domains are federated domains with Azure Active Directory.
4. Bob's userPrincipalName does not get updated and results in a "FederatedDomainChangeError" sync error.

How to fix

If a user's UserPrincipalName suffix was updated from bob@contoso.com to bob@fabrikam.com, where both contoso.com and fabrikam.com are **federated domains**, then follow these steps to fix the sync error

1. Update the user's UserPrincipalName in Azure AD from bob@contoso.com to bob@contoso.onmicrosoft.com. You can use the following PowerShell command with the Azure AD PowerShell Module: Set-MsolUserPrincipalName -UserPrincipalName bob@contoso.com -NewUserPrincipalName bob@contoso.onmicrosoft.com
2. Allow the next sync cycle to attempt synchronization. This time synchronization will be successful and it will update the UserPrincipalName of Bob to bob@fabrikam.com as expected.

LargeObject

When an attribute exceeds the allowed size limit, length limit or count limit set by Azure Active Directory schema, the synchronization operation results in the **LargeObject** or **ExceededAllowedLength** sync error. Typically this error occurs for the following attributes

- userCertificate
- userSMIMECertificate
- thumbnailPhoto
- proxyAddresses

Possible scenarios

1. Bob's userCertificate attribute is storing too many certificates assigned to Bob. These may include older, expired certificates. The hard limit is 15 certificates.
2. Bob's userSMIMECertificate attribute is storing too many certificates assigned to Bob. These may include older, expired certificates. The hard limit is 15 certificates.
3. Bob's thumbnailPhoto set in Active Directory is too large to be synced in Azure AD.
4. During automatic population of the ProxyAddresses attribute in Active Directory, an object has too many ProxyAddresses assigned.

How to fix

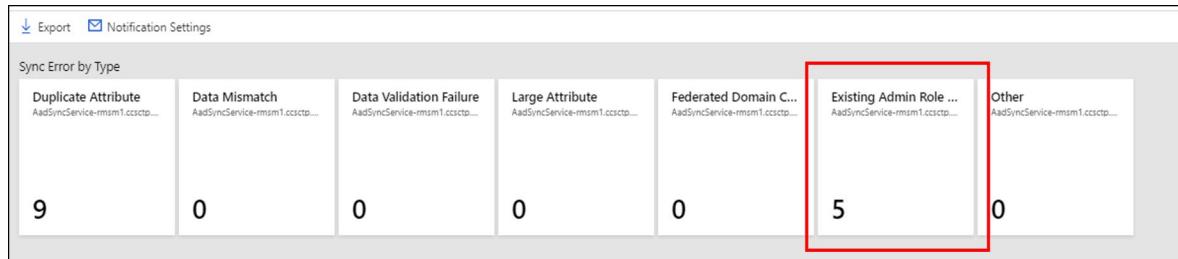
Ensure that the attribute causing the error is within the allowed limitation.

Admin role conflict

An **Existing Admin Role Conflict** will occur on a user object during synchronization when that user object has:

- administrative permissions and
- the same UserPrincipalName as an existing Azure AD object

Azure AD Connect is not allowed to soft match a user object from on-premises AD with a user object in Azure AD that has an administrative role assigned to it.



How to fix

To resolve this issue do the following:

1. Remove the Azure AD account (owner) from all admin roles.
2. **Hard Delete** the Quarantined object in the cloud.
3. The next sync cycle will take care of soft-matching the on-premises user to the cloud account (since the cloud user is now no longer a global GA).
4. Restore the role memberships for the owner.

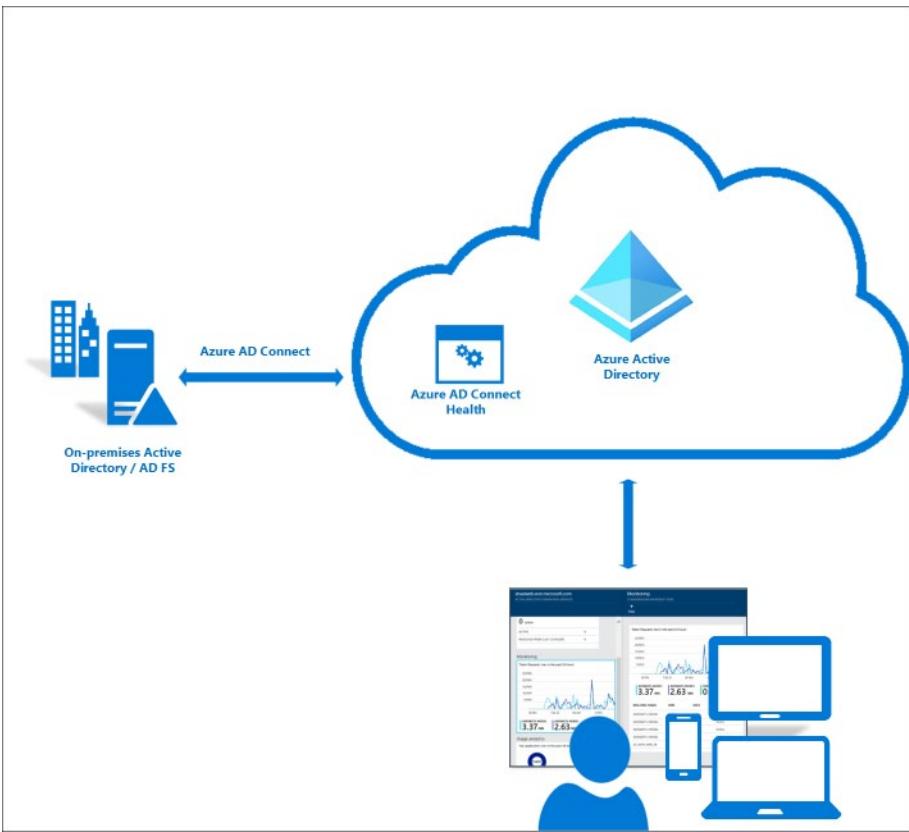
Note - You can assign the administrative role to the existing user object again after the soft match between the on-premises user object and the Azure AD user object has completed.

Implement Azure AD Connect Health

Azure Active Directory (Azure AD) Connect Health provides robust monitoring of your on-premises identity infrastructure. It enables you to maintain a reliable connection to Microsoft 365 and Microsoft Online Services. This reliability is achieved by providing monitoring capabilities for your key identity components. Also, it makes the key data points about these components easily accessible.

The information is presented in the **Azure AD Connect Health portal³³**. Use the Azure AD Connect Health portal to view alerts, performance monitoring, usage analytics, and other information. Azure AD Connect Health enables the single lens of health for your key identity components in one place.

³³ <https://aka.ms/aadconnecthealth>



Using the Azure AD Connect Health feature requires an Azure AD Premium P1 license.

Azure AD Connect Health agent installation

This section provides instructions for installing and configuring the Azure Active Directory (Azure AD) Connect Health agents.

Requirements

- Azure AD Premium is installed.
 - You're a global administrator in Azure AD.
 - The Azure AD Connect Health agent is installed on each targeted server.
 - The Azure service endpoints have outbound connectivity.
 - Outbound connectivity is based on IP addresses.
 - TLS inspection for outbound traffic is filtered or disabled.
 - Firewall ports on the server are running the agent.
-
- The agent requires the following firewall ports to be open so that it can communicate with the Azure AD Connect Health service endpoints: **TCP port 443** and **TCP port 5671**
 - The latest version of the agent doesn't require port 5671. Upgrade to the latest version so that only port 443 is required.

- PowerShell version 4.0 or newer is installed.
- FIPS (Federal Information Processing Standard) is disabled.

Install the agent

Download and install the Azure AD Connect Health agent from the Download Center.

Install the agent for Active Directory Federation Service

Note - Your Active Directory Federation Server (AD FS) server should be different from your Sync server. Don't install the AD FS agent on your Sync server.

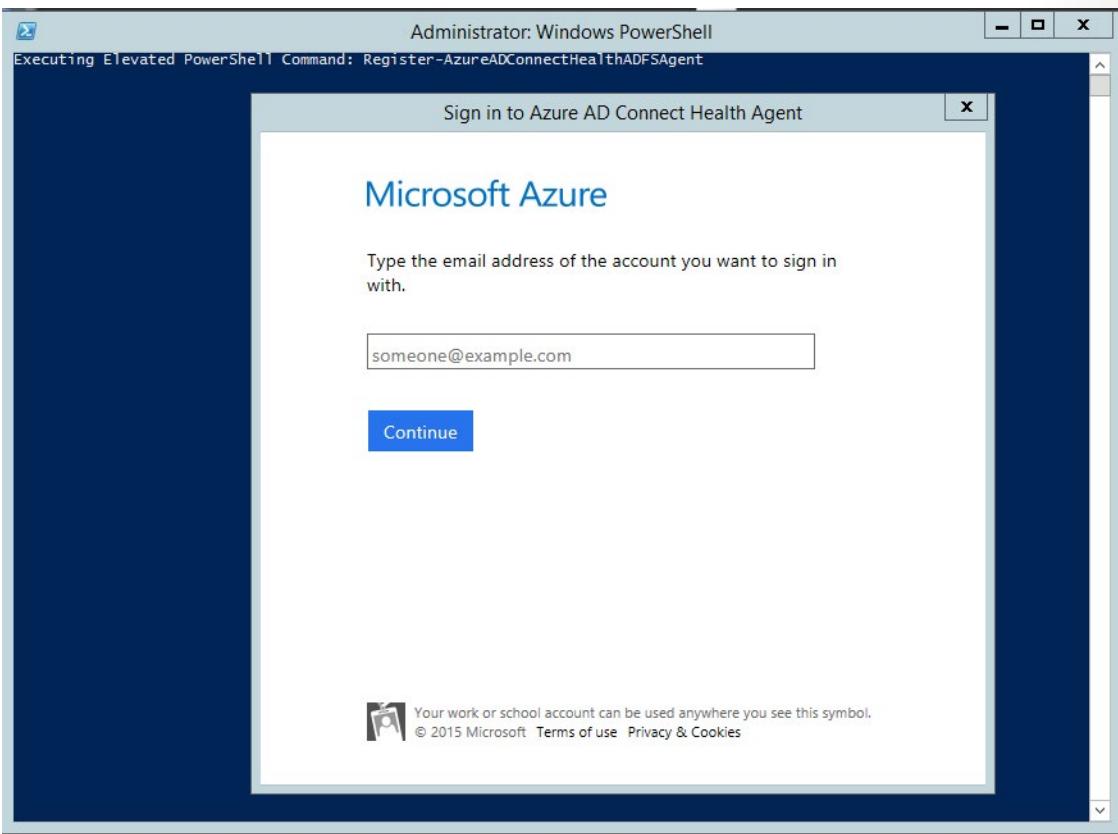
Before you install the agent, make sure your AD FS server host name is unique and isn't present in the AD FS service. To start the agent installation, double-click the .exe file that you downloaded. In the first window, select **Install**.



After the installation finishes, select **Configure Now**.



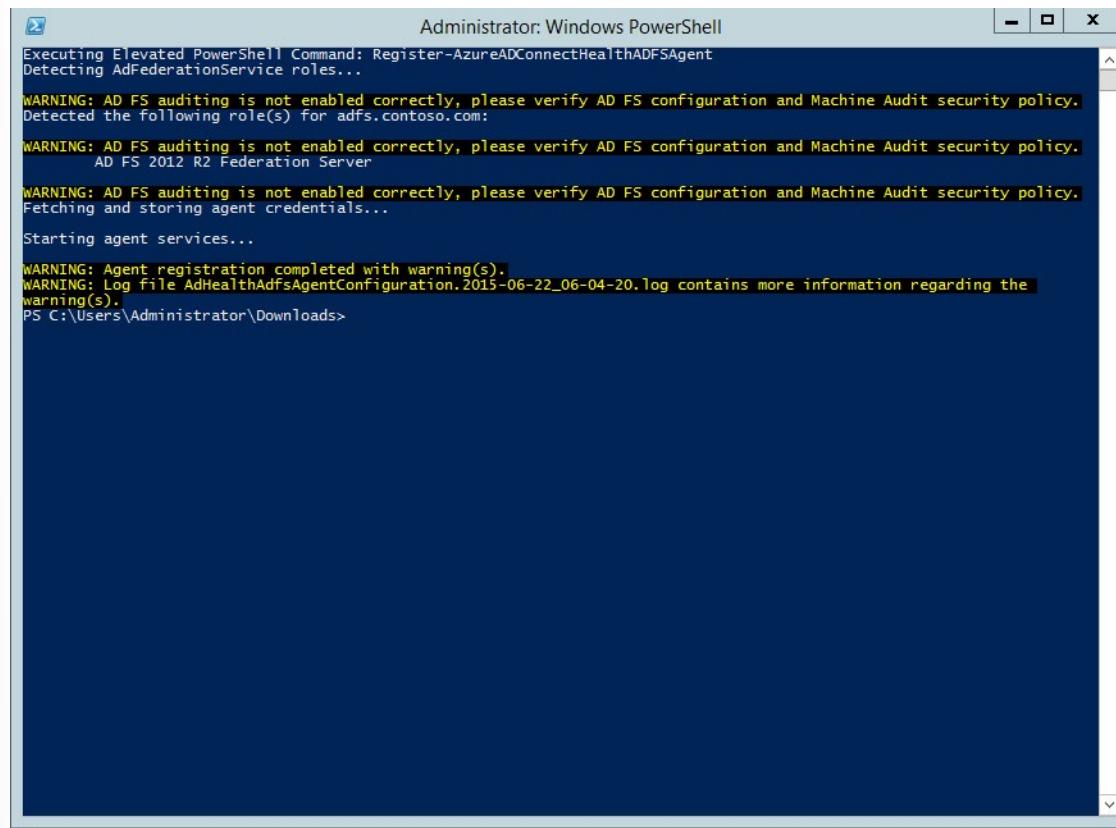
A PowerShell window opens to start the agent registration process. When you're prompted, sign in by using an Azure AD account that has permissions to register the agent. By default, the global admin account has permissions.



After you sign in, PowerShell continues. When it finishes, you can close PowerShell. The configuration is complete.

At this point, the agent services should start automatically to allow the agent to securely upload the required data to the cloud service.

If you haven't met all of the prerequisites, warnings appear in the PowerShell window. Be sure to complete the requirements before you install the agent. The following screenshot shows an example of these warnings.

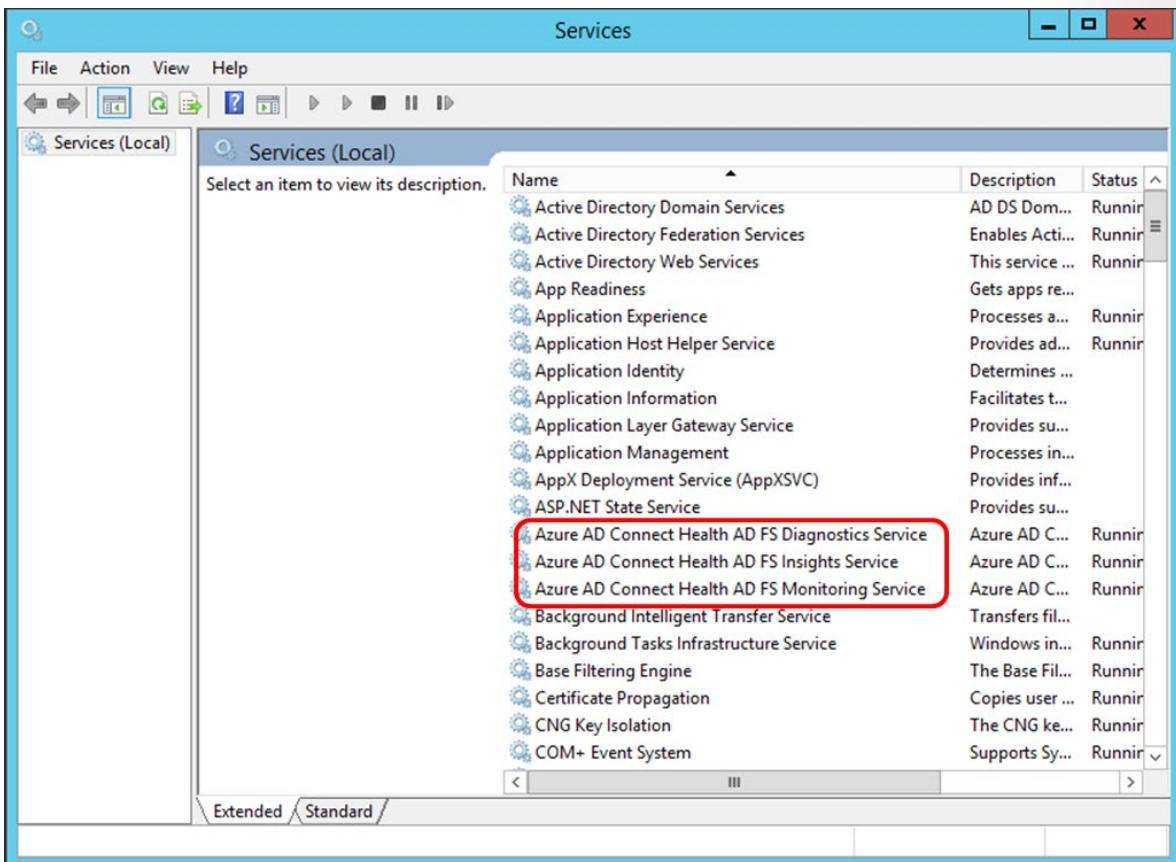


The screenshot shows a Windows PowerShell window titled "Administrator: Windows PowerShell". The command executed is "Register-AzureADConnectHealthADFSAgent". The output indicates that the command is executing and detecting AdFederationService roles. It also displays several warning messages about AD FS auditing being disabled and machine audit security policies. The process involves fetching and storing agent credentials and starting agent services. The log file "AdHealthAdfsAgentConfiguration.2015-06-22_06-04-20.log" is mentioned as containing more information regarding the warnings.

```
Administrator: Windows PowerShell
Executing Elevated PowerShell Command: Register-AzureADConnectHealthADFSAgent
Detecting AdFederationService roles...
WARNING: AD FS auditing is not enabled correctly, please verify AD FS configuration and Machine Audit security policy.
Detected the following role(s) for adfs.contoso.com:
WARNING: AD FS auditing is not enabled correctly, please verify AD FS configuration and Machine Audit security policy.
AD FS 2012 R2 Federation Server
WARNING: AD FS auditing is not enabled correctly, please verify AD FS configuration and Machine Audit security policy.
Fetching and storing agent credentials...
Starting agent services...
WARNING: Agent registration completed with warning(s).
WARNING: Log file AdHealthAdfsAgentConfiguration.2015-06-22_06-04-20.log contains more information regarding the warning(s).
PS C:\Users\Administrator\Downloads>
```

To verify that the agent was installed, look for the following services on the server. If you completed the configuration, they should already be running. Otherwise, they're stopped until the configuration is complete.

- Azure AD Connect Health AD FS Diagnostics Service
- Azure AD Connect Health AD FS Insights Service
- Azure AD Connect Health AD FS Monitoring Service



Install the agent for Sync

The Azure AD Connect Health agent for Sync is installed automatically in the latest version of Azure AD Connect. To use Azure AD Connect for Sync, download the latest version of Azure AD Connect and install it.

To verify the agent has been installed, look for the following services on the server. If you completed the configuration, the services should already be running. Otherwise, the services are stopped until the configuration is complete.

- Azure AD Connect Health Sync Insights Service
- Azure AD Connect Health Sync Monitoring Service

Azure AD Connect Health Sync Insights Service	Azure AD C...	Running	Automatic (D...)	[Stop]
Azure AD Connect Health Sync Monitoring Service	Azure AD C...	Running	Automatic (D...)	[Stop]

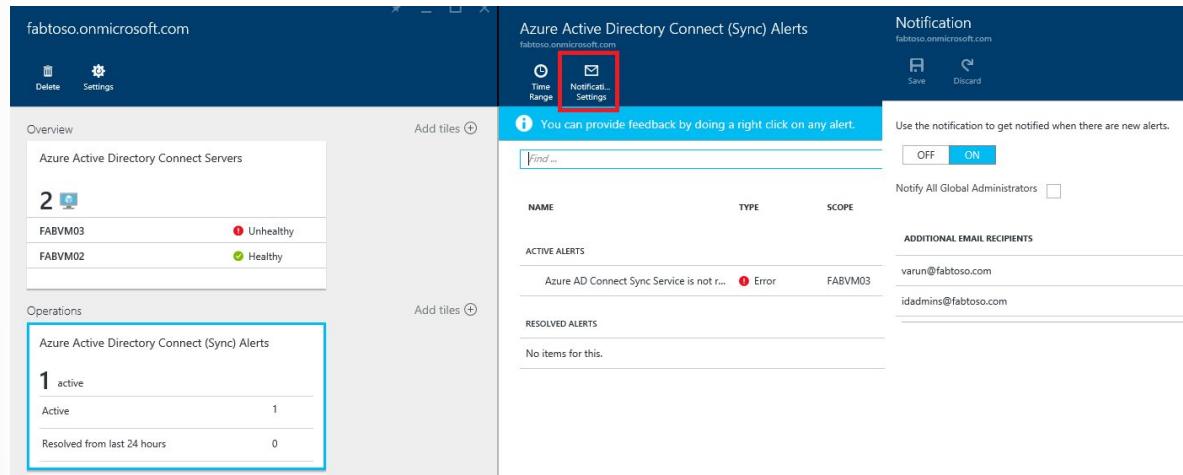
Note - Remember that you must have Azure AD Premium to use Azure AD Connect Health. If you don't have Azure AD Premium, you can't complete the configuration in the Azure portal.

Manage Azure AD Connect Health

This section describes various operations you can perform by using Azure Active Directory (Azure AD) Connect Health.

Enable email notifications

You can configure the Azure AD Connect Health service to send email notifications when alerts indicate that your identity infrastructure is not healthy. This occurs when an alert is generated, and when it is resolved.



Note - Email notifications are enabled by default.

To enable Azure AD Connect Health email notifications

1. Open the **Alerts** blade for the service for which you want to receive email notification.
2. From the action bar, click **Notification Settings**.
3. At the email notification switch, select **ON**.
4. Select the check box if you want all global administrators to receive email notifications.
5. If you want to receive email notifications at any other email addresses, specify them in the **Additional Email Recipients** box. To remove an email address from this list, right-click the entry and select **Delete**.
6. To finalize the changes, click **Save**. Changes take effect only after you save.

Note - When there are issues processing synchronization requests in our back-end service, this service sends a notification email with the details of the error to the administrative contact email address(es) of your tenant. We heard feedback from customers that in certain cases the volume of these messages is prohibitively large so we are changing the way we send these messages. Instead of sending a message for every sync error every time it occurs we will send out a daily digest of all errors the back-end service has returned. This enables customers to process these errors in a more efficient manner and reduces the number of duplicate error messages.

Delete a server or service instance

Note - Azure AD premium license is required for the deletion steps.

In some instances, you might want to remove a server from being monitored. Here's what you need to know to remove a server from the Azure AD Connect Health service.

When you're deleting a server, be aware of the following:

- This action stops collecting any further data from that server. This server is removed from the monitoring service. After this action, you are not able to view new alerts, monitoring, or usage analytics data for this server.
- This action does not uninstall the Health Agent from your server. If you have not uninstalled the Health Agent before performing this step, you might see errors related to the Health Agent on the server.
- This action does not delete the data already collected from this server. That data is deleted in accordance with the Azure data retention policy.
- After performing this action, if you want to start monitoring the same server again, you must uninstall and reinstall the Health Agent on this server.

Delete a server from the Azure AD Connect Health service

Note - Azure AD premium license is required for the deletion steps.

Azure AD Connect Health for Active Directory Federation Services (AD FS) and Azure AD Connect (Sync):

1. Open the **Server** blade from the **Server List** blade by selecting the server name to be removed.
2. On the **Server** blade, from the action bar, click **Delete**.

NAME	ACTIVE ALERTS	LAST BOOT TIME	LAST UPLOADED	STATUS
FABVM02	0	1/9/2019, 21:11:56	1/22/2019, 12:27:17	✓
FABVM01	0	1/9/2019, 18:43:41	1/22/2019, 12:24:33	✓

3. Confirm by typing the server name in the confirmation box.

4. Click **Delete**.

Azure AD Connect Health for Azure Active Directory Domain Services:

5. Open the **Domain Controllers** dashboard.
6. Select the domain controller to be removed.
7. From the action bar, click **Delete Selected**.
8. Confirm the action to delete the server.
9. Click **Delete**.

Delete a service instance from Azure AD Connect Health service

In some instances, you might want to remove a service instance. Here's what you need to know to remove a service instance from the Azure AD Connect Health service.

When you're deleting a service instance, be aware of the following:

- This action removes the current service instance from the monitoring service.
- This action does not uninstall or remove the Health Agent from any of the servers that were monitored as part of this service instance. If you have not uninstalled the Health Agent before performing this step, you might see errors related to the Health Agent on the servers.
- All data from this service instance is deleted in accordance with the Azure data retention policy.
- After performing this action, if you want to start monitoring the service, uninstall and reinstall the Health Agent on all the servers. After performing this action, if you want to start monitoring the same server again, uninstall, reinstall, and register the Health Agent on that server.

To delete a service instance from the Azure AD Connect Health service

1. Open the **Service** blade from the **Service List** blade by selecting the service identifier (farm name) that you want to remove.
2. On the **Service** blade, from the action bar, click **Delete**.



3. Confirm by typing the service name in the confirmation box (for example: sts.contoso.com).
4. Click **Delete**.

Manage access with Azure Role Based Access Control

Azure role-based access control (Azure RBAC)³⁴ for Azure AD Connect Health provides access to users and groups other than global administrators. Azure RBAC assigns roles to the intended users and groups, and provides a mechanism to limit the global administrators within your directory.

Roles

Azure AD Connect Health supports the following built-in roles:

³⁴ <https://docs.microsoft.com/azure/role-based-access-control/role-assignments-portal>

Role	Permissions
Owner	Owners can <i>manage access</i> (for example, assign a role to a user or group), <i>view all information</i> (for example, view alerts) from the portal, and <i>change settings</i> (for example, email notifications) within Azure AD Connect Health. By default, Azure AD global administrators are assigned this role, and this cannot be changed.
Contributor	Contributors can <i>view all information</i> (for example, view alerts) from the portal, and <i>change settings</i> (for example, email notifications) within Azure AD Connect Health.
Reader	Readers can <i>view all information</i> (for example, view alerts) from the portal within Azure AD Connect Health.

All other roles (such as User Access Administrators or DevTest Labs Users) have no impact to access within Azure AD Connect Health, even if the roles are available in the portal experience.

Access scope

Azure AD Connect Health supports managing access at two levels:

- **All service instances:** This is the recommended path in most cases. It controls access for all service instances (for example, an AD FS farm) across all role types that are being monitored by Azure AD Connect Health.
- **Service instance:** In some cases, you might need to segregate access based on role types or by a service instance. In this case, you can manage access at the service instance level.

Permission is granted if an end user has access either at the directory or service instance level.

Allow users or groups access to Azure AD Connect Health

The following steps show how to allow access.

Step 1: Select the appropriate access scope

To allow a user access at the *all service instances* level within Azure AD Connect Health, open the main blade in Azure AD Connect Health.

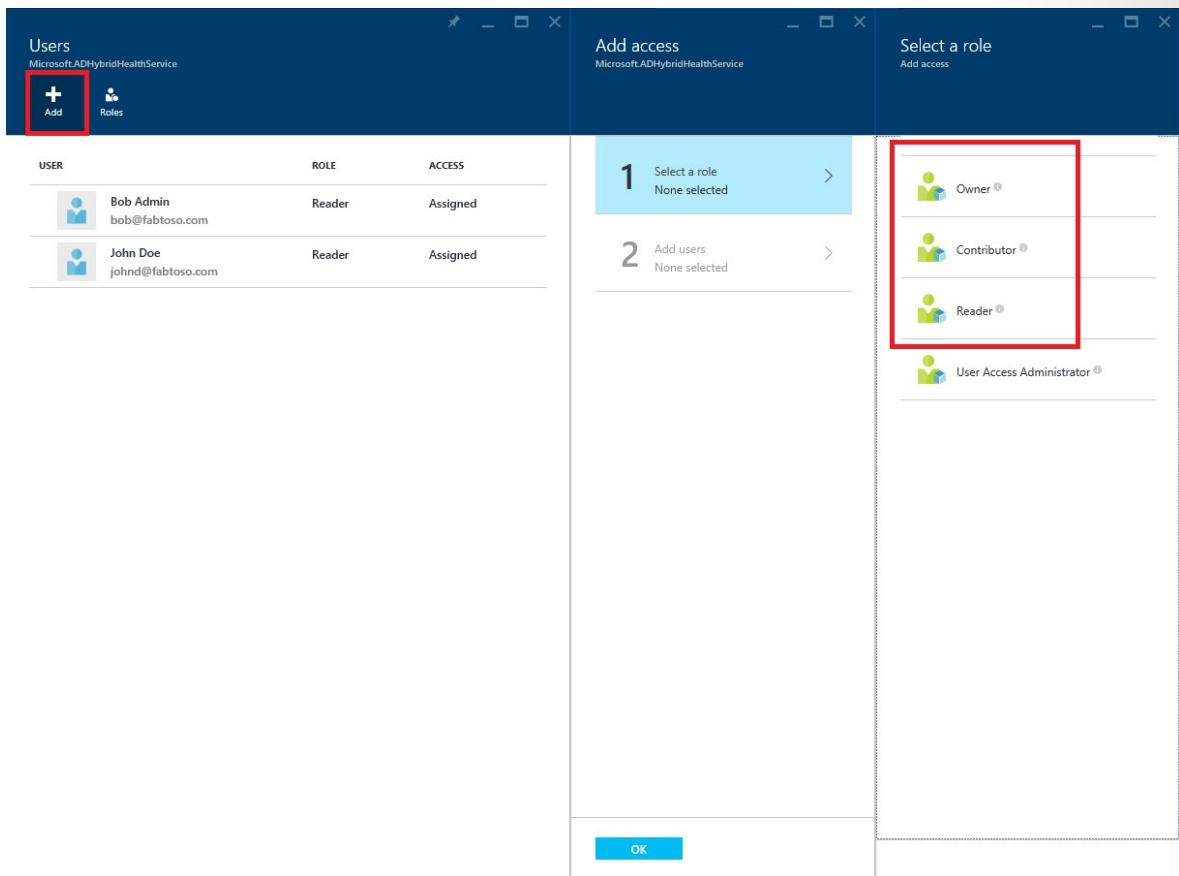
Step 2: Add users and groups, and assign roles

1. From the **Configure** section, click **Users**.

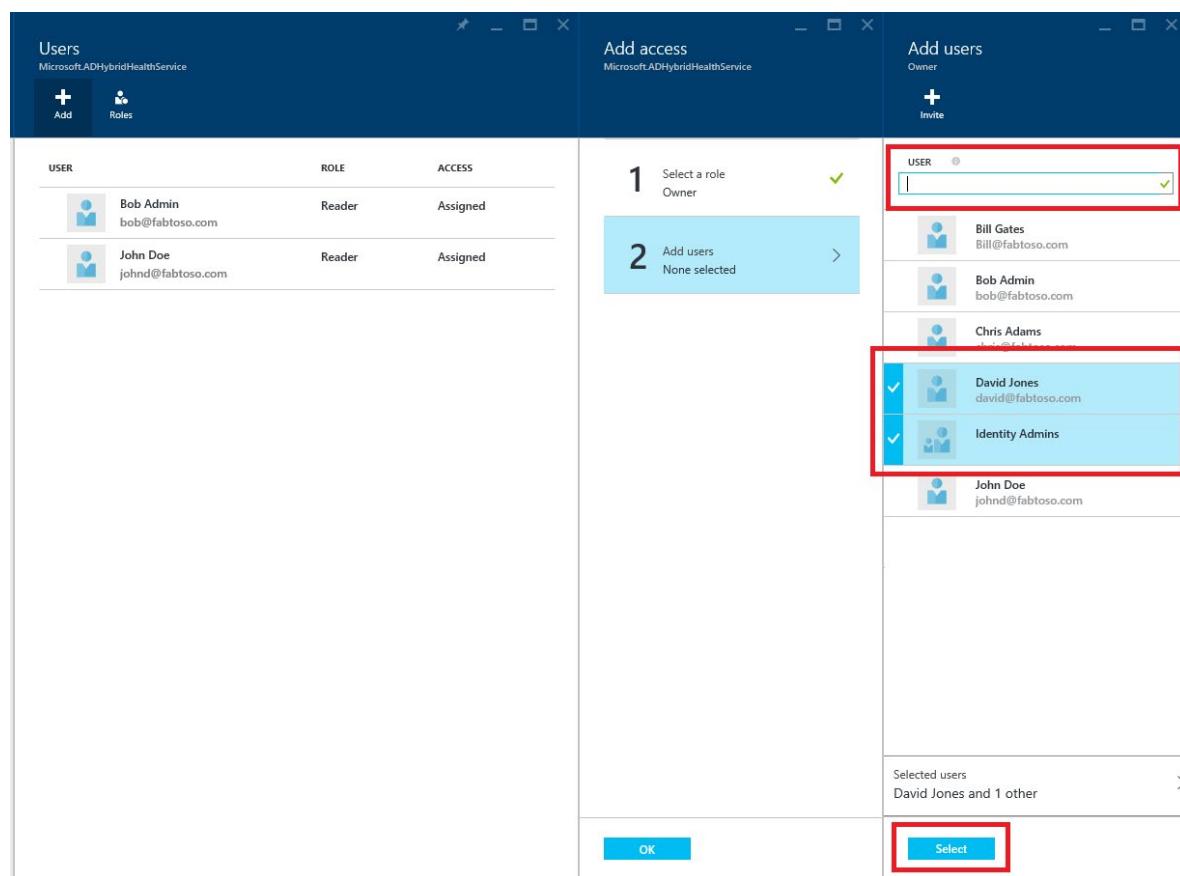
The screenshot shows the left sidebar of the Azure portal. It includes sections for 'AZURE ACTIVE DIRECTORY CONNECT (SYNC)', 'ACTIVE DIRECTORY FEDERATION SERVICES', 'ACTIVE DIRECTORY DOMAIN SERVICES', 'CONFIGURE', and 'TROUBLESHOOTING + SUPPORT'. A red box highlights the 'Role based access control (IAM)' link under the 'CONFIGURE' section.

- Quick start
- AZURE ACTIVE DIRECTORY CONNECT (SYNC)
 - Sync errors
 - Sync services
- ACTIVE DIRECTORY FEDERATION SERVICES
 - AD FS services
- ACTIVE DIRECTORY DOMAIN SERVICES
 - AD DS services
- CONFIGURE
 - Settings
 - Role based access control (IAM)
- TROUBLESHOOTING + SUPPORT
 - Troubleshoot

2. Select **Add**.
3. In the **Select a role** pane, select a role (for example, **Owner**).



4. Type the name or identifier of the targeted user or group. You can select one or more users or groups at the same time. Click **Select**.



5. Select **OK**.
6. After the role assignment is complete, the users and groups appear in the list.

Now the listed users and groups have access, according to their assigned roles.

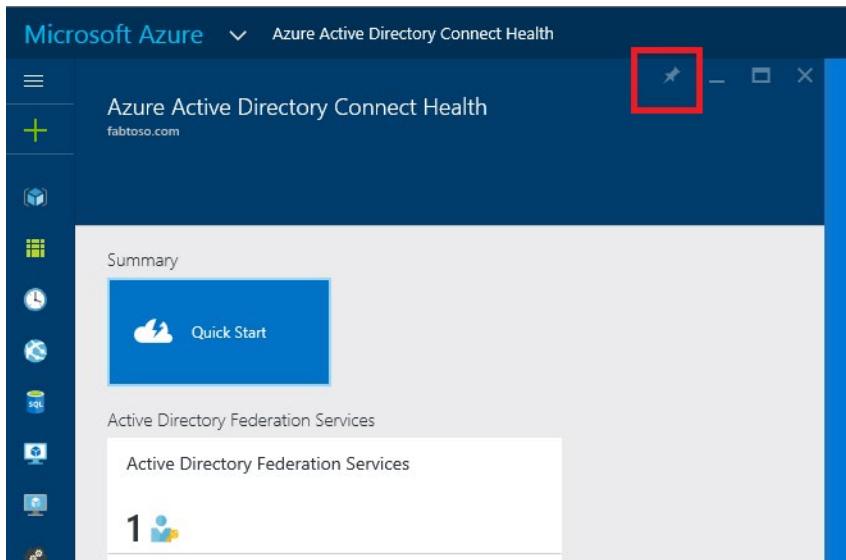
Note

- Global administrators always have full access to all the operations, but global administrator accounts are not present in the preceding list.
- The Invite Users feature is not supported within Azure AD Connect Health.

Step 3: Share the blade location with users or groups

7. After you assign permissions, a user can access Azure AD Connect Health by going [here³⁵](#).
8. On the blade, the user can pin the blade, or different parts of it, to the dashboard. Simply click the **Pin to dashboard** icon.

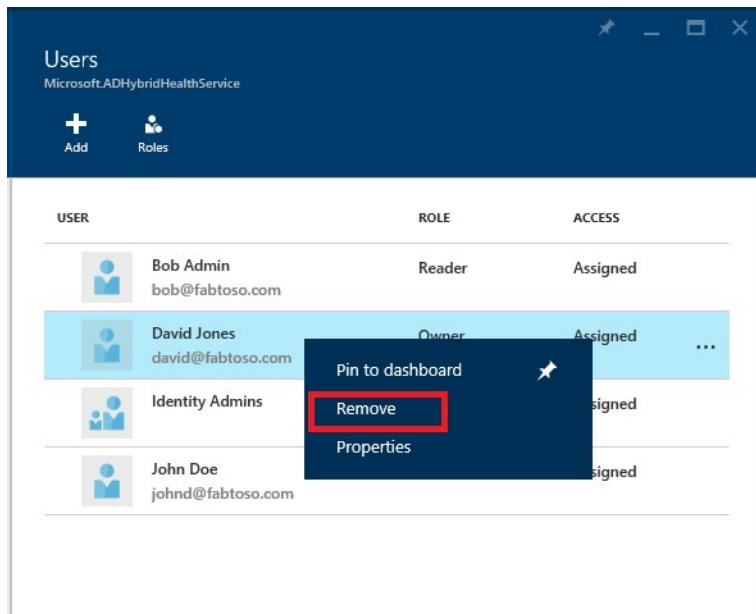
³⁵ <https://aka.ms/aadconnecthealth>



Note - A user with the Reader role assigned is not able to get Azure AD Connect Health extension from the Azure Marketplace. The user cannot perform the necessary “create” operation to do so. The user can still get to the blade by going to the preceding link. For subsequent usage, the user can pin the blade to the dashboard.

Remove users or groups

You can remove a user or a group added to Azure AD Connect Health and Azure RBAC. Simply right-click the user or group, and select **Remove**.



Demo: Azure AD Connect Health monitors on-premises AD Domain Services

	Watch this video to learn more about using Azure AD Connect Health for monitoring.
-----------------------------------------------------------------------------------	------------------------------------------------------------------------------------



<https://channel9.msdn.com/Series/Azure-Active-Directory-Videos-Demos/Azure-AD-Connect-Health-monitors-on-premises-AD-Domain-Services/player?format=ny>

Diagnose and remediate duplicated attribute sync errors

Overview

Taking one step farther to highlight sync errors, Azure Active Directory (Azure AD) Connect Health introduces self-service remediation. It troubleshoots duplicated attribute sync errors and fixes objects that are orphaned from Azure AD. The diagnosis feature has these benefits:

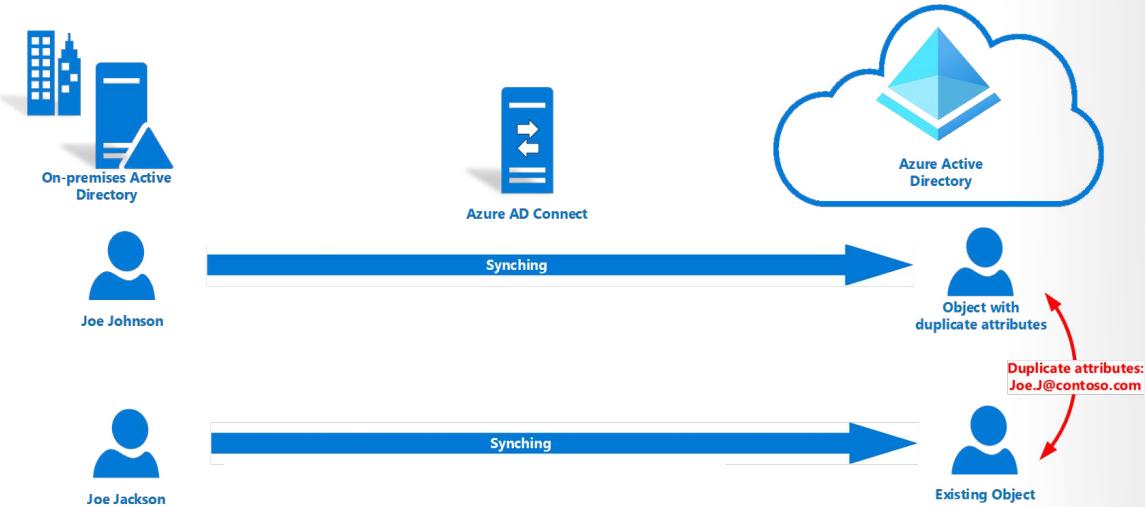
- It provides a diagnostic procedure that narrows down duplicated attribute sync errors. And it gives specific fixes.
- It applies a fix for dedicated scenarios from Azure AD to resolve the error in a single step.
- No upgrade or configuration is required to enable this feature.

Problems

A common scenario

When **QuarantinedAttributeValueMustBeUnique** and **AttributeValueMustBeUnique** sync errors happen, it's common to see a **UserPrincipalName** or **Proxy Addresses** conflict in Azure AD. You might

solve the sync errors by updating the conflicting source object from the on-premises side. The sync error will be resolved after the next sync. For example, this image indicates that two users have a conflict of their **UserPrincipalName**. Both are **Joe.J@contoso.com**. The conflicting objects are quarantined in Azure AD.



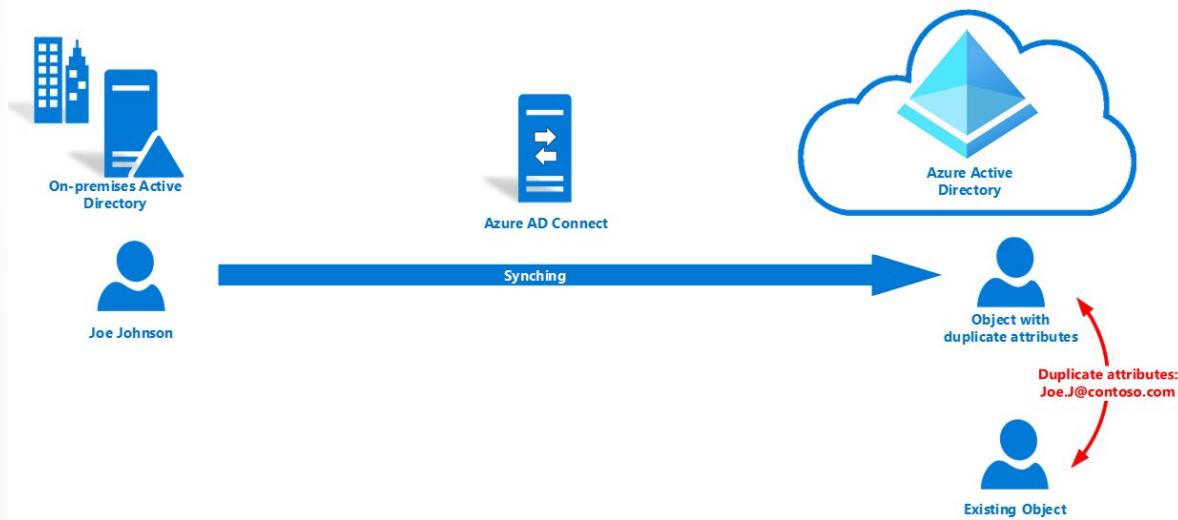
Orphaned object scenario

Occasionally, you might find that an existing user loses the **Source Anchor**. The deletion of the source object happened in on-premises Active Directory. But the change of deletion signal never got synchronized to Azure AD. This loss happens for reasons like sync engine issues or domain migration. When the same object gets restored or recreated, logically, an existing user should be the user to sync from the **Source Anchor**.

When an existing user is a cloud-only object, you can also see the conflicting user synchronized to Azure AD. The user can't be matched in sync to the existing object. There's no direct way to remap the **Source Anchor**.

As an example, the existing object in Azure AD preserves the license of Joe. A newly synchronized object with a different **Source Anchor** occurs in a duplicated attribute state in Azure AD. Changes for Joe in on-premises Active Directory won't be applied to Joe's original user (existing object) in Azure AD.

Orphaned User Object Case



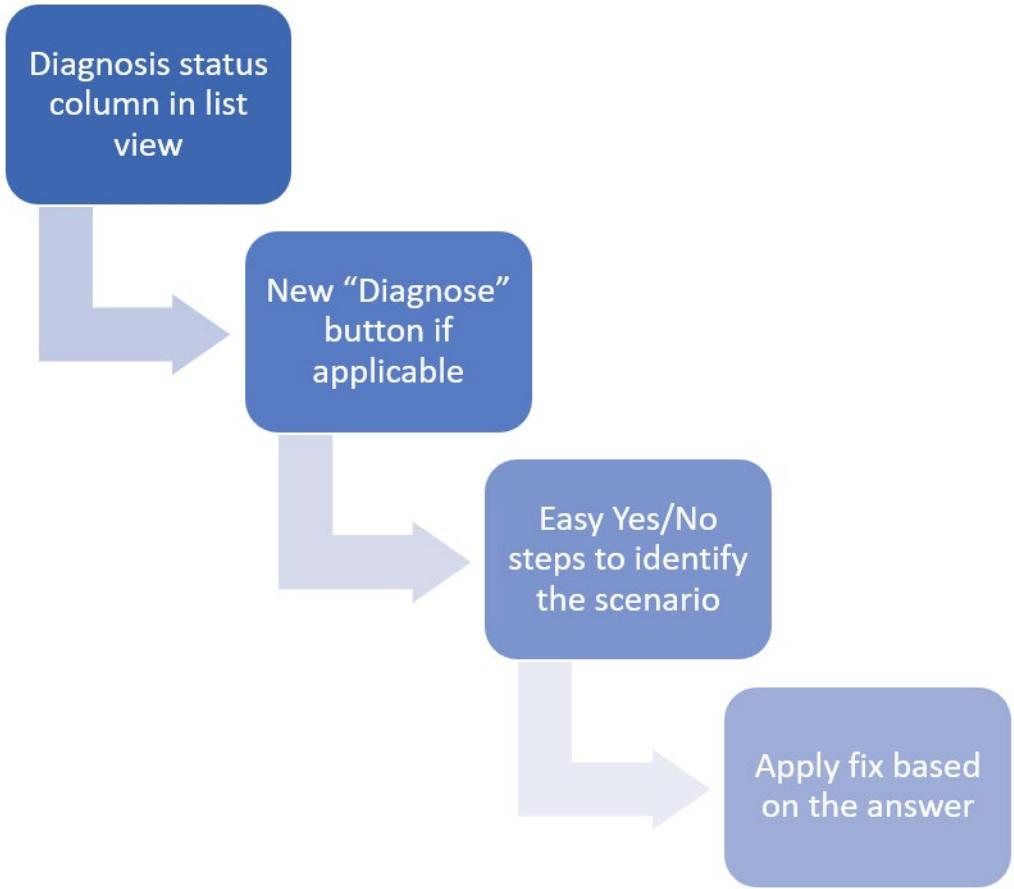
Diagnostic and troubleshooting steps in Connect Health

The diagnose feature supports user objects with the following duplicated attributes:

Attribute name	Synchronization error types
UserPrincipalName	QuarantinedAttributeValueMustBeUnique or AttributeValueMustBeUnique
ProxyAddresses	QuarantinedAttributeValueMustBeUnique or AttributeValueMustBeUnique
SipProxyAddress	AttributeValueMustBeUnique
OnPremiseSecurityIdentifier	AttributeValueMustBeUnique

Important - To access this feature, Global Admin permission, or Contributor permission from Azure RBAC, is required.

Follow the steps from the Azure portal to narrow down the sync error details and provide more specific solutions:



From the Azure portal, take a few steps to identify specific fixable scenarios:

1. Check the **Diagnose status** column. The status shows if there's a possible way to fix a sync error directly from Azure Active Directory. In other words, a troubleshooting flow exists that can narrow down the error case and potentially fix it.

Status	What does it mean?
Not Started	You haven't visited this diagnosis process. Depending on the diagnostic result, there's a potential way to fix the sync error directly from the portal.
Manual Fix Required	The error doesn't fit the criteria of available fixes from the portal. Either conflicting object types aren't users, or you already went through the diagnostic steps, and no fix resolution was available from the portal. In the latter case, a fix from the on-premises side is still one of the solutions.
Pending Sync	A fix was applied. The portal is waiting for the next sync cycle to clear the error.

Important - The diagnostic status column will reset after each sync cycle.

2. Select the **Diagnose** button under the error details. You'll answer a few questions and identify the sync error details. Answers to the questions help identify an orphaned object case.

3. If a **Close** button appears at the end of the diagnostics, there's no quick fix available from the portal based on your answers. Refer to the solution shown in the last step. Fixes from on-premises are still the solutions. Select the **Close** button. The status of the current sync error switches to **Manual fix required**. The status stays during the current sync cycle.
4. After an orphaned object case is identified, you can fix the duplicated attributes sync errors directly from the portal. To trigger the process, select the **Apply Fix** button. The status of the current sync error updates to **Pending sync**.
5. After the next sync cycle, the error should be removed from the list.

Module 1 Review Questions

Module 1 Review Questions

Review Question 1

What is the defining feature of hybrid identity solutions?

- They create common user identities for authenticating and authorizing users who operate workstations that run on a variety of operating systems.
- They create common user identities that are trusted for authentication and authorization between organizations.
- They create common user identities for authentication and authorization to both on-premises and cloud-based resources.

Review Question 2

Which authentication method requires the least effort regarding deployment, maintenance, and infrastructure?

- Password hash synchronization (PHS).
- Pass-through authentication (PTA).
- Federated authentication.

Review Question 3

Some situations might require the removal of a server from being monitored by the Azure AD Connect Health service. What needs to be done to start monitoring the same server again?

- The Azure AD Connect Health service needs to be stopped and restarted on any other targeted server in the network.
- The Health Agent needs to be uninstalled and reinstalled on this server.
- The data already collected from this server needs to be deleted and then the Health Agent needs to be reactivated on the server.

Review Question 4

Users who are assigned limited administrator directory roles can use the Azure portal to invite B2B collaboration users. In addition to being invited to a directory or to a group, what else can B2B collaboration users be invited to?

- Limited self-service functionality for modifying their profiles.
- Network resources such as printers.
- An application.

Review Question 5

Azure AD B2B can be configured to federate with identity providers that use either of two protocols. One protocol is Security Assertion Markup Language (SAML); what is the other protocol?

- WS-Federation (WS-Fed)
- Layer Two Tunneling Protocol (L2TP)
- Resource Location Protocol (RLP)

Review Question 6

What are dynamic groups?

- Dynamic groups are Microsoft 365 groups whose memberships consist of Dynamics 365 users, who require special attribute configurations.
- Dynamic groups are security groups whose memberships are based on user attributes (such as userType, department, or country/region).
- Dynamic groups are groups whose membership numbers fluctuate significantly within a given timeframe.

Review Question 7

Typically, Azure AD defines users in three ways. Cloud identities and guest users are two of the ways. What is the third way Azure AD defines users?

- As non-connected users.
- As transitional users.
- As directory-synchronized identities.

Review Question 8

Azure AD group-based licensing makes large scale management easier. Typically, how soon are license modifications effective after group membership changes are made?

- Within the timeframe of local domain controllers being refreshed.
- Within minutes of a membership change.
- Within 24 hours of a membership change.

Review Question 9

Azure AD allows for the definition of two different types of groups; one type is Security groups, which are used to manage member and computer access to shared resources. What is the other type of group?

- Distribution groups, which are used for communications purposes via applications such as Teams and Exchange.
- Licensing groups, which are used to make it easier to administer software licenses.
- Microsoft 365 groups, which provide access to shared mailboxes, calendars, SharePoint sites, and so on.

Review Question 10

A domain name is included as part of a user name or email address for users and groups. Can a domain name also be included as part of an application or other resource?

- Yes, a domain name can be included as part of an application or other resource if the domain name is owned by the organization that contains the resource.
- A domain name can be included as part of the app ID URI for an application, but cannot be included as part of other resources.
- No, a domain name cannot be included as part of an application or other resource.

Review Question 11

The proliferation of many types of devices and bring your own device (BYOD) concept require IT professionals to accommodate two rather different goals. One goal is to allow users to be productive wherever and anytime. What is the other goal?

- Provide antimalware apps for a various devices.
- Establish baseline security guidelines for users.
- Protect the organization's assets.

Review Question 12

Azure AD guest users have restricted directory permissions. Which of the following answers best describes guest users capabilities?

- They can manage their own profile, change their own password, and add other B2B guests to groups.
- They can manage their own profile, change their own password, and retrieve some information about other users, groups, and apps.
- They can manage their own profile, change their own password, and identify group members or other directory objects.

Module 1 Hands-on Exercises

Lab 1: Manage User Roles

To download the most recent version of this lab, please visit the SC-300 [GitHub repository³⁶](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

Your company recently hired a new employee who will perform duties as an application administrator. You must create a new user and assign the appropriate role.

Objectives

After you complete this lab, you will be able to:

- Add a new users.
- Assign and remove roles from a user.

Lab setup

- Estimated time: 10 minutes

Lab 2: Working with Tenant Properties

To download the most recent version of this lab, please visit the SC-300 [GitHub repository³⁷](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

You need to identify and update the different properties associated with your tenant.

Objectives

After you complete this lab, you will be able to:

- Changing the tenant display name.
- Finding the country or region associated with you tenant.

³⁶ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

³⁷ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

- Finding the location associated with your tenant.
- Finding the tenant-ID.
- Changing the Technical Contact and adding Privacy information to your Azure AD.

Lab setup

- Estimated time: 10 minutes

Lab 3: Assigning licenses using group membership

To download the most recent version of this lab, please visit the SC-300 [GitHub repository³⁸](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

Your organization has decided to use security groups in Azure AD to manage licenses. You need to configure a new security group and assign a license to that group and verify group member license's have been updated.

Objectives

After you complete this lab, you will be able to:

- Create a new user in Azure Active Directory
- Create a security group in Azure Active Directory
- Assign a license to a group

Lab setup

- Estimated time: 10 minutes

Lab 4: Restore a deleted user

To download the most recent version of this lab, please visit the SC-300 [GitHub repository³⁹](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

³⁸ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

³⁹ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

Lab scenario

It may happen that an account is deleted and then needs to be recovered. You need to verify you can recover an account that has been deleted recently.

Objectives

After you complete this lab, you will be able to:

- Remove a user from Azure Active Directory
- Restore a deleted user

Lab setup

- Estimated time: 5 minutes

Lab 5: Adding groups to Azure AD

To download the most recent version of this lab, please visit the SC-300 [GitHub repository⁴⁰](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

Part of your duties as an Azure AD administrator is to create different types of groups. You need to create a new Microsoft 365 group for your organization's sales department.

Objectives

After you complete this lab, you will be able to:

- Create an Microsoft 365 group in Azure Active Directory

Lab setup

- Estimated time: 5 minutes

Lab 6: Change group license assignments

To download the most recent version of this lab, please visit the SC-300 [GitHub repository⁴¹](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

⁴⁰ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

⁴¹ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

Lab scenario

Occasionally, you may need to change the license assignment that are used by an Azure AD security group. You must ensure you are familiar with the procedure for changing a group's license assignment.

Objectives

After you complete this lab, you will be able to:

- Change group license assignments

Lab setup

- Estimated time: 5 minutes

Lab 7: Change user account license assignments

To download the most recent version of this lab, please visit the SC-300 [GitHub repository⁴²](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

Some user accounts in your organization will not be provided all available products in their assigned license or will need updates or additions to their license assignment. You need to ensure you are able to update a user account's license assignment in Azure AD.

Objectives

After you complete this lab, you will be able to:

- Create a new user in Azure Active Directory
- Update user license assignments

Lab setup

- Estimated time: 5 minutes

Lab 8: Configure external collaboration settings

To download the most recent version of this lab, please visit the SC-300 [GitHub repository⁴³](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version

⁴² <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

⁴³ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

You must enable external collaboration settings for your organization for approved guests access.

Objectives

After you complete this lab, you will be able to:

- Configure external collaboration settings

Lab setup

- Estimated time: 5 minutes

Lab 9: Add guest users to the directory

To download the most recent version of this lab, please visit the SC-300 [GitHub repository⁴⁴](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

Your company works with many vendors and, on occasion, you need to add some vendor accounts to your directory as a guest.

Objectives

After you complete this lab, you will be able to:

- Add guest users to the directory

Lab setup

- Estimated time: 5 minutes

Lab 10: Invite guest users in bulk

To download the most recent version of this lab, please visit the SC-300 [GitHub repository⁴⁵](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version

⁴⁴ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

⁴⁵ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

A recent partnership has been established with another company. For now, employees of the partner company will be added as guests. You need to ensure you can import multiple guest users at one time.

Objectives

After you complete this lab, you will be able to:

- Invite guest users in bulk

Lab setup

- Estimated time: 10 minutes

Lab 11: Working with dynamic groups

To download the most recent version of this lab, please visit the SC-300 [GitHub repository⁴⁶](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

As your company grows, manually group management is too time consuming. Since standardizing the directory, you can now take advantage of dynamic groups. You must create a new dynamic group to ensure you're ready for dynamic group creation in production.

Objectives

After you complete this lab, you will be able to:

- Creating a dynamic group with all users as members

Lab setup

- Estimated time: 10 minutes

⁴⁶ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

Module 1 Summary

Summary for Module 1

During this module you saw how companies can implement their initial Azure AD system; then we explored topics around creating and managing both internal and external identities. Finally, we looked at how hybrid identity enables you to connect your on-premises and cloud identity solutions and resources.

Implement the Initial Configuration of Azure AD

During this lesson you explored how to:

- Configure and manage Azure Active Directory roles.
- Configure and manage custom domains.
- Configure and manage device registration options.
- Configure delegation by using administrative units.
- Configure tenant-wide settings

Create, Configure and Manage Identities

During this lesson you explored how to:

- Create, configure, and manage users
- Create, configure, and manage groups
- Manage licenses

Implement and Manage External Identities

During this lesson you explored how to:

- Manage external collaboration settings in Azure AD
- Invite external users (individually or in bulk)
- Manage external user accounts in Azure AD
- Configure identity providers (social and SAML/WS-fed)

Implement and Manage Hybrid Identity

During this lesson you explored how to:

- Plan, design, and implement Azure Active Directory Connect (AADC), including password hash synchronization (PHS), pass-through authentication (PTA), seamless single sign-on (Seamless SSO), and federation
- Manage Azure Active Directory Connect (AADC)
- Manage password hash synchronization (PHS)
- Manage pass-through authentication (PTA)
- Manage seamless single sign-on (Seamless SSO)

- Manage federation excluding manual ADFS deployments
- Troubleshoot synchronization errors
- Implement and manage Azure Active Directory Connect Health

Supplemental Resources

Use these resources to discover more:

- Information about which roles manage Azure resources and which roles manage Azure AD resources is available at **Classic subscription administrator roles, Azure roles, and Azure AD roles**⁴⁷.
- For more information about roles, see **Understand Azure role definitions**⁴⁸.
- For information about how to use PIM, see **Privileged Identity Management**⁴⁹.
- The following step-by-step guides provide information on how you can use Conditional Access to configure equivalent policies to those policies enabled by security defaults:
 - **Require MFA for administrators**⁵⁰
 - **Require MFA for Azure management**⁵¹
 - **Block legacy authentication**⁵²
 - **Require MFA for all users**⁵³
 - **Require Azure AD MFA registration**⁵⁴ - Requires Azure AD Identity Protection, a part of Azure AD Premium P2.

⁴⁷ <https://docs.microsoft.com/azure/role-based-access-control/rbac-and-directory-admin-roles>

⁴⁸ <https://docs.microsoft.com/azure/role-based-access-control/role-definitions>

⁴⁹ <https://docs.microsoft.com/azure/active-directory/privileged-identity-management/>

⁵⁰ <https://docs.microsoft.com/azure/active-directory/conditional-access/howto-conditional-access-policy-admin-mfa>

⁵¹ <https://docs.microsoft.com/azure/active-directory/conditional-access/howto-conditional-access-policy-azure-management>

⁵² <https://docs.microsoft.com/azure/active-directory/conditional-access/howto-conditional-access-policy-block-legacy>

⁵³ <https://docs.microsoft.com/azure/active-directory/conditional-access/howto-conditional-access-policy-all-users-mfa>

⁵⁴ <https://docs.microsoft.com/azure/active-directory/identity-protection/howto-identity-protection-configure-mfa-policy>

Answers

Review Question 1

What is the defining feature of hybrid identity solutions?

- They create common user identities for authenticating and authorizing users who operate workstations that run on a variety of operating systems.
- They create common user identities that are trusted for authentication and authorization between organizations.
- They create common user identities for authentication and authorization to both on-premises and cloud-based resources.

Explanation

Authentication and authorization are essential for hybrid identity solutions.

Review Question 2

Which authentication method requires the least effort regarding deployment, maintenance, and infrastructure?

- Password hash synchronization (PHS).
- Pass-through authentication (PTA).
- Federated authentication.

Explanation

PHS requires the least effort regarding deployment, maintenance, and infrastructure, which typically applies to organizations that only need their users to sign in to Microsoft 365, SaaS apps, and other Azure AD-based resources.

Review Question 3

Some situations might require the removal of a server from being monitored by the Azure AD Connect Health service. What needs to be done to start monitoring the same server again?

- The Azure AD Connect Health service needs to be stopped and restarted on any other targeted server in the network.
- The Health Agent needs to be uninstalled and reinstalled on this server.
- The data already collected from this server needs to be deleted and then the Health Agent needs to be reactivated on the server.

Explanation

To start monitoring a server again, the Health Agent needs to be uninstalled and reinstalled.

Review Question 4

Users who are assigned limited administrator directory roles can use the Azure portal to invite B2B collaboration users. In addition to being invited to a directory or to a group, what else can B2B collaboration users be invited to?

- Limited self-service functionality for modifying their profiles.
- Network resources such as printers.
- An application.

Explanation

B2B collaboration users can also be invited to an application.

Review Question 5

Azure AD B2B can be configured to federate with identity providers that use either of two protocols. One protocol is Security Assertion Markup Language (SAML); what is the other protocol?

- WS-Federation (WS-Fed)
- Layer Two Tunneling Protocol (L2TP)
- Resource Location Protocol (RLP)

Explanation

WS-Fed is one of two protocols that Azure AD B2B can make use of to federate with identity providers.

Review Question 6

What are dynamic groups?

- Dynamic groups are Microsoft 365 groups whose memberships consist of Dynamics 365 users, who require special attribute configurations.
- Dynamic groups are security groups whose memberships are based on user attributes (such as userType, department, or country/region).
- Dynamic groups are groups whose membership numbers fluctuate significantly within a given timeframe.

Explanation

Dynamic groups are security groups whose memberships are based on user attributes

Review Question 7

Typically, Azure AD defines users in three ways. Cloud identities and guest users are two of the ways.

What is the third way Azure AD defines users?

- As non-connected users.
- As transitional users.
- As directory-synchronized identities.

Explanation

Azure AD defines users as cloud identities, guest users, and as directory-synchronized identities

Review Question 8

Azure AD group-based licensing makes large scale management easier. Typically, how soon are license modifications effective after group membership changes are made?

- Within the timeframe of local domain controllers being refreshed.
- Within minutes of a membership change.
- Within 24 hours of a membership change.

Explanation

License modifications that result from group membership changes are typically effective within minutes of a membership change.

Review Question 9

Azure AD allows for the definition of two different types of groups; one type is Security groups, which are used to manage member and computer access to shared resources. What is the other type of group?

- Distribution groups, which are used for communications purposes via applications such as Teams and Exchange.
- Licensing groups, which are used to make it easier to administer software licenses.
- Microsoft 365 groups, which provide access to shared mailboxes, calendars, SharePoint sites, and so on.

Explanation

Azure AD allows for the definition of Security groups and Microsoft 365 groups.

Review Question 10

A domain name is included as part of a user name or email address for users and groups. Can a domain name also be included as part of an application or other resource?

- Yes, a domain name can be included as part of an application or other resource if the domain name is owned by the organization that contains the resource.
- A domain name can be included as part of the app ID URI for an application, but cannot be included as part of other resources.
- No, a domain name cannot be included as part of an application or other resource.

Explanation

When an organization that contains an application or other resources, the domain can be included if it is owned by the same organization.

Review Question 11

The proliferation of many types of devices and bring your own device (BYOD) concept require IT professionals to accommodate two rather different goals. One goal is to allow users to be productive wherever and anytime. What is the other goal?

- Provide antimalware apps for a various devices.
- Establish baseline security guidelines for users.
- Protect the organization's assets.

Explanation

Identity is new perimeter is a common security phrase these days, meaning that validation of both people and devices are required to protect company assets.

Review Question 12

Azure AD guest users have restricted directory permissions. Which of the following answers best describes guest users capabilities?

- They can manage their own profile, change their own password, and add other B2B guests to groups.
- They can manage their own profile, change their own password, and retrieve some information about other users, groups, and apps.
- They can manage their own profile, change their own password, and identify group members or other directory objects.

Explanation

Guest users can only manage aspects of their own profile information, like their password; and view available resource like apps.

Module 2 Implement Authentication and Access Solution

Learning Objectives

Learning Objectives

After completing this module, you'll be able to:

- Plan and implement Azure Multifactor Authentication (MFA)
 - plan Azure MFA deployment (excluding MFA Server)
 - implement and manage Azure MFA settings
 - manage MFA settings for users
- Manage user authentication
 - administer authentication methods (FIDO2 and Passwordless)
 - implement an authentication solution based on Windows Hello for Business
 - configure and deploy self-service password reset
 - deploy and manage password protection
 - implement and manage tenant restrictions
- Plan, implement, and administer conditional access
 - plan and implement security defaults
 - plan conditional access policies
 - implement conditional access policy controls and assignments (targeting, applications, and conditions)
 - testing and troubleshooting conditional access policies

- implement application controls
- implement session management
- configure smart lockout thresholds
- Manage Azure AD Identity Protection
 - implement and manage a user risk policy
 - implement and manage sign-in risk policies
 - implement and manage MFA registration policy
 - monitor, investigate and remediate elevated risky users

Plan and Implement Azure Multifactor Authentication (MFA)

Introduction

Imagine you are security engineer for a large manufacturing firm. Your company works on several big contracts for popular personal electronics companies including Microsoft. Clients send you their confidential designs which are then stored in your Azure infrastructure. Plenty of hackers would love to get their hands on the next generation designs and it's your job to protect them.

You've done a lot of work hardening your network and ensuring that only the right people have access to client data, but there's still a big hole to protect - user accounts. This module will look at one of the best ways to stop unauthorized users from gaining access through a username and password. That technology is multi-factor authentication.

Learning objectives

In this module, you will:

- Learn about Azure AD Multi-Factor Authentication (MFA)
- Create a plan to deploy Azure AD MFA
- Turn on Azure AD MFA for users and specific apps

Prerequisites

- Basic knowledge of the Azure portal
- Basic knowledge of Azure Active Directory

Azure Multi-Factor Authentication

Protecting your cloud assets is one of the primary goals for security group. One of the primary ways unauthorized users get access to systems is by obtaining a valid username and password combination. Azure can help mitigate this with several features of Azure Active Directory including:

- **Password complexity rules** - This will force users to generate hard(er)-to-guess passwords.
- **Password expiration rules** - You can force users to change their passwords on a periodic basis (and avoid using previous-used passwords).
- **Self-service password reset (SSPR)** - This allows users to self-serve and reset their password if they have forgotten it without involving an IT department.
- **Azure AD Identity Protection** - To help protect your organization's identities, you can configure risk-based policies that automatically respond to risky behaviors. These policies can either automatically block the behaviors or initiate remediation, including requiring password changes.
- **Azure AD password protection** - You can block commonly used and compromised passwords via a globally banned-password list.
- **Azure AD smart lockout** - Smart lockout helps lock out malicious hackers who are trying to guess your users' passwords or use brute-force methods to get in. It recognizes sign-ins coming from valid users and treats them differently than the ones of malicious hackers and other unknown sources.

- **Azure AD Application Proxy** - You can provision security-enhanced remote access to on-premises web applications.
- **Single sign-on (SSO)** - Access to your applications. This includes thousands of pre-integrated SaaS apps.
- **Azure AD Connect** - Create and manage a single identity for each user across your hybrid enterprise, keeping users, groups, and devices in sync.

These are all great options which deter someone **guessing** or **brute-forcing** a password. However, sometimes passwords are obtained through social engineering, or poor physical security practices (like putting your password on a sticky note under your keyboard!). In these cases, the above features won't stop an intrusion. Instead, security administrators will want to turn to **Azure AD Multi-Factor Authentication (MFA)**.

What is Azure AD MFA?

Azure AD Multi-Factor Authentication (MFA) supplies added security for your identities by requiring two or more elements for full authentication.

These elements fall into three categories:

- **Something you know** - which might be a password or the answer to a security question.
- **Something you possess** - which might be a mobile app that receives a notification or a token-generating device.
- **Something you are** - which typically is a biometric property, such as a fingerprint or face scan used on many mobile devices.

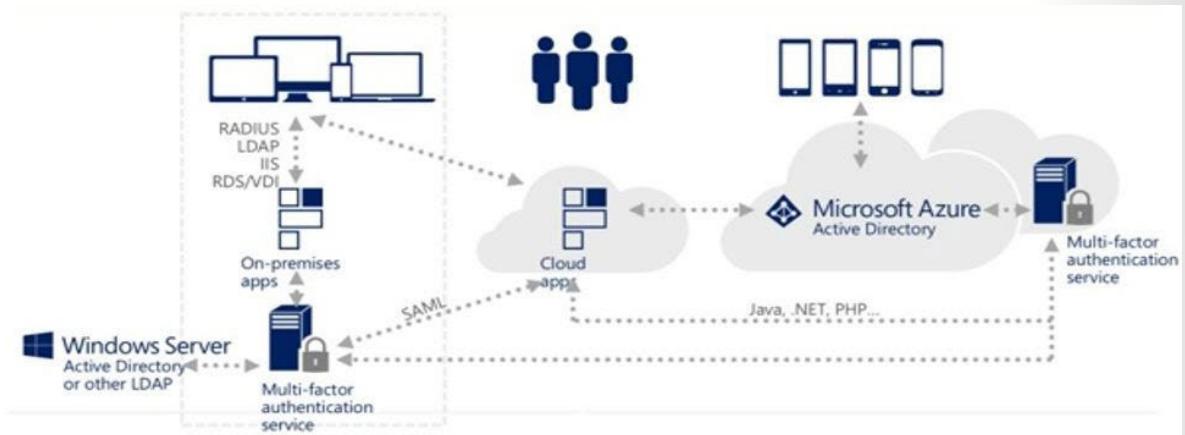


Using Azure AD MFA increases identity security by limiting the impact of credential exposure. To fully authenticate, a malicious hacker who has a user's password would also need their phone or their fingerprint. Authentication with only a single factor is insufficient, and without authentication from Azure AD MFA, a malicious hacker is unable to use those credentials to authenticate. You should enable Azure AD MFA wherever possible, because it adds enormous benefits to security.

Azure AD MFA is the Microsoft two-step verification solution. Azure AD MFA helps safeguard access to data and applications while meeting user demand for a simple sign-in process. It delivers strong authentication via a range of verification methods, including phone call, text message, or mobile app verification. The security of Azure AD MFA lies in its layered approach. Compromising multiple authentication factors presents a significant challenge for malicious hackers. Even if a malicious hacker manages to learn the user's password, it is useless without also possessing the trusted device. If the user loses the device, a person who finds it won't be able to use it without the user's password.

How Multi-Factor Authentication works

Here's what happens when someone tries to connect to a resource that's security enhanced by Azure AD MFA, and the service is on-premises:



1. The local Azure AD MFA service validates the initial sign-in request by passing the authentication request to on-premises Active Directory.
2. If the correct credentials were entered and validated, the service sends the request to **Azure AD Multi-Factor Authentication Server**.
3. The Azure AD Multi-Factor Authentication Server sends an additional verification challenge to the user. The methods you can easily configure are:
 - **Phone call** - Azure AD Multi-Factor Authentication Server places a call to the user's registered phone.
 - **Text message** - Azure AD Multi-Factor Authentication Server sends a six-digit code to the user's mobile phone.
 - **Mobile app notification** - Azure AD Multi-Factor Authentication Server sends a verification request to a user's smartphone, which asks them to complete the verification by selecting Verify in the mobile app.
 - **Mobile app verification code** - Azure AD Multi-Factor Authentication Server sends a six-digit code to the user's mobile app. The user then enters this code on the sign-in page.
 - **Initiative for Open Authentication (OATH) compliant tokens** - You can use these as a verification method.

If the service is running in Azure:

1. The service sends the sign-in request first to Azure AD for the initial validation and then to Azure AD Multi-Factor Authentication Server.
2. Azure AD Multi-Factor Authentication Server sends an additional verification challenge to the user, as just described.

Azure AD MFA allows the provider of the request service to validate that users are real people and not bots, that they have their devices with them, and that they can provide any additional information.

Azure AD MFA improves security for the requesting users, because someone can't easily impersonate them. You should require Azure AD MFA on all services, especially on mobile services.

How to get Multi-Factor Authentication?

Multi-Factor Authentication comes as part of the following offerings:

- **Azure Active Directory Premium or Microsoft 365 Business** - Both of these offerings support Azure AD Multi-Factor Authentication using Conditional Access policies to require multi-factor authentication.
- **Azure AD Free** or standalone **Microsoft 365** licenses - Use pre-created Conditional Access baseline protection policies to require multi-factor authentication for your users and administrators.
- **Azure Active Directory Global Administrators** - A subset of Azure AD Multi-Factor Authentication capabilities are available as a means to protect global administrator accounts.

Planning for MFA

Before starting a deployment of Azure AD Multi-Factor Authentication, there are several things you should decide.

First, consider rolling out MFA in waves. Start with a small group of pilot users to evaluate the complexity of your environment and identify any setup issues or unsupported apps or devices. Then broaden that group over time and evaluating the results with each pass until your entire company is enrolled.

Next, make sure to create a full communication plan. Azure AD MFA has several user interaction requirements including a registration process. Keep users informed every step of the way and let them know what they are required to do, important dates, and how to get answers to questions if they have trouble. Microsoft provides **communication templates**¹ including posters, and email templates to help draft your communications.

Azure AD MFA policies

Azure AD Multi-factor Authentication is enforced with **Conditional Access** policies. Conditional Access policies are **IF-THEN** statements. **IF** a user wants to access a resource, **THEN** they must complete an action. For example, a payroll manager wants to access the payroll application and is required to perform multi-factor authentication to access it. Other common access requests that might require MFA include:

- IF a specific cloud application is accessed
- IF a user is accessing a specific network
- IF a user is accessing a specific client application
- IF a user is registering a new device

Deciding supported authentication methods

When you turn on Azure AD MFA, you can choose the authentication methods you want to make available. You should always support more than one method so users have a backup option in case their primary method is unavailable. You can choose from the following methods:

¹ https://www.microsoft.com/download/details.aspx?id=57600&WT.mc_id=rss_alldownloads_all

Method	Description
Mobile App Verification code	A mobile authentication app such as the Microsoft Authenticator app can be used to retrieve an OATH verification code which is then entered into the sign-in interface. This code is changed every 30 seconds and the app works even if connectivity is limited. Note that this approach doesn't work in China on Android devices.
Call to a phone	Azure can call a supplied phone number. The user then approves the authentication using the keypad. This is a preferred backup method.
Text message to a phone	A text message with a verification code can be sent to a mobile phone. The user then enters the verification code into the sign-in interface to complete the authentication.

Administrators can enable one or more of the options above and then users can opt-in to each support authentication method they want to use.

Selecting an authentication method

Finally, you must decide how users will register their selected methods. The easiest approach is to use **Azure Active Directory Identity Protection**. If your organization has licenses for Identity Protection, you can configure it to prompt users to register for MFA the next time they sign in.

Users can also be prompted to register for MFA when they try to use an application or service that requires multi-factor authentication. Finally, you can enforce registration using a Conditional Access policy applied to an Azure group containing all users in your organization. This approach requires some manual work to periodically review the group to remove registered users. There are some **useful scripts in the documentation**² to automate some of this process.

Configure Authentication Methods

As mentioned earlier, it's recommended that administrators enable users to be able to select more than one authentication method in case their primary method is unavailable.

When a user signs into a service that requires MFA the first time, they will be asked to register their preferred multi-factor authentication method as shown in the following screenshot.

² <https://docs.microsoft.com/azure/active-directory/authentication/howto-mfa-getstarted#enforcing-registration>

 Microsoft | ?

Additional security verification

Secure your account by adding phone verification to your password. View video to know how to secure your account

Step 1: How should we contact you?

Authentication phone

Select your country or region

Method

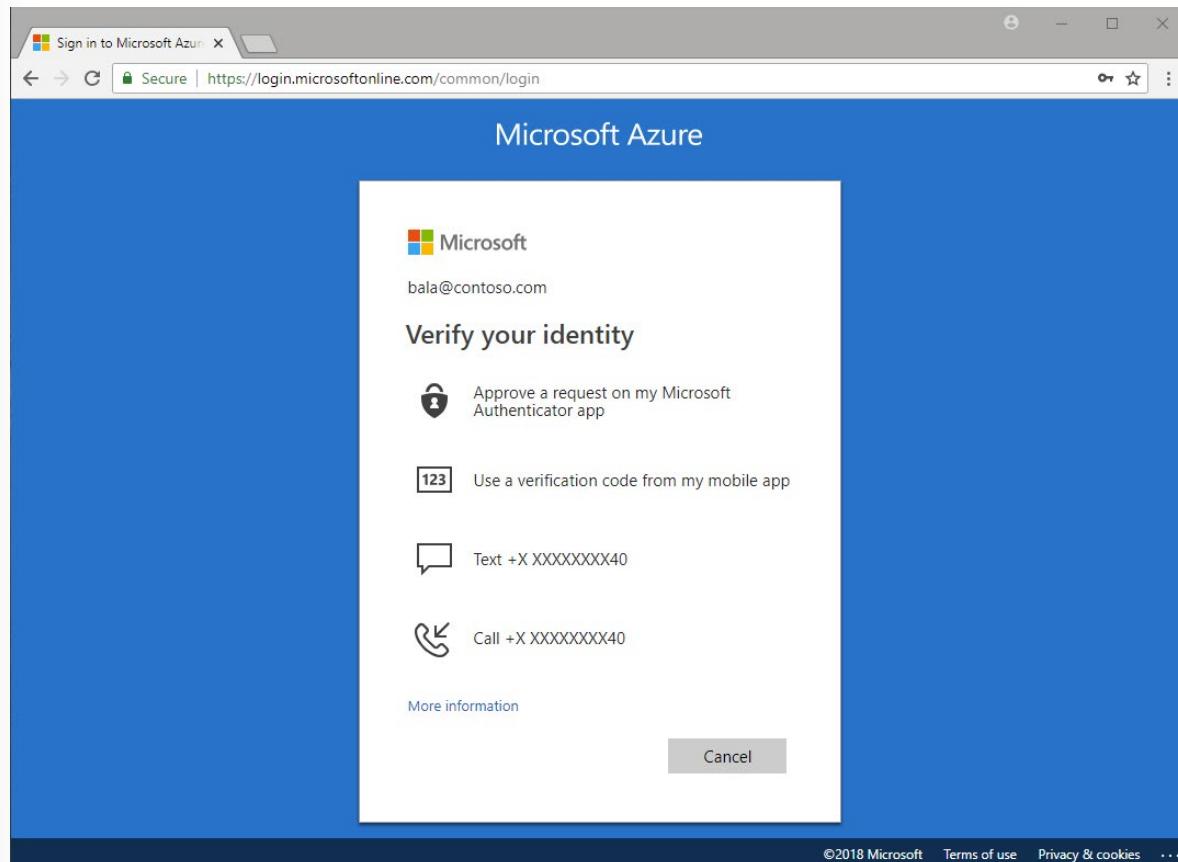
Send me a code by text message

Call me

Your phone numbers will only be used for account security. Standard telephone and SMS charges will apply.

Tip - If you followed the previous exercise and turned on MFA for an account and app, then you can try accessing that app with the given user account and you should see the above flow.

Once they've registered, each time they sign into a service or app that requires MFA the Azure login process will prompt for the additional authentication information as shown in the following image.



Azure Authentication Methods

As we saw earlier, there are several possible authentication methods that an administrator can set up. Some of these also support Self-Service Password Reset (SSPR) which allows users to reset their password by supplying a secondary form of authentication. You can couple this service with Azure AD MFA to ease the burden on IT staff.

The following table lists the authentication methods and the services that can use them.

Authentication method	Services
Password	Azure AD MFA and SSPR
Security questions	SSPR
Email address	SSPR
Microsoft Authenticator app	Azure AD MFA and SSPR
OATH hardware token	Azure AD MFA and SSPR
Text message	Azure AD MFA and SSPR
Voice call	Azure AD MFA and SSPR
App passwords	Azure AD MFA in certain cases

Let's explore these in a bit more detail.

Password

This is the only method that you can't disable.

Security questions

This method is available only for non-administrative accounts that use Self-Service Password Reset.

- Azure stores security questions privately and in a security-enhanced manner on a user object in the directory. Only users can answer the questions and only during registration. An administrator can't read or change a user's questions or answers.
- Azure provides 35 predefined questions, all translated and localized based on the browser locale.
- You can customize the questions by using the administrative interface; however, Azure displays them in the language entered. The maximum length is 200 characters.

Email address

This method is available only in SSPR. It's recommended that you avoid the use of an email account that doesn't require the user's Azure AD password to access it.

Microsoft Authenticator app

This method is available for Android and iOS. Users can register their mobile app at <https://aka.ms/mfasetup>

- The Microsoft Authenticator app helps prevent unauthorized access to accounts and helps stop fraudulent transactions by pushing a notification to your smartphone or tablet. Users view the notification and, if it's legitimate, select Verify. Otherwise, they select Deny.

- Users can use the Microsoft Authenticator app or a third-party app as a software token to generate an OATH verification code. After entering the username and password, the users enter the code provided by the app on the sign-in screen. The verification code provides a second form of authentication.

OATH hardware tokens

OATH is an open standard that specifies how to generate one-time password codes. Azure AD supports the use of OATH-TOTP SHA-1 tokens of the 30-second or 60-second variety. Customers can get these tokens from the vendor of their choice. Note that secret keys are limited to 128 characters, which might not be compatible with all tokens.

Text message

Azure sends a verification code to a mobile phone using SMS. The user must enter the code into the browser within a specific time period to continue.

Voice call

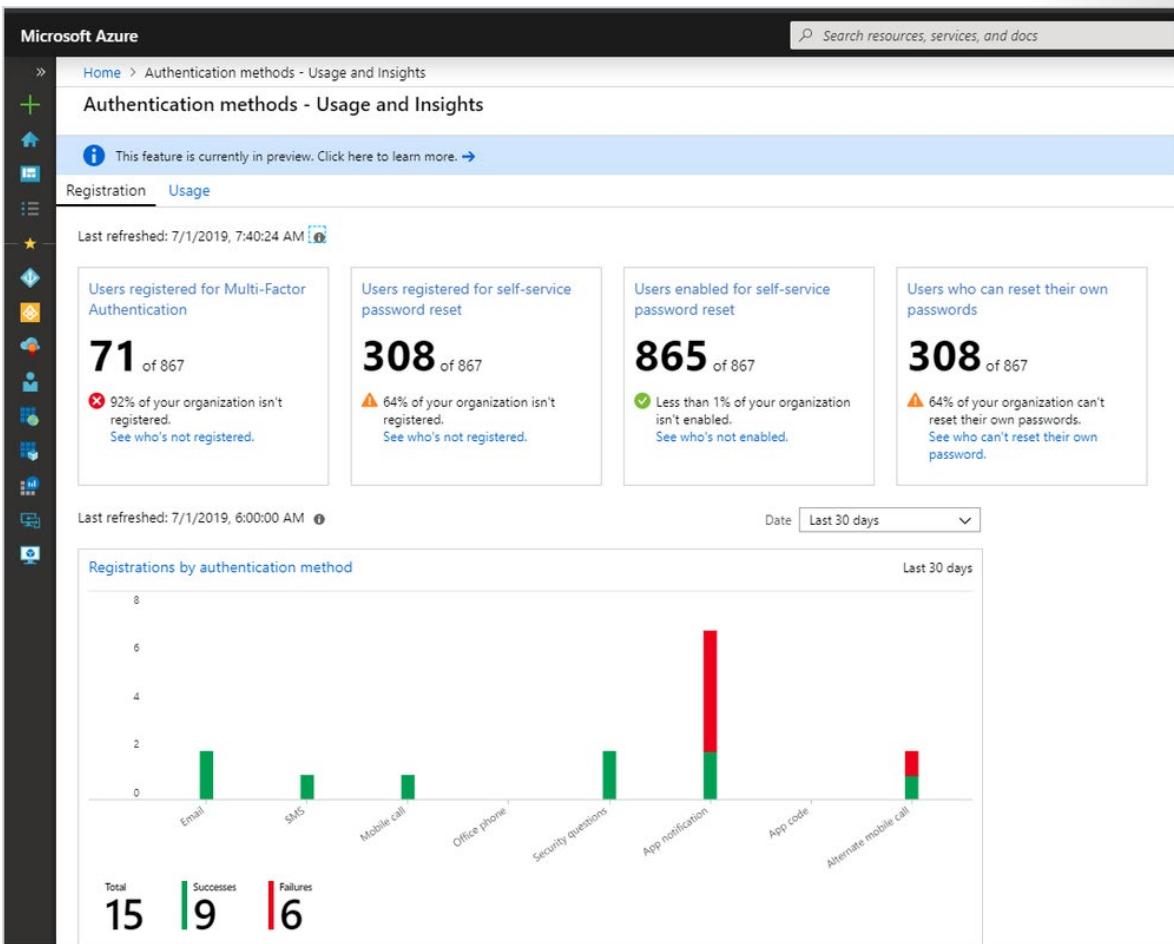
Azure uses an automated voice system to call the number and the owner uses the keypad to confirm the authentication. Note that this option is not available to the free/trial Azure AD tier.

App password

Certain non-browser apps don't support Azure AD MFA. If users are enabled for Azure AD MFA and try to use non-browser apps, they'll be unable to authenticate. The app password allows users to continue to authenticate.

Monitoring adoption

Azure AD includes a **Usage & insights** view in the **Monitoring** section where you can monitor the authentication methods activity. From here you can view the adoption of MFA and SSPR:

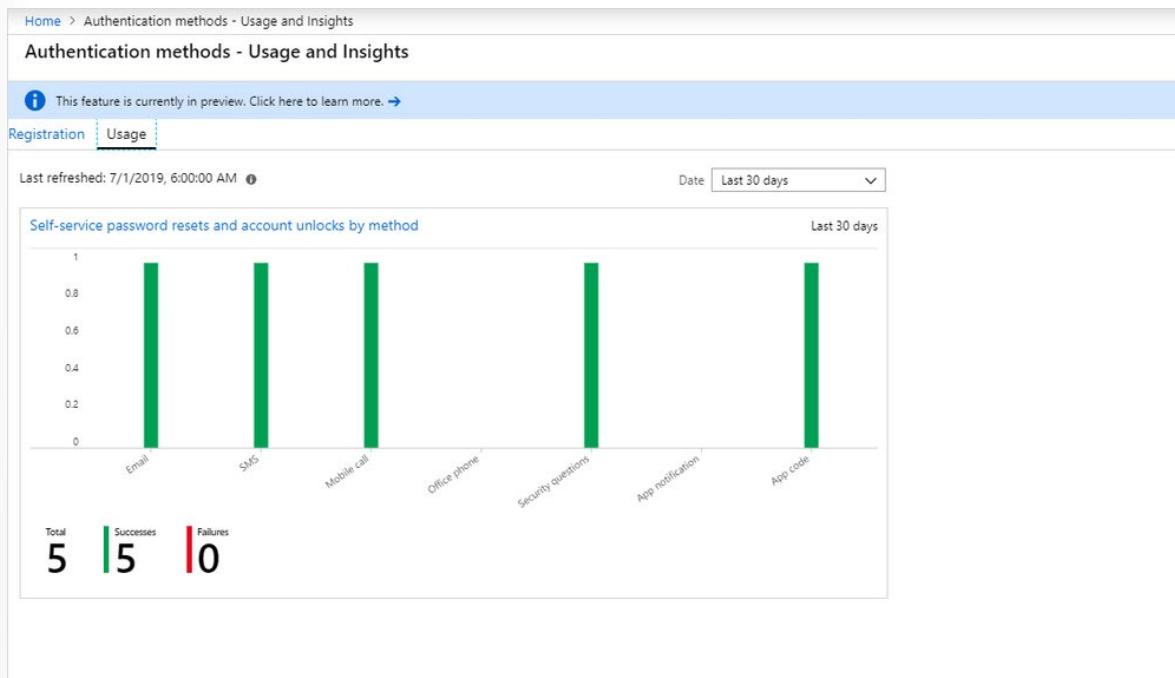


In addition to the overall registration numbers, you can also see the success and failure of registrations per authentication method. This allows you to understand which authentication methods your users most commonly registered and which ones are easy for them to register. This data is calculated using the last 30 days of audit logs from the combined security info registration and SSPR registration experiences.

You can drill down and see the latest registration audit information for each user by clicking the chart.

Insights - Authentication methods registration details							
Columns		Refresh	Download				
Name <input type="text"/>		SSPR Registered <input type="button" value="All"/>	SSPR Enabled <input type="button" value="All"/>	MFA Registered <input type="button" value="Registered"/>	Methods Registered <input type="button" value="All"/>		
Last updated: 7/1/2019, 8:50:00 AM							
USER NAME	USER	SSPR REGISTERED	SSPR ENABLED	MFA REGISTERED	METHODS REGISTERED		
abrekkan@f128.info	Abby Brekkan	Registered	Enabled	Registered	Mobile phone, Office phone		
adam@f128.info	Adam Steenwyk	Registered	Enabled	Registered	Email, Mobile phone, App code		
aditya@f128.info	Aditya	Registered	Enabled	Registered	Email, Mobile phone, App code		
audrey.z.oliver@gmail.com#EXT#@f128.onmicrosoft...	audrey.z.oliver	Registered	Enabled	Registered	Mobile phone		
cgreen@f128.info	Chris Green	Registered	Enabled	Registered	Mobile phone, App code		
chris@f128.info	Chris Padgett	Registered	Enabled	Registered	Email, Mobile phone, App code		
deloitte1@f128.info	Deloitte1	Registered	Enabled	Registered	Email, Mobile phone, App code		
ehuntington@f128.info	Edward Huntington	Registered	Enabled	Registered	Email, Mobile phone, Office phone, Security ques...		
elijah@f128.info	Elijah Henry	Registered	Enabled	Registered	Email, Mobile phone, App code		
frankm@f128.info	Frank Miller	Registered	Enabled	Registered	App code		
jamesblue@f128.info	James Blue	Registered	Enabled	Registered	App code		
joflores_microsoft.onmicrosoft.com#EXT#@f128.onmicrosoft.c...	John Flores	Registered	Enabled	Registered	Mobile phone		
joraja@f128.info	Jose Rojas	Registered	Enabled	Registered	App code		
julija1@f128.info	julija1	Registered	Enabled	Registered	Mobile phone, App code		
laura@f128.info	Laura Jones	Registered	Enabled	Registered	Email, Mobile phone		

You can also learn more about SSPR usage in your organization through the **Usage** tab on the main view as shown in the following image.



Manage User Authentication

Introduction

One of the main features of an identity platform is to verify, or authenticate, credentials when a user signs in to a device, application, or service. In Azure Active Directory (Azure AD), authentication involves more than just verifying a username and password. To improve security and reduce the need for help desk assistance, Azure AD authentication includes the following components:

- Self-service password reset
- Azure AD Multi-Factor Authentication
- Hybrid integration to write password changes back to on-premises environment
- Hybrid integration to enforce password protection policies for an on-premises environment
- Passwordless authentication

This module examines these components and explains how to plan, implement, and manage user authentication in Azure AD.

Learning objectives

In this module, you will:

- Administer authentication methods (FIDO2/Passwordless).
- Implement an authentication solution based on Windows Hello for Business.
- Configure and deploy self-service password reset.
- Deploy and manage password protection.
- Implement and manage tenant restrictions.

Administer FIDO 2 and Passwordless Authentication

As part of the sign-in experience for accounts in Azure AD, there are different ways that users can authenticate themselves. Historically, a username and password is the most common way a user would provide credentials. With modern authentication and security features in Azure AD, that basic password should be supplemented or replaced with more secure authentication methods.

Bad: Password	Good: Password and...	Better: Password and...	Best: Passwordless
123456	 SMS	 Microsoft Authenticator	 Windows Hello
qwertystuff			 Microsoft Authenticator (Preview)
password	 Voice	 Software Tokens OTP	 FIDO2 security key (Preview)
iloveyou		 Hardware Tokens OTP (Preview)	
Password1			

Passwordless authentication methods such as Windows Hello, FIDO2 security keys, and the Microsoft Authenticator app provide the most secure sign-in events.

Azure AD Multi-Factor Authentication (MFA) adds additional security over only using a password when a user signs in. The user can be prompted for additional forms of authentication, such as to respond to a push notification, enter a code from a software or hardware token, or respond to an SMS or phone call.

To simplify the user on-boarding experience and register for both MFA and self-service password reset (SSPR), we recommend that you enable combined security information registration. For resiliency, we recommend that you require users to register multiple authentication methods. When one method isn't available for a user during sign-in or SSPR, they can choose to authenticate with another method.

Authentication method strength and security

When you deploy features like Azure AD Multi-Factor Authentication in your organization, review the available authentication methods. Choose the methods that meet or exceed your requirements in terms of security, usability, and availability. Where possible, use authentication methods with the highest level of security.

The following table outlines the security considerations for the available authentication methods. Availability is an indication of the user being able to use the authentication method, not of the service availability in Azure AD:

Authentication method	Security	Usability	Availability
Windows Hello for Business	High	High	High
Microsoft Authenticator app	High	High	High
FIDO2 security key (preview)	High	High	High
OATH hardware tokens (preview)	Medium	Medium	High
OATH software tokens	Medium	Medium	High

Authentication method	Security	Usability	Availability
SMS	Medium	High	Medium
Voice	Medium	Medium	Medium
Password	Low	High	High

Tip - For flexibility and usability, we recommend that you use the Microsoft Authenticator app. This authentication method provides the best user experience and multiple modes, such as passwordless, MFA push notifications, and OATH codes.

How each authentication method works

Some authentication methods can be used as the primary factor when you sign-in to an application or device, such as using a FIDO2 security key or a password. Other authentication methods are only available as a secondary factor when you use Azure AD Multi-Factor Authentication or SSPR.

The following table outlines when an authentication method can be used during a sign-in event:

Method	Primary authentication	Secondary authentication
Windows Hello for Business	Yes	MFA
Microsoft Authenticator app	Yes (preview)	MFA and SSPR
FIDO2 security key (preview)	Yes	MFA
OATH hardware tokens (preview)	No	MFA
OATH software tokens	No	MFA
SMS	Yes (preview)	MFA and SSPR
Voice call	No	MFA and SSPR
Password	Yes	

All of these authentication methods can be configured in the Azure portal and increasingly using the Microsoft Graph REST API beta.

Note - In Azure AD, a password is often one of the primary authentication methods. You can't disable the password authentication method. If you use a password as the primary authentication factor, increase the security of sign-in events using Azure AD Multi-Factor Authentication.

The following additional verification methods can be used in certain scenarios:

- App passwords - used for old applications that don't support modern authentication and can be configured for per-user Azure AD Multi-Factor Authentication.
- Security questions - only used for SSPR.
- Email address - only used for SSPR.

What is FIDO2

The FIDO (Fast IDentity Online) Alliance helps to promote open authentication specifications and reduce the use of passwords as a form of authentication. FIDO2 is the latest specification that incorporates the web authentication (WebAuthn) specification. Users can register and then select a FIDO2 security key at the sign-in interface as their main means of authentication. These FIDO2 security keys are typically USB devices, but could also use Bluetooth or NFC. With a hardware device that handles the authentication, the security of an account is increased as there's no password that could be exposed or guessed. FIDO2 security keys can be used to sign in to their Azure AD or hybrid Azure AD joined Windows 10 devices and get single-sign on to their cloud and on-premises resources. Users can also sign in to supported brows-

ers. FIDO2 security keys are a great option for enterprises who are very security sensitive or have scenarios or employees who aren't willing or able to use their phone as a second factor.

- FIDO2 security keys are an unphishable specification-based passwordless authentication method that can come in any form factor
- Fast Identity Online (FIDO) is an open specification for passwordless authentication
- FIDO allows users and organizations to leverage the specification to sign in to their resources without a username or password using an external security key or a platform key built into a device

Enable FIDO2 security key method

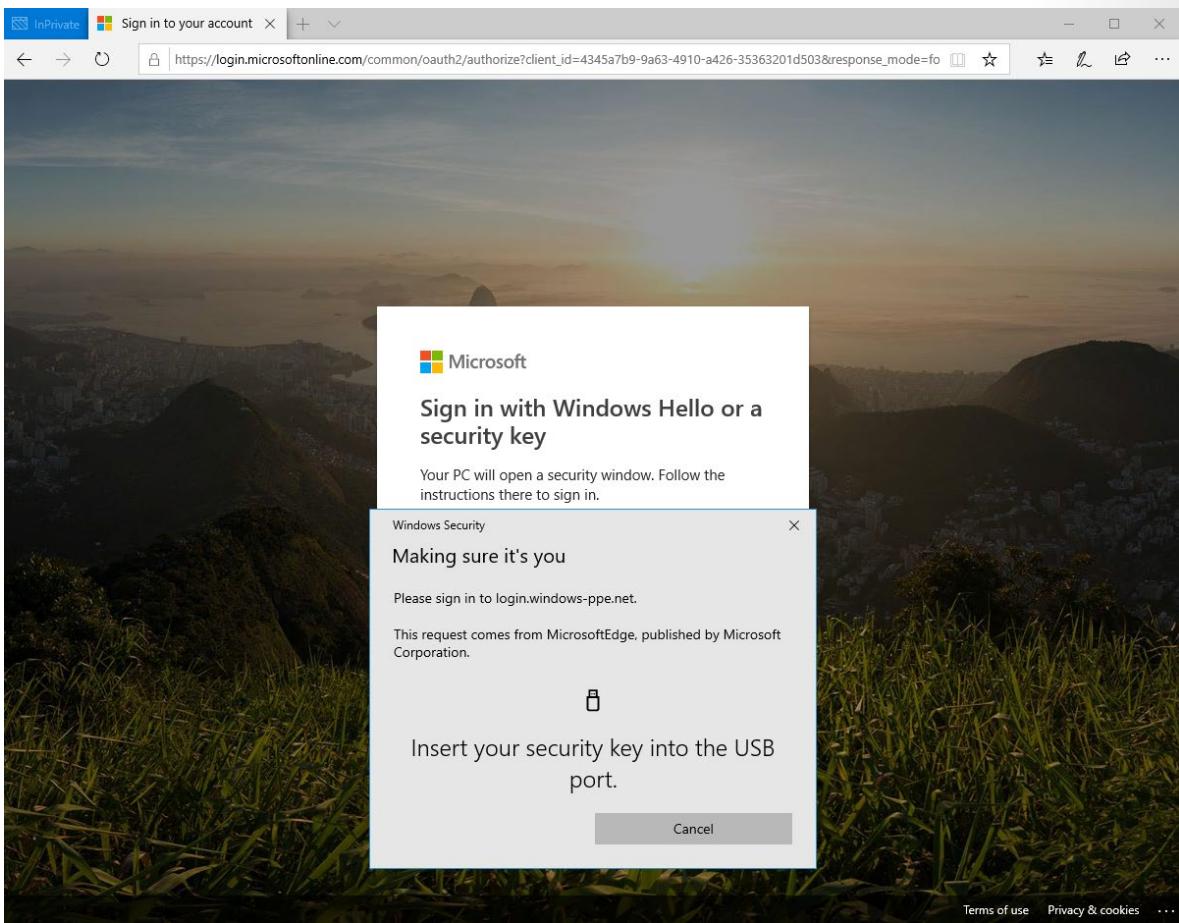
1. Sign in to the Azure portal.
2. Browse to **Azure Active Directory - Security - Authentication methods - Authentication method policy (Preview)**.
3. Under the method **FIDO2 Security Key**, choose the following options:
 - **Enable** - Yes or No
 - **Target** - All users or Select users
4. **Save** the configuration.

Manage user registration and FIDO2 security keys

1. Browse to <https://myprofile.microsoft.com>.
2. Sign in if you haven't already.
3. Click **Security Info**.
4. If the user already has at least one Azure AD Multi-Factor Authentication method registered, they can immediately register a FIDO2 security key.
5. If they don't have at least one Azure AD Multi-Factor Authentication method registered, they must add one.
6. Add a FIDO2 security key by clicking **Add method** and choosing **Security key**.
7. Choose **USB device** or **NFC device**.
8. Have your key ready and choose **Next**.
9. A box will appear and ask the user to create/enter a PIN for your security key and then perform the required gesture for the key, either biometric or touch.
10. The user will be returned to the combined registration experience and asked to provide a meaningful name for the key so the user can identify which one if they have multiple. Click **Next**.
11. Click **Done** to complete the process.

Sign in with passwordless credential

In the example below a user has already provisioned their FIDO2 security key. The user can choose to sign in on the web with their FIDO2 security key inside of a supported browser on Windows 10 version 1903 or higher.



Prerequisites for cloud-only deployment

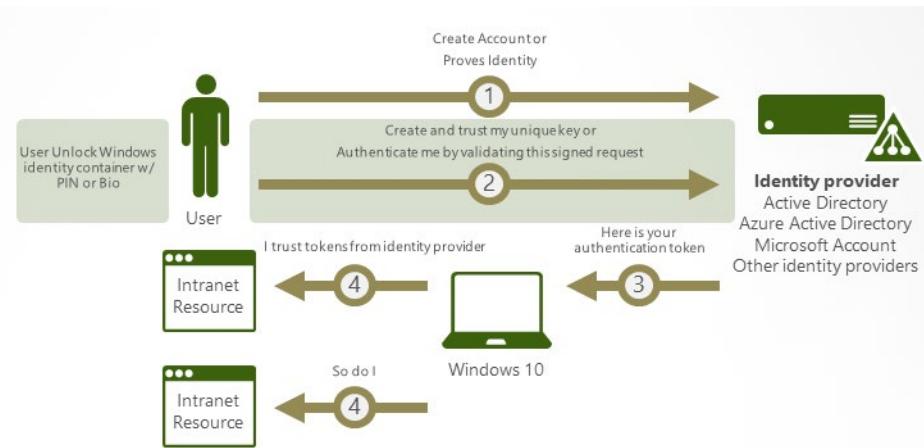
- Windows 10, version 1511 or later
- Microsoft Azure account
- Azure AD
- Azure AD Multi-Factor Authentication
- Modern Management - **optional**, Intune or supported third-party mobile-device management (MDM)
- Azure AD Premium subscription - **optional**, needed for automatic MDM enrollment when the device joins Azure AD

Implement Authentication based on Windows Hello for Business

In Windows 10, Windows Hello for Business replaces passwords with strong two-factor authentication on PCs and mobile devices. This authentication consists of a new type of user credential that is tied to a device and uses a biometric or PIN. Windows Hello for Business lets user authenticate to an Active Directory or Azure AD account.

Windows Hello addresses the following problems with passwords:

- Strong passwords can be difficult to remember, and users often reuse passwords on multiple sites.
- Server breaches can expose symmetric network credentials (passwords).
- Passwords are subject to replay attacks.
- Users can inadvertently expose their passwords due to phishing attacks.



How Windows Hello for Business works: key points

- Windows Hello credentials are based on certificate or asymmetrical key pair. Windows Hello credentials can be bound to the device, and the token that is obtained using the credential is also bound to the device.
- Identity provider (such as Active Directory, Azure AD, or a Microsoft account) validates user identity and maps the Windows Hello public key to a user account during the registration step.
- Keys can be generated in hardware (TPM 1.2 or 2.0 for enterprises, and TPM 2.0 for consumers) or software, based on the policy.
- Authentication is the two-factor authentication with the combination of a key or certificate tied to a device and something that the person knows (a PIN) or something that the person is (biometrics). The Windows Hello gesture does not roam between devices and is not shared with the server. Biometrics templates are stored locally on a device. The PIN is never stored or shared.
- The private key never leaves a device when using TPM. The authenticating server has a public key that is mapped to the user account during the registration process.
- PIN entry and biometric gesture both trigger Windows 10 to use the private key to cryptographically sign data that is sent to the identity provider. The identity provider verifies the user's identity and authenticates the user.
- Personal (Microsoft account) and corporate (Active Directory or Azure AD) accounts use a single container for keys. All keys are separated by identity providers' domains to help ensure user privacy.
- Certificate private keys can be protected by the Windows Hello container and the Windows Hello gesture.

Creating security groups

Windows Hello for Business uses several security groups to simplify the deployment and management.

Important - If your environment has one or more Windows Server 2016 domain controllers in the domain to which you are deploying Windows Hello for Business, then skip the Create the KeyCredentials Admins Security Group. Domains that include Windows Server 2016 domain controllers use the KeyAdmins group, which is created during the installation of the first Windows Server 2016 domain controller.

Create the KeyCredential Admins security group

Azure Active Directory Connect synchronizes the public key on the user object created during provisioning. You assign write and read permission to this group to the Active Directory attribute to ensure the Azure AD Connect service can add and remove keys as part of its normal workflow.

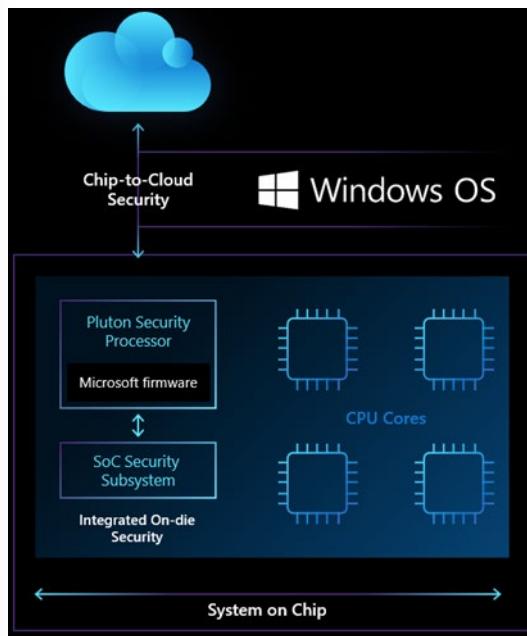
1. Sign in a domain controller or management workstation with *Domain Admin* equivalent credentials.
2. Open **Active Directory Users and Computers**.
3. Click **View** and click **Advance Features**.
4. Expand the domain node from the navigation pane.
5. Right-click the **Users** container. Click **New**. Click **Group**.
6. Type **KeyCredential Admins** in the **Group Name** text box.
7. Click **OK**.

Create the Windows Hello for Business Users security group

The Windows Hello for Business Users group is used to make it easy to deploy Windows Hello for Business in phases. You assign Group Policy and Certificate template permissions to this group to simplify the deployment by adding the users to the group. This provides users with the proper permissions to provision Windows Hello for Business and to enroll in the Windows Hello for Business authentication certificate.

1. Sign in a domain controller or management workstation with *Domain Admin* equivalent credentials.
2. Open **Active Directory Users and Computers**.
3. Click **View** and click **Advanced Features**.
4. Expand the domain node from the navigation pane.
5. Right-click the **Users** container. Click **New**. Click **Group**.
6. Type **Windows Hello for Business Users** in the **Group Name** text box.
7. Click **OK**.

Pluton Security Processor



Today, the heart of operating system security on most PCs lives in a chip separate from the CPU, called the Trusted Platform Module (TPM). The TPM is a hardware component which is used to help securely store keys and measurements that verify the integrity of the system. TPMs have been supported in Windows for more than 10 years and power many critical technologies such as Windows Hello and BitLocker. Given the effectiveness of the TPM at performing critical security tasks, attackers have begun to innovate ways to attack it, particularly in situations where an attacker can steal or temporarily gain physical access to a PC. These sophisticated attack techniques target the communication channel between the CPU and TPM, which is typically a bus interface. This bus interface provides the ability to share information between the main CPU and security processor, but it also provides an opportunity for attackers to steal or modify information in-transit using a physical attack.

The Pluton design removes the potential for that communication channel to be attacked by building security directly into the CPU. Windows PCs using the Pluton architecture will first emulate a TPM that works with the existing TPM specifications and APIs, which will allow customers to immediately benefit from enhanced security for Windows features that rely on TPMs like BitLocker and System Guard. Windows devices with Pluton will use the Pluton security processor to protect credentials, user identities, encryption keys, and personal data. None of this information can be removed from Pluton even if an attacker has installed malware or has complete physical possession of the PC.

- Built in collaboration with AMD, Intel, Qualcomm and others
- Security Hardware Cryptographic Key (SHACK)
- Update / replacement for the TPM chip, which hackers are starting to learn how to get around.
- Based on technology pioneered in Azure Sphere and Xbox security.

Deploy and Manage Password Protection

Users often create passwords that use common local words such as a school, sports team, or famous person. These passwords are easy to guess and weak against dictionary-based attacks. To enforce strong

passwords in your organization, Azure AD Password Protection provides a global and custom banned password list. A password change request fails if there's a match in these banned passwords list.

Azure AD Password Protection is designed with the following principles in mind:

- Domain controllers (DCs) never have to communicate directly with the internet.
- No new network ports are opened on DCs.
- No AD DS schema changes are required. The software uses the existing AD DS container and service-ConnectionPoint schema objects.
- No minimum AD DS domain or forest functional level (DFL/FFL) is required.
- The software doesn't create or require accounts in the AD DS domains that it protects.
- User clear-text passwords never leave the DC, either during password validation operations or at any other time.
- The software isn't dependent on other Azure AD features. For example, Azure AD password hash sync (PHS) isn't related or required for Azure AD Password Protection.
- Incremental deployment is supported, however the password policy is only enforced where the Domain Controller Agent (DC Agent) is installed.

Create an Azure account and add Azure Active Directory Premium P2 trial licenses

The tasks in this exercise and the exercises in this learning path require you to already have an Azure subscription that you can use or to sign up for an Azure trial account. If you already have your own Azure subscription, you may skip this task and continue to the next.

1. In a web browser, go to the [Azure portal³](#).
2. Scroll down through the page to learn more about the benefits and free services available.
3. Select **Start free**.
4. Use the wizard to sign up for your Azure trial subscription.
5. You will need to an Azure AD P2 license to complete some of the exercises. In the organization you created, search for and then select **Azure Active Directory**.
6. In the left navigation menu, select **Getting started**.
7. Under Getting started with Azure AD, select **Get a free trial for Azure AD Premium**.
8. In the Activate pane, under **AZURE AD PREMIUM P2**, select **Free trial** and then select **Activate**.
9. In the navigation menu on the left, select **Overview**.
10. Refresh the browser until you see Azure AD Premium P2 under the organization name. It may take a couple of minutes.
11. You may need to sign out and sign back into Microsoft Azure if you encounter any problems with expected features not being available.

³ <https://azure.microsoft.com/free/>

How Azure AD Password Protection works

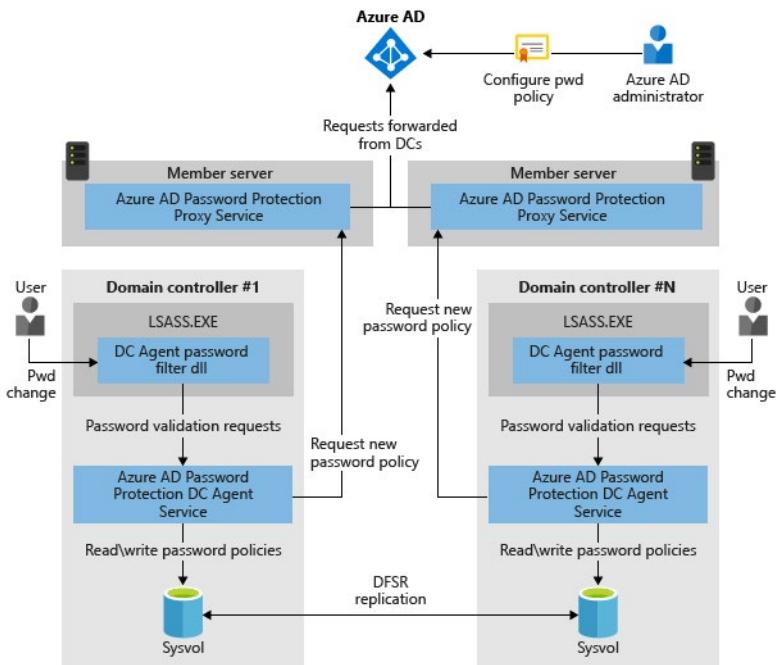
The on-premises Azure AD Password Protection components work as follows:

1. Each Azure AD Password Protection proxy service instance advertises itself to the DCs in the forest by creating a `serviceConnectionPoint` object in Active Directory.
2. Each DC Agent service for Azure AD Password Protection also creates a `serviceConnectionPoint` object in Active Directory. This object is used primarily for reporting and diagnostics.
3. The DC Agent service is responsible for initiating the download of a new password policy from Azure AD. The first step is to locate an Azure AD Password Protection proxy service by querying the forest for proxy `serviceConnectionPoint` objects.
4. When an available proxy service is found, the DC Agent sends a password policy download request to the proxy service. The proxy service in turn sends the request to Azure AD, and then returns the response to the DC Agent service.
5. After the DC Agent service receives a new password policy from Azure AD, the service stores the policy in a dedicated folder at the root of its domain `sysvol` folder share. The DC Agent service also monitors this folder in case newer policies replicate in from other DC Agent services in the domain.
6. The DC Agent service always requests a new policy at service startup. After the DC Agent service is started, it checks the age of the current locally available policy hourly. If the policy is older than one hour, the DC Agent requests a new policy from Azure AD via the proxy service, as described previously. If the current policy isn't older than one hour, the DC Agent continues to use that policy.
7. When password change events are received by a DC, the cached policy is used to determine if the new password is accepted or rejected.

To protect your on-premises Active Directory Domain Services (AD DS) environment, you can install and configure Azure AD Password Protection to work with your on-premises DC. This unit shows you how to install and register the Azure AD Password Protection proxy service and Azure AD Password Protection DC agent in your on-premises environment.

Deployment strategy

The following diagram shows how the basic components of Azure AD Password Protection work together in an on-premises Active Directory environment:



We recommend that you start deployments in *audit* mode. Audit mode is the default initial setting, where passwords can continue to be set. Passwords that would be blocked are recorded in the event log. After you deploy the proxy servers and DC agents in audit mode, monitor the impact that the password policy will have on users when the policy is enforced.

During the audit stage, many organizations find that the following situations apply:

- They need to improve existing operational processes to use more secure passwords.
- Users often use unsecure passwords.
- They need to inform users about the upcoming change in security enforcement, possible impact on them, and how to choose more secure passwords.

It's also possible for stronger password validation to affect your existing Active Directory domain controller deployment automation. We recommend that at least one DC promotion and one DC demotion happen during the audit period evaluation to help uncover issues such as weak passwords preventing promotion and demotion.

After the feature has been running in audit mode for a reasonable period, you can switch the configuration from *Audit* to *Enforce* to require more secure passwords. Additional monitoring during this time is a good idea.

Important - Azure AD Password Protection can only validate passwords during password change or set operations. Passwords that were accepted and stored in Active Directory prior to the deployment of Azure AD Password Protection will never be validated and will continue working as is. Over time, all users and accounts will eventually start using Azure AD Password Protection-validated passwords as their existing passwords expire. Accounts configured with "password never expires" are exempt from this.

Multiple forest considerations

There are no additional requirements to deploy Azure AD Password Protection across multiple forests.

Each forest is independently configured. Each Azure AD Password Protection proxy can only support domain controllers from the forest that it's joined to.

The Azure AD Password Protection software in any forest is unaware of password protection software that's deployed in other forests, regardless of Active Directory trust configurations.

Read-only domain controller considerations

Password change or set events aren't processed and persisted on read-only domain controllers (RODCs). Instead, they're forwarded to writable domain controllers. You don't have to install the Azure AD Password Protection DC agent software on RODCs.

Further, it's not supported to run the Azure AD Password Protection proxy service on a read-only domain controller.

High availability considerations

The main concern for password protection is the availability of Azure AD Password Protection proxy servers when the DCs in a forest try to download new policies or other data from Azure. Each Azure AD Password Protection DC agent uses a simple round-robin-style algorithm when deciding which proxy server to call. The agent skips proxy servers that aren't responding.

For most fully connected Active Directory deployments that have healthy replication of both directory and sysvol folder state, two Azure AD Password Protection proxy servers is enough to ensure availability. This configuration results in timely download of new policies and other data. You can deploy additional Azure AD Password Protection proxy servers if desired.

The design of the Azure AD Password Protection DC agent software mitigates the usual problems that are associated with high availability. The Azure AD Password Protection DC agent maintains a local cache of the most recently downloaded password policy. Even if all registered proxy servers become unavailable, the Azure AD Password Protection DC agents continue to enforce their cached password policy.

A reasonable update frequency for password policies in a large deployment is usually days, not hours or less. So, brief outages of the proxy servers don't significantly impact Azure AD Password Protection.

Deployment requirements

Licensing requirements for AD Password Protection are as follows:

Users	Azure AD Password Protection with global banned password list	Azure AD Password Protection with custom banned password list
Cloud-only users	Azure AD Free	Azure AD Premium P1 or P2
Users synchronized from on-premises AD DS	Azure AD Premium P1 or P2	Azure AD Premium P1 or P2

The following core requirements apply:

- You need an account that has Active Directory domain administrator privileges in the forest root domain to register the Windows Server Active Directory forest with Azure AD.
- The Key Distribution Service must be enabled on all domain controllers in the domain that run Windows Server 2012. By default, this service is enabled via manual trigger start.
- Network connectivity must exist between at least one domain controller in each domain and at least one server that hosts the proxy service for Azure AD Password Protection. This connectivity must allow the domain controller to access RPC endpoint mapper port 135 and the RPC server port on the proxy

service.

- By default, the RPC server port is a dynamic RPC port, but it can be configured to use a static port.
- All machines where the Azure AD Password Protection proxy service will be installed must have network access to the following endpoints:

Endpoint	Purpose
https://login.microsoftonline.com	Authentication requests
https://enterpriseregistration.windows.net	Azure AD Password Protection functionality

Azure AD Password Protection DC agent

The following requirements apply to the Azure AD Password Protection DC agent:

- All machines where the Azure AD Password Protection DC agent software will be installed must run Windows Server 2012 or later.
 - The Active Directory domain or forest doesn't need to be at Windows Server 2012 domain functional level (DFL) or forest functional level (FFL). There is no minimum DFL or FFL required for either the DC agent or proxy software to run.
- All machines that run the Azure AD Password Protection DC agent must have .NET 4.5 installed.
- Any Active Directory domain that runs the Azure AD Password Protection DC agent service must use Distributed File System Replication (DFSR) for sysvol replication.

Azure AD Password Protection proxy service

The following requirements apply to the Azure AD Password Protection proxy service:

- All machines where the Azure AD Password Protection proxy service will be installed must run Windows Server 2012 R2 or later.
Note - The Azure AD Password Protection proxy service deployment is a mandatory requirement for deploying Azure AD Password Protection even though the domain controller may have outbound direct internet connectivity.
- All machines where the Azure AD Password Protection proxy service will be installed must have .NET 4.7 installed.
- All machines that host the Azure AD Password Protection proxy service must be configured to grant domain controllers the ability to log on to the proxy service. This ability is controlled via the "Access this computer from the network" privilege assignment.
- All machines that host the Azure AD Password Protection proxy service must be configured to allow outbound TLS 1.2 HTTP traffic.
- A *Global Administrator* or *Security Administrator* account is required to register the Azure AD Password Protection proxy service and forest with Azure AD.
- Network access must be enabled for the set of ports and URLs specified in the **Application Proxy environment setup procedures**⁴.

⁴ <https://docs.microsoft.com/azure/active-directory/app-proxy/application-proxy-add-on-premises-application>

Warning - Azure AD Password Protection proxy and Azure AD Application Proxy install different versions of the Microsoft Azure AD Connect Agent Updater service, which is why the instructions refer to Application Proxy content. These different versions are incompatible when installed side by side. Doing so will prevent the Agent Updater service from contacting Azure for software updates, so you should never install Azure AD Password Protection Proxy and Application Proxy on the same machine.

Download required software

Two installers are required for an on-premises Azure AD Password Protection deployment:

- Azure AD Password Protection DC agent (*AzureADPasswordProtectionDCAgentSetup.msi*)
- Azure AD Password Protection proxy (*AzureADPasswordProtectionProxySetup.exe*)

Install and configure the proxy service

The Azure AD Password Protection proxy service is typically on a member server in your on-premises AD DS environment. Once installed, the Azure AD Password Protection proxy service communicates with Azure AD to maintain a copy of the global and customer banned password lists for your Azure AD tenant.

Install the DC agent service

To install the Azure AD Password Protection DC agent service, run the *AzureADPasswordProtectionDCAgentSetup.msi* package.

You can automate the software installation by using standard MSI procedures, as shown in the following example:

```
msiexec.exe /i AzureADPasswordProtectionDCAgentSetup.msi /quiet /qn /norestart
```

The */norestart* flag can be omitted if you prefer to have the installer automatically reboot the machine.

The software installation, or uninstallation, requires a restart. This requirement is because password filter DLLs are only loaded or unloaded by a restart.

The installation of on-premises Azure AD Password Protection is complete after the DC agent software is installed on a domain controller and that computer is rebooted. No other configuration is required or possible. Password change events against the on-premises DCs use the configured banned password lists from Azure AD.

Tip - You can install the Azure AD Password Protection DC agent on a machine that's not yet a domain controller. In this case, the service starts and runs but remains inactive until the machine is promoted to be a domain controller.

Upgrading the proxy service

The Azure AD Password Protection proxy service supports automatic upgrade. Automatic upgrade uses the Microsoft Azure AD Connect Agent Updater service, which is installed side by side with the proxy service. Automatic upgrade is on by default and may be enabled or disabled using the *Set-AzureADPasswordProtectionProxyConfiguration* cmdlet.

The current setting can be queried using the `Get-AzureADPasswordProtectionProxyConfiguration` cmdlet. We recommend that the automatic upgrade setting always is enabled.

The `Get-AzureADPasswordProtectionProxy` cmdlet may be used to query the software version of all currently installed Azure AD Password Protection proxy servers in a forest.

Manual upgrade process

A manual upgrade is accomplished by running the latest version of the `AzureADPasswordProtectionProxySetup.exe` software installer. The latest version of the software is available on the Microsoft Download Center.

It's not required to uninstall the current version of the Azure AD Password Protection proxy service—the installer performs an in-place upgrade. No reboot should be required when upgrading the proxy service. The software upgrade may be automated using standard MSI procedures, such as `AzureADPasswordProtectionProxySetup.exe /quiet`.

Upgrading the DC agent

When a newer version of the Azure AD Password Protection DC agent software is available, the upgrade is accomplished by running the latest version of the `AzureADPasswordProtectionDCAgentSetup.msi` software package. The latest version of the software is available on the Microsoft Download Center.

It's not required to uninstall the current version of the DC agent software—the installer performs an in-place upgrade. A reboot is always required when upgrading the DC agent software. This requirement is caused by core Windows behavior.

The software upgrade may be automated using standard MSI procedures, such as `msiexec.exe /i AzureADPasswordProtectionDCAgentSetup.msi /quiet /qn /norestart`.

You may omit the `/norestart` flag if you prefer to have the installer automatically reboot the machine.

The `Get-AzureADPasswordProtectionDCAgent` cmdlet may be used to query the software version of all currently installed Azure AD Password Protection DC agents in a forest.

Implement and Manage Tenant Restrictions

Large organizations that emphasize security want to move to cloud services like Microsoft 365, but they 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**⁵ and **login.microsoftonline.com**⁶. Blocking these addresses would keep users from accessing Outlook on the web entirely, instead of merely restricting them to approved identities and resources.

The Azure AD solution to this challenge is a feature called tenant restrictions. With tenant restrictions, organizations can control access to SaaS cloud applications, based on the Azure AD tenant the applications use for single sign-on. For example, you may 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 their users are permitted to access. Azure AD then only grants access to these permitted tenants.

⁵ <https://outlook.office.com/>

⁶ <https://login.microsoftonline.com/>

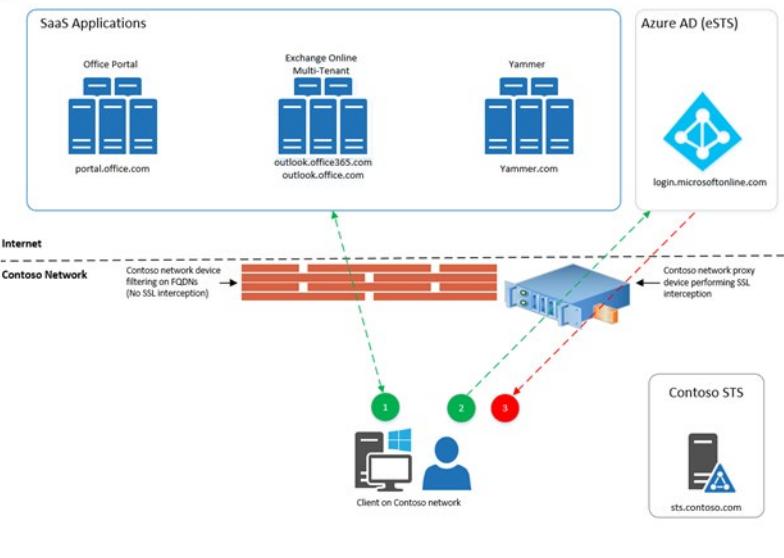
This article focuses on tenant restrictions for Microsoft 365, but the feature should work with any SaaS cloud app that uses modern authentication protocols with Azure AD for single sign-on. If you use SaaS apps with a different Azure AD tenant from the tenant used by Microsoft 365, make sure that all required tenants are permitted.

How it works

The overall solution comprises the following components:

- **Azure AD:** If the `Restrict-Access-To-Tenants: <permitted tenant list>` header is present, Azure AD only issues security tokens for the permitted tenants.
- **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 Azure AD.
- **Client software:** To support tenant restrictions, client software must request tokens directly from Azure AD, 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).
- **Modern Authentication:** Cloud services must use modern authentication to use tenant restrictions and block access to all non-permitted tenants. You must configure Microsoft 365 cloud services to use modern authentication protocols by default.

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



In the scenario depicted here, a user is trying to access a shared SaaS application via an allowed fully qualified domain name (FQDN) to get access to the Fabrikam application instance, or tenant, while on the Contoso network:

1. The client accesses the allowed FQDN (for example, outlook.office.com) and is redirected to Azure AD.
2. The client tries to use a Fabrikam credential. The Contoso proxy intercepts the traffic and inserts an HTTP header indicating that Contoso is an allowed tenant. Fabrikam's tenant is not allowed.

3. Azure AD does not issue a service token for the Fabrikam user, so the client cannot gain access to the Fabrikam SaaS application instance.

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 Azure AD URLs to authenticate: login.microsoftonline.com, login.microsoft.com, and login.windows.net. Additionally, to access Office 365, your clients must be able to connect to the fully qualified domain names (FQDNs), URLs, and IP addresses defined in **Office 365 URLs and IP address ranges**⁷.

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.
- Azure AD Premium 1 licenses are required for use of Tenant Restrictions.

Configuration

For each incoming 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 - When configuring SSL interception and header injection, ensure that traffic to https://device.login.microsoftonline.com is excluded. This URL is used for device authentication. Performing TLS break-and-inspect may interfere with Client Certificate authentication, which may cause issues with device registration and device-based Conditional Access.

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, as well as 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,72f988bf-86f1-41af-91ab-2d7cd011db47
- 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**: 456ff232-3512-5h23-b3b3-3236w0826f3d. You **must** use your own directory ID in this spot.

⁷ <https://support.office.com/article/Office-365-URLs-and-IP-address-ranges-8548a211-3fe7-47cb-abb1-355ea5aa88a2>

Tip - You can find your directory ID in the Azure Active Directory portal. Sign in as an administrator, select Azure Active Directory, then select Properties. To validate that a directory ID or domain name refers to the same tenant, use that ID or domain 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 non-approved tenants, the proxy needs to replace the **Restrict-Access-To-Tenants** header if it is 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.

Note - Do not include subdomains under `*.login.microsoftonline.com` in your proxy configuration. Doing so will include `device.login.microsoftonline.com` and may 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` from TLS break-and-inspect and header injection.

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 non-permitted tenant for the Contoso instance, the user sees an access denial message, which says you're trying to access a resource that belongs to an organization unapproved by your IT department.

Admin experience

While configuration of tenant restrictions is done on the corporate proxy infrastructure, admins can access the tenant restrictions reports in the Azure portal directly. To view the reports:

1. Sign in to the Azure Active Directory portal. The **Azure Active Directory admin center** dashboard appears.
2. In the left pane, select **Azure Active Directory**. The Azure Active Directory overview page appears.
3. On the Overview page, select **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.

Note - The report may 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 ("00000000-0000-0000-0000-00000000@domain.com").

Like other reports in the Azure portal, 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**
- **Application**
- **Status**
- **Date**
- **Date (UTC)** (where UTC is Coordinated Universal Time)
- **MFA Auth Method** (multifactor authentication method)
- **MFA Auth Detail** (multifactor authentication detail)
- **MFA Result**
- **IP Address**
- **Client**
- **Username**
- **Location**
- **Target tenant ID**

Microsoft 365 support

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

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

Microsoft 365 browser-based applications (the Office Portal, Yammer, SharePoint sites, Outlook on the Web, and more) 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 may still able to use legacy protocols against tenants where modern authentication isn't enabled, effectively bypassing tenant restrictions. Tenant restrictions may 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 may choose to implement restrictions preventing end users from adding unapproved mail accounts to their profiles. For example, see the Prevent adding non-default Exchange accounts group policy setting.

Plan, Implement, and Administer Conditional Access

Introduction

Conditional Access gives a fine granularity of control over which users can perform specific activities, access resources, and ensure data and systems are safe.

Learning objectives

In this module, you will:

- Plan and implement security defaults.
- Plan Conditional Access policies.
- Implement Conditional Access policy controls and assignments (targeting, applications, and conditions).
- Test and troubleshoot Conditional Access policies.
- Implement application controls.
- Implement session management.
- Configure smart lockout thresholds.

Plan for Security Defaults

Managing security can be difficult with common identity-related attacks like password spray, replay, and phishing becoming more and more popular. Security defaults provide secure default settings that Microsoft manages on behalf of organizations to keep customers safe until organizations are ready to manage their own identity security story. Security defaults provide preconfigured security settings, such as:

- Requiring all users to register for Azure Active Directory Multi-Factor Authentication.
- Requiring administrators to perform multi-factor authentication.
- Blocking legacy authentication protocols.
- Requiring users to perform multi-factor authentication when necessary.
- Protecting privileged activities like access to the Azure portal.

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with various icons and links. In the main area, under 'Contoso - Properties' (Azure Active Directory), there's a 'Properties' section. Below it, there's a 'Manage Security defaults' button. To the right, there's a detailed view of 'Directory properties' including fields for Name, Country or region, Location, Notification language, Directory ID, Technical contact, Global privacy contact, and Privacy statement URL. There's also a section for 'Access management for Azure resources' with a 'Yes' button. At the bottom right, there's a 'Save' button.

Availability

Microsoft security defaults are available to everyone. The goal is to ensure that all organizations have a basic level of security enabled at no extra cost. You turn on security defaults in the Azure portal. If your tenant was created on or after October 22, 2019, it is possible security defaults are already enabled in your tenant. To protect all of our users, the security defaults feature is being rolled out to all new tenants created.

Who's it for?

Who should use security defaults?	Who shouldn't use security defaults?
Organizations that want to increase their security posture but don't know how or where to start or Organizations utilizing the free tier of Azure Active Directory Licensing	Organizations currently using Conditional Access policies to bring signals together, make decisions, and enforce organizational policies or Organizations with Azure Active Directory Premium licenses or Organizations with complex security requirements that warrant using Conditional Access

Policies enforced

Unified multifactor authentication registration

All users in your tenant must register for multifactor authentication (MFA) in the form of the Azure Active Directory (Azure AD) Multi-Factor Authentication. Users have 14 days to register for Azure AD Multi-Factor Authentication by using the Microsoft Authenticator app. After the 14 days have passed, the user won't be able to sign in until registration is completed. A user's 14-day period begins after their first successful interactive sign-in after enabling security defaults.

Protecting administrators

Users with privileged access have increased access to your environment. Due to the power these accounts have, you should treat them with special care. One common method to improve the protection of privileged accounts is to require a stronger form of account verification for sign-in. In Azure AD, you can get a stronger account verification by requiring multifactor authentication.

After registration with Azure AD Multi-Factor Authentication is finished, the following nine Azure AD administrator roles will be required to perform additional authentication every time they sign in:

- Global Administrator
- SharePoint Administrator
- Exchange Administrator
- Conditional Access Administrator
- Security Administrator
- Helpdesk Administrator
- Billing Administrator
- User Administrator
- Authentication Administrator

Protecting all users

We tend to think that administrator accounts are the only accounts that need extra layers of authentication. Administrators have broad access to sensitive information and can make changes to subscription-wide settings. But attackers frequently target end users.

After these attackers gain access, they can request access to privileged information on behalf of the original account holder. They can even download the entire directory to perform a phishing attack on your whole organization.

One common method to improve protection for all users is to require a stronger form of account verification, such as multifactor authentication, for everyone. After users complete Azure Active Directory Multi-Factor Authentication registration, they'll be prompted for additional authentication whenever necessary. This functionality protects all applications registered with Azure AD, including SaaS applications.

Blocking legacy authentication

To give your users easy access to your cloud apps, Azure AD supports a variety of authentication protocols, including legacy authentication. **Legacy authentication** is an authentication request made by:

- Clients that don't use modern authentication (for example, an Office 2010 client). Modern authentication encompasses clients that implement protocols, such as OAuth 2.0, to support features like multifactor authentication and smart cards. Legacy authentication typically only supports less secure mechanisms like passwords.
- Any client that uses older mail protocols such as IMAP, SMTP, or POP3.

Today, the majority of compromising sign-in attempts come from legacy authentication. Legacy authentication does not support multifactor authentication. Even if you have a multifactor authentication policy enabled on your directory, an attacker can authenticate by using an older protocol and bypass multifactor authentication.

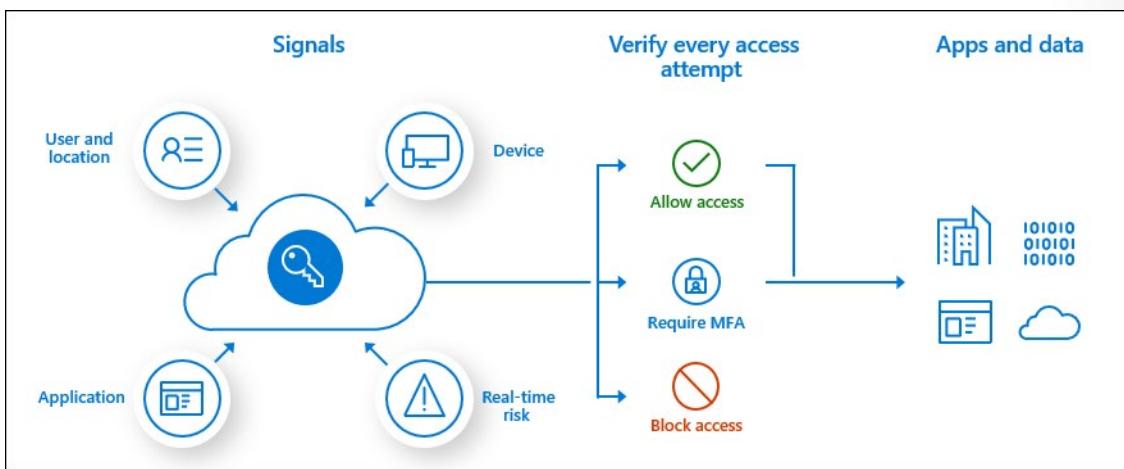
After security defaults are enabled in your tenant, all authentication requests made by an older protocol will be blocked. Security defaults blocks Exchange Active Sync basic authentication.

Plan your Conditional Access Policies

Planning your Conditional Access deployment is critical to achieving your organization's access strategy for apps and resources.

In a mobile-first, cloud-first world, your users access your organization's resources from anywhere using a variety of devices and apps. As a result, focusing on who can access a resource is no longer enough. You also need to consider where the user is, the device being used, the resource being accessed, and more.

Azure AD Conditional Access (CA) analyzes signals, such as user, device, and location, to automate decisions and enforce organizational access policies for resource. You can use CA policies to apply access controls like multifactor authentication (MFA). CA policies allow you to prompt users for MFA when needed for security and to stay out of users' way when not needed.



Although security defaults ensure a basic level of security, your organization may need more flexibility than security defaults offer. You can use CA to customize security defaults with more granularity and to configure new policies that meet your requirements.

Benefits

The benefits of deploying CA are:

- Increase productivity. Only interrupt users with a sign-in condition like MFA when one or more signals warrants it. CA policies allow you to control when users are prompted for MFA, when access is blocked, and when they must use a trusted device.
- Manage risk. Automating risk assessment with policy conditions means risky sign-ins are at once identified and remediated or blocked. Coupling Conditional Access with Identity Protection, which detects anomalies and suspicious events, allows you to target when access to resources is blocked or gated.
- Address compliance and governance. CA enables you to audit access to applications, present terms of use for consent, and restrict access based on compliance policies.
- Manage cost. Moving access policies to Azure AD reduces the reliance on custom or on-premises solutions for CA and their infrastructure costs.
- Zero trust. Conditional Access helps you move toward a zero-trust environment.

Understand Conditional Access policy components

CA policies are if-then statements: If an assignment is met, then apply these access controls. When configuring CA policies, conditions are called *assignments*. CA policies allow you to enforce access controls on your organization's apps based on certain assignments.

The screenshot shows the 'New' policy configuration page. At the top, there is a blue header bar with the text 'Want to switch back to the previous configuration experience? Click to leave the preview.' Below this, the 'Name' field is set to 'Device compliance app policy'. The 'Assignments' section is expanded, showing 'Users and groups' (0 selected), 'Cloud apps or actions' (0 selected), and 'Conditions' (0 selected). The 'Access controls' section is also expanded, showing 'Grant' (0 selected) and 'Session' (0 selected). On the right side, there are sections for 'Device platforms', 'Locations', 'Client apps (Preview)', and 'Device state (Preview)', all currently set to 'Not configured'.

Assignments define the users and groups to be affected by the policy, the cloud apps or actions to which the policy will apply, and the conditions under which the policy will apply. Access control settings grant or block access to different cloud apps and can enable limited experiences within specific cloud apps.

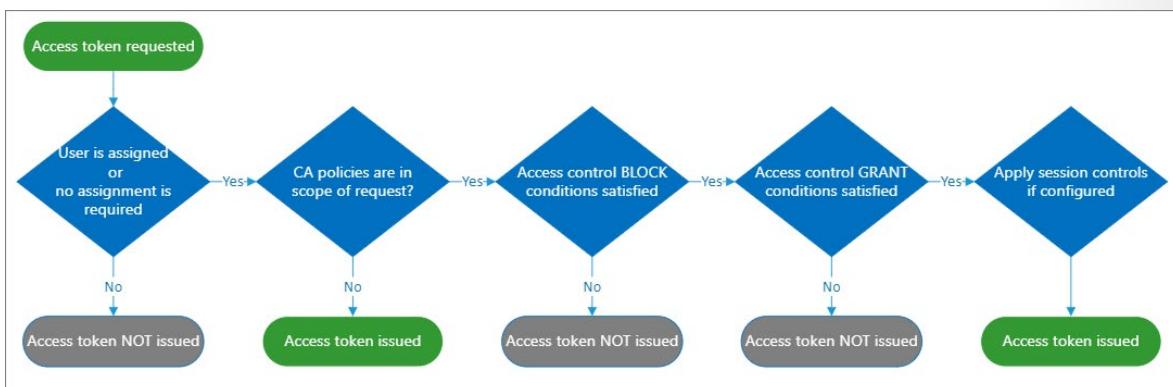
Some common questions about assignments, access controls, and session controls:

- Users and Groups: Which users and groups will be included in or excluded from the policy? Does this policy include all users, specific group of users, directory roles, or external users?
- Cloud apps or actions: What application(s) will the policy apply to? What user actions will be subject to this policy?
- Conditions: Which device platforms will be included in or excluded from the policy? What are the organization's trusted locations?
- Access controls: Do you want to grant access to resources by implementing requirements such as MFA, devices marked as compliant, or hybrid Azure AD joined devices?
- Session controls: Do you want to control access to cloud apps by implementing requirements such as app enforced permissions or Conditional Access App Control?

Access token issuance

Access tokens enable clients to securely call protected web APIs, and they're used by web APIs to perform authentication and authorization. Per the OAuth specification, access tokens are opaque strings without a set format. Some identity providers (IDPs) use GUIDs; others use encrypted blobs. The Microsoft identity platform uses a variety of access token formats depending on the configuration of the API that accepts the token.

It's important to understand how access tokens are issued.



Note - If no assignment is required, and no CA policy is in effect, the default behavior is to issue an access token.

For example, consider a policy where:

IF user is in Group 1, THEN force MFA to access App 1.

If a user not in Group 1 attempts to access the app, no **if** condition is met, and a token is issued. Excluding users outside of Group 1 requires a separate policy to block all other users.

Follow best practices

The Conditional Access framework provides you with great configuration flexibility. However, great flexibility also means you should carefully review each configuration policy before releasing it to avoid undesirable results.

Set up emergency access accounts

If you misconfigure a policy, it can lock the organizations out of the Azure portal. Mitigate the impact of accidental administrator lock out by creating two or more emergency access accounts in your organization. You will learn more about emergency access accounts later in this course.

Set up report-only mode

It can be difficult to predict the number and names of users affected by common deployment initiatives such as:

- Blocking legacy authentication.
- Requiring MFA.
- Implementing sign-in risk policies.

Report-only mode allows administrators to evaluate the impact of CA policies before enabling them in their environment.

Exclude countries from which you never expect a sign-in

Azure active directory allows you to create named locations. Create a named location that includes all of the countries from which you would never expect a sign-in to occur. Then create a policy for all apps that blocks sign in from that named location. **Be sure to exempt your administrators from this policy.**

Common policies

When planning your CA policy solution, assess whether you need to create policies to achieve the following outcomes.

- **Require MFA.** Common use cases include requiring MFA by admins, to specific apps, for all users, or from network locations you don't trust.
- **Respond to potentially compromised accounts.** Three default policies can be enabled: require all users to register for MFA, require a password change for users who are high-risk, and require MFA for users with medium or high sign-in risk.
- **Require managed devices.** The proliferation of supported devices to access your cloud resources helps to improve the productivity of your users. You probably don't want certain resources in your environment to be accessed by devices with an unknown protection level. For those resources, require that users can only access them using a managed device.
- **Require approved client applications.** Employees use their mobile devices for both personal and work tasks. For BYOD scenarios, you must decide whether to manage the entire device or just the data on it. If managing only data and access, you can require approved cloud apps that can protect your corporate data.
- **Block access.** Blocking access overrides all other assignments for a user and has the power to block your entire organization from signing on to your tenant. It can be used, for example, when you are migrating an app to Azure AD, but you aren't ready for anyone to sign in to it yet. You can also block certain network locations from accessing your cloud apps or block apps using legacy authentication from accessing your tenant resources.

Important - If you create a policy to block access for all users, be sure to exclude emergency access accounts and consider excluding all administrators from the policy.

Build and test policies

At each stage of your deployment, ensure that you're evaluating that results are as expected.

When new policies are ready, deploy them in phases in the production environment:

- Provide internal change communication to end users.
- Start with a small set of users, and verify that the policy behaves as expected.
- When you expand a policy to include more users, continue to exclude all administrators. Excluding administrators ensures that someone still has access to a policy if a change is required.
- Apply a policy to all users only after it's thoroughly tested. Ensure you have at least one administrator account to which a policy doesn't apply.

Create test users

Create a set of test users that reflect the users in your production environment. Creating test users enables you to verify policies work as expected before you impact real users and potentially disrupt their access to apps and resources.

Some organizations have test tenants for this purpose. However, it can be difficult to recreate all conditions and apps in a test tenant to fully test the outcome of a policy.

Create a test plan

The test plan is important to have a comparison between the expected results and the actual results. You should always have an expectation before testing something. The following table outlines example test cases. Adjust the scenarios and expected results based on how your CA policies are configured.

Policy	Scenario	Expected Result
Require MFA when not at work	Authorized user signs into app while on a trusted location / work	User is not prompted to MFA
Require MFA when not at work	Authorized user signs into app while not on a trusted location / work	User is prompted to MFA and can sign in successfully
Require MFA (for admin)	Global Admin signs into app	Admin is prompted to MFA
Risky sign-ins	User signs into app using an unapproved browser	Admin is prompted to MFA
Device management	Authorized user attempts to sign in from an authorized device	Access granted
Device management	Authorized user attempts to sign in from an unauthorized device	Access blocked
Password change for risky users	Authorized user attempts to sign in with compromised credentials (high risk sign-in)	User is prompted to change password or access is blocked based on your policy

License requirements

Using this feature requires an Azure AD Premium P1 license. Customers with Microsoft 365 Business Premium licenses also have access to Conditional Access features.

Implement Conditional Access Policies, Controls, and Assignments

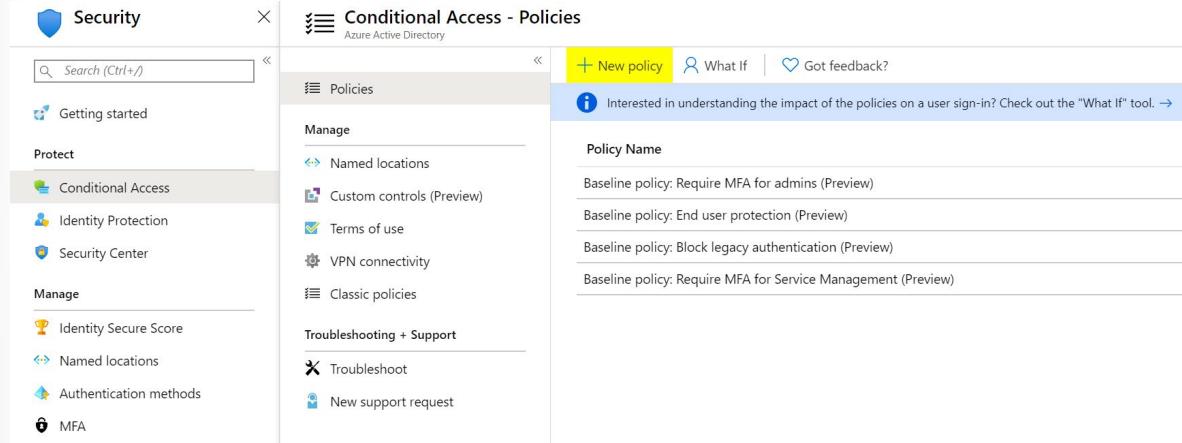
Visual Studio App Center supports Azure AD Conditional Access, an advanced feature of Azure AD that enables you to specify detailed policies that control who can access your resources. Using Conditional Access, you can protect your applications by limiting users' access based on things like group, device type, location, and role.

Setting up Conditional Access

This is an abbreviated guide to setting up Conditional Access. Full documentation is available at <https://docs.microsoft.com/azure/active-directory/conditional-access/overview>.

In the Azure portal, open your Active Directory tenant, then open the **Security** settings, and click on **Conditional Access**.

In **Conditional Access** settings, click **New policy** to create a policy.



In **New policy** settings, click on **Cloud apps or actions** and select **Visual Studio App Center** as the target of the policy. Then select the other conditions that you want to apply, enable the policy, and click **Create** to save it.

The screenshot shows the Microsoft Azure Conditional Access - Policies New blade. On the left, there's a sidebar with various icons. The main area has tabs for 'Info' and 'Cloud apps or actions'. The 'Cloud apps or actions' tab is selected, showing a section titled 'Select what this policy applies to' with 'Cloud apps' selected. It includes options for 'Include' and 'Exclude', and radio buttons for 'None', 'All cloud apps', and 'Select apps', with 'Select apps' selected. A list of apps is shown, with 'Visual Studio App Center' selected. At the bottom, there are 'Create' and 'Done' buttons.

Sign-in risk-based Conditional Access

Most users have a normal behavior that can be tracked. When they fall outside of this norm, it could be risky to allow them to just sign in. You may want to block that user or maybe just ask them to perform multifactor authentication to prove that they are really who they say they are.

A sign-in risk represents the probability that a given authentication request isn't authorized by the identity owner. Organizations with Azure AD Premium P2 licenses can create Conditional Access policies incorporating Azure AD Identity Protection sign-in risk detections.

This policy can be assigned either through Conditional Access itself or through Azure AD Identity Protection. Organizations should choose one of two options to enable a sign-in risk-based Conditional Access policy requiring a secure password change.

User risk-based Conditional Access

Microsoft works with researchers, law enforcement, various security teams at Microsoft, and other trusted sources to find leaked username and password pairs. Organizations with Azure AD Premium P2 licenses can create Conditional Access policies incorporating Azure AD Identity Protection user risk detections.

Like sign-in risk-based Conditional Access, this policy can be assigned either through Conditional Access itself or through Azure AD Identity Protection.

Securing security info registration

Securing when and how users register for Azure AD Multi-Factor Authentication and self-service password reset is now possible with user actions in Conditional Access policy. This preview feature is available to organizations that have enabled the combined registration preview. This functionality may be enabled in organizations where they want to use conditions like trusted network location to restrict access to register for Azure AD Multi-Factor Authentication and self-service password reset (SSPR).

Create a policy to require registration from a trusted location

The following policy applies to all selected users who attempt to register using the combined registration experience, and it blocks access unless they are connecting from a location marked as a trusted network.

1. In the **Azure portal**, browse to **Azure Active Directory - Security - Conditional Access**.
2. Select **New policy**.
3. In **Name**, Enter a Name for this policy. For example, **Combined Security Info Registration on Trusted Networks**.
4. Under **Assignments**, select **Users and groups**, and select the users and groups you want this policy to apply to.

Warning - Users must be enabled for the combined registration.

1. Under **Exclude**, select **Users and groups** and choose your organization's emergency access or break-glass accounts.
2. Select **Done**.
3. Under **Cloud apps or actions**, select **User actions**, check **Register security information**.
4. Under **Conditions - Locations**.
 1. Configure **Yes**.
 2. Include **Any location**.
 3. Exclude **All trusted locations**.
 4. Select **Done** on the **Locations** blade.
 5. Select **Done** on the **Conditions** blade.
6. Under **Conditions - Client apps (Preview)**, set **Configure** to **Yes**, and select **Done**.

8. Under **Access controls - Grant**.

1. Select **Block access**.

2. Then click **Select**.

9. Set **Enable policy** to **On**.

10. Then select **Save**.

At step 6 in this policy, organizations have choices they can make. The policy above requires registration from a trusted network location. Organizations can choose to utilize any available conditions in place of **Locations**. Remember that this policy is a block policy, so anything included is blocked and anything that does not match the include is allowed.

Some may choose to use device state instead of location in step 6 above:

11. Under **Conditions - Device state (Preview)**.

12. Configure **Yes**.

13. Include **All device state**.

14. Exclude **Device Hybrid Azure AD joined** and/or **Device marked as compliant**.

15. Select **Done** on the **Locations** blade.

16. Select **Done** on the **Conditions** blade.

Warning - If you use device state as a condition in your policy, this may impact guest users in the directory. Report-only mode can help determine the impact of policy decisions. Note that report-only mode is not applicable for Conditional Access policies with "User Actions" scope.

Block access by location

With the location condition in Conditional Access, you can control access to your cloud apps based on the network location of a user. The location condition is commonly used to block access from countries/regions where your organization knows traffic should not come from.

Define locations

1. Sign in to the **Azure portal** as a Global Administrator, Security Administrator, or Conditional Access Administrator.

2. Browse to **Azure Active Directory - Security - Conditional Access - Named locations**.

3. Choose **New location**.

4. Give your location a name.

5. Choose **IP ranges** if you know the specific externally accessible IPv4 address ranges that make up that location or **Countries/Regions**.

1. Provide the **IP ranges** or select the **Countries/Regions** for the location you are specifying.

- If you choose Countries/Regions, you can optionally choose to include unknown areas.

6. Choose **Save**.

Create a Conditional Access policy

1. Sign in to the **Azure portal** as a Global Administrator, Security Administrator, or Conditional Access Administrator.
2. Browse to **Azure Active Directory - Security - Conditional Access**.
3. Select **New policy**.
4. Give your policy a name. We recommend that organizations create a meaningful standard for the names of their policies.
5. Under **Assignments**, select **Users and groups**.
 1. Under **Include**, select **All users**.
 2. Under **Exclude**, select **Users and groups** and choose your organization's emergency access or break-glass accounts.
 3. Select **Done**.
6. Under **Cloud apps or actions - Include**, and select **All cloud apps**.
7. Under **Conditions - Location**.
 1. Set **Configure** to **Yes**.
 2. Under **Include**, select **Selected locations**.
 3. Select the blocked location you created for your organization.
 4. Click **Select**.
8. Under **Access controls** - select **Block Access**, and select **Select**.
9. Confirm your settings and set **Enable policy** to **On**.
10. Select **Create** to create Conditional Access Policy.

Require compliant devices

Organizations that have deployed Microsoft Intune can use the information returned from their devices to identify devices that meet compliance requirements, such as:

- Requiring a PIN to unlock.
- Requiring device encryption.
- Requiring a minimum or maximum operating system version.
- Requiring a device is not jailbroken or rooted.

This policy compliance information is forwarded to Azure AD where Conditional Access can make decisions to grant or block access to resources.

Create a Conditional Access policy

The following steps will help create a Conditional Access policy to require devices accessing resources be marked as compliant with your organization's Intune compliance policies.

1. Sign in to the **Azure portal** as a Global Administrator, Security Administrator, or Conditional Access Administrator.
2. Browse to **Azure Active Directory - Security - Conditional Access**.

3. Select **New policy**.
4. Give your policy a name. We recommend that organizations create a meaningful standard for the names of their policies.
5. Under **Assignments**, select **Users and groups**.
 1. Under **Include**, select **All users**.
 2. Under **Exclude**, select **Users and groups** and choose your organization's emergency access or break-glass accounts.
 3. Select **Done**.
6. Under **Cloud apps or actions - Include**, select **All cloud apps**.
 1. If you must exclude specific applications from your policy, you can choose them from the **Exclude** tab under **Select excluded cloud apps** and choose **Select**.
 2. Select **Done**.
7. Under **Conditions - Client apps (Preview) - Select the client apps this policy will apply to**, leave all defaults selected and select **Done**.
8. Under **Access controls - Grant**, select **Require device to be marked as compliant**.
 1. Select **Select**.
 9. Confirm your settings and set **Enable policy** to **On**.
10. Select **Create** to create to enable your policy.

Note - You can enroll your new devices to Intune even if you select Require device to be marked as compliant for All users and All cloud apps using the steps above. Require device to be marked as compliant control does not block Intune enrollment.

Known behavior

On Windows 7, iOS, Android, macOS, and some third-party web browsers, Azure AD identifies the device using a client certificate that is provisioned when the device is registered with Azure AD. When a user first signs in through the browser, the user is prompted to select the certificate. The end user must select this certificate before they can continue to use the browser.

Block access

For organizations with a conservative cloud migration approach, the block all policy is an option that can be used.

Warning - Misconfiguration of a block policy can lead to organizations being locked out of the Azure portal.

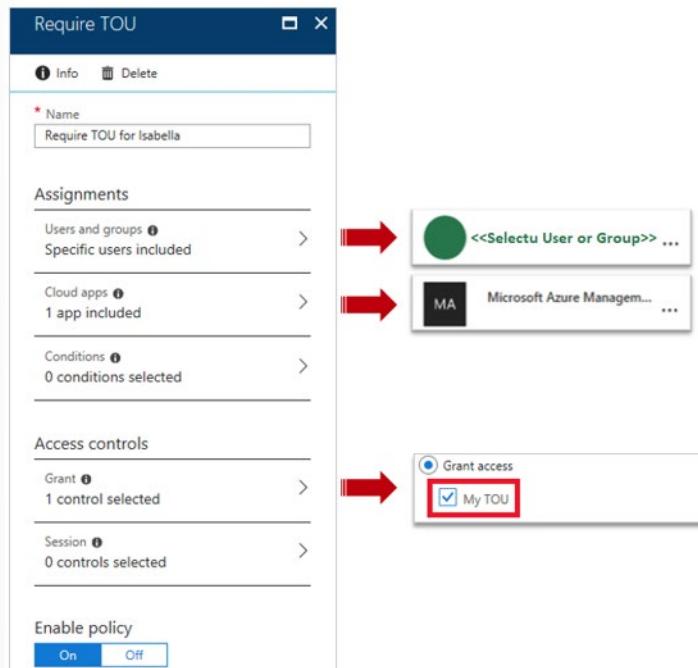
Policies like these can have unintended side effects. Proper testing and validation are vital before enabling. When making changes, administrators should utilize tools such as Conditional Access report-only mode and the What If tool in Conditional Access.

User exclusions

Conditional Access policies are powerful tools. We recommend excluding the following accounts from your policy:

- **Emergency access or break-glass** accounts to prevent tenant-wide account lockout. In the unlikely scenario that all administrators are locked out of your tenant, your emergency-access administrative account can be used to log in to the tenant and take steps to recover access.
- **Service accounts and service principals**, such as the Azure AD Connect Sync Account. Service accounts are non-interactive accounts that are not tied to any particular user. They are normally used by back-end services allowing programmatic access to applications, but they are also used to sign in to systems for administrative purposes. Service accounts like these should be excluded since MFA can't be completed programmatically. Calls made by service principals are not blocked by Conditional Access.
- If your organization has these accounts in use in scripts or code, consider replacing them with managed identities. As a temporary workaround, you can exclude these specific accounts from the baseline policy.

Conditional Access Terms of Use (TOU)



The linking of consent (accept terms before access) and conditional access is getting more and more traction. Organizations get the ability to enforce a user to consent to the terms of use. Additionally, organizations can expire the consent given or change the terms of use, and request the user attest again.

Before accessing certain cloud apps in your environment, you might want to get consent from users in form of accepting your terms of use (ToU). Azure Active Directory (Azure AD) Conditional Access provides you with:

- A simple method to configure ToU
- The option to require accepting your terms of use through a Conditional Access policy

Test and Troubleshoot Conditional Access Policies

The Conditional Access framework provides you with great configuration flexibility. However, great flexibility also means that you should carefully review each configuration policy before releasing it to avoid undesirable results. In this context, you should pay special attention to assignments affecting complete sets such as **all users / groups / cloud apps**.

Organizations should avoid the following configurations:

For all users, all cloud apps:

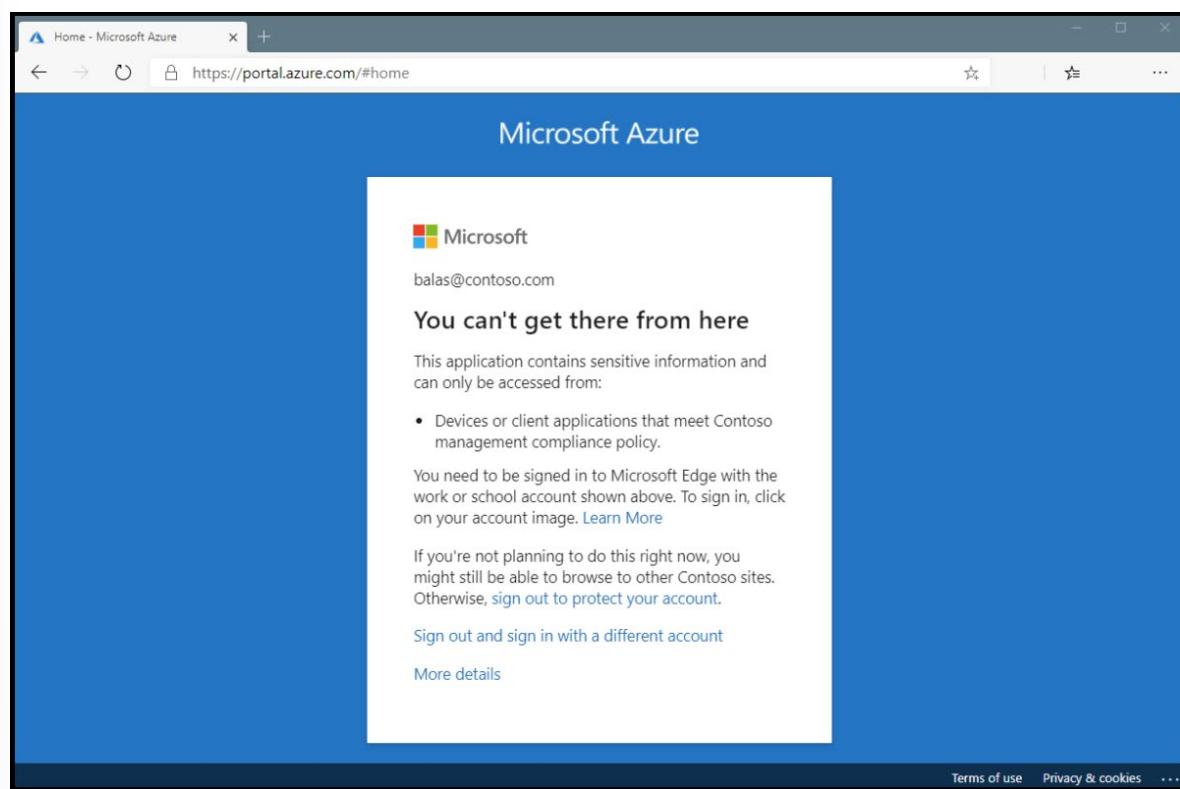
- **Block access** - This configuration blocks your entire organization.
- **Require device to be marked as compliant** - For users who have not yet enrolled their devices, this policy blocks all access including access to the Intune portal. If you are an administrator without an enrolled device, this policy blocks you from getting back into the Azure portal to change the policy.
- **Require Hybrid Azure AD domain joined device** - This access-blocking policy also has the potential to block access for all users in your organization if they don't have a hybrid Azure AD joined device.
- **Require app protection policy** - This access-blocking policy also has the potential to block access for all users in your organization if you don't have an Intune policy. If you are an administrator without a client application that has an Intune app protection policy, this policy blocks you from getting back into portals such as Intune and Azure.

For all users, all cloud apps, all device platforms:

- **Block access** - This configuration blocks your entire organization.

Conditional Access sign-in interrupt

The first way is to review the error message that appears. For problems signing in when using a web browser, the error page itself has detailed information. This information alone may describe what the problem is and suggest a solution.

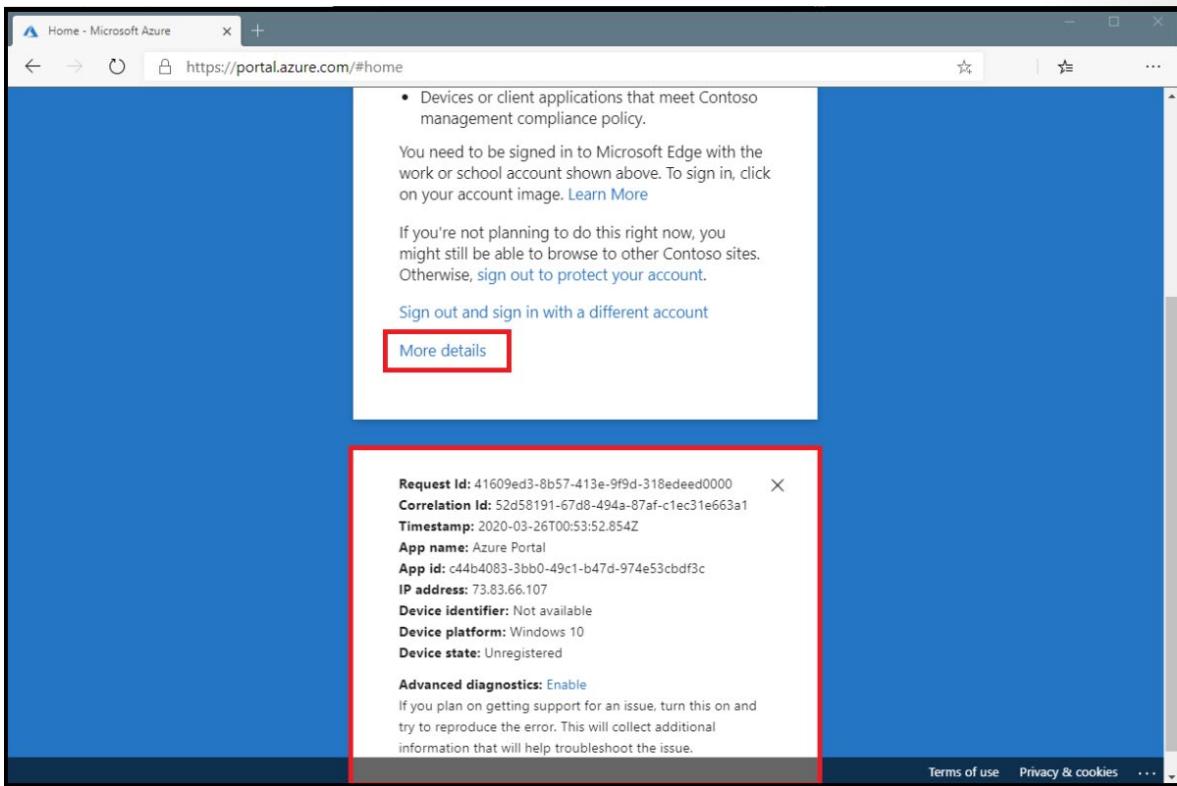


In the above error, the message states that the application can only be accessed from devices or client applications that meet the company's mobile device management policy. In this case, the application and device do not meet that policy.

Azure Active Directory sign-in events

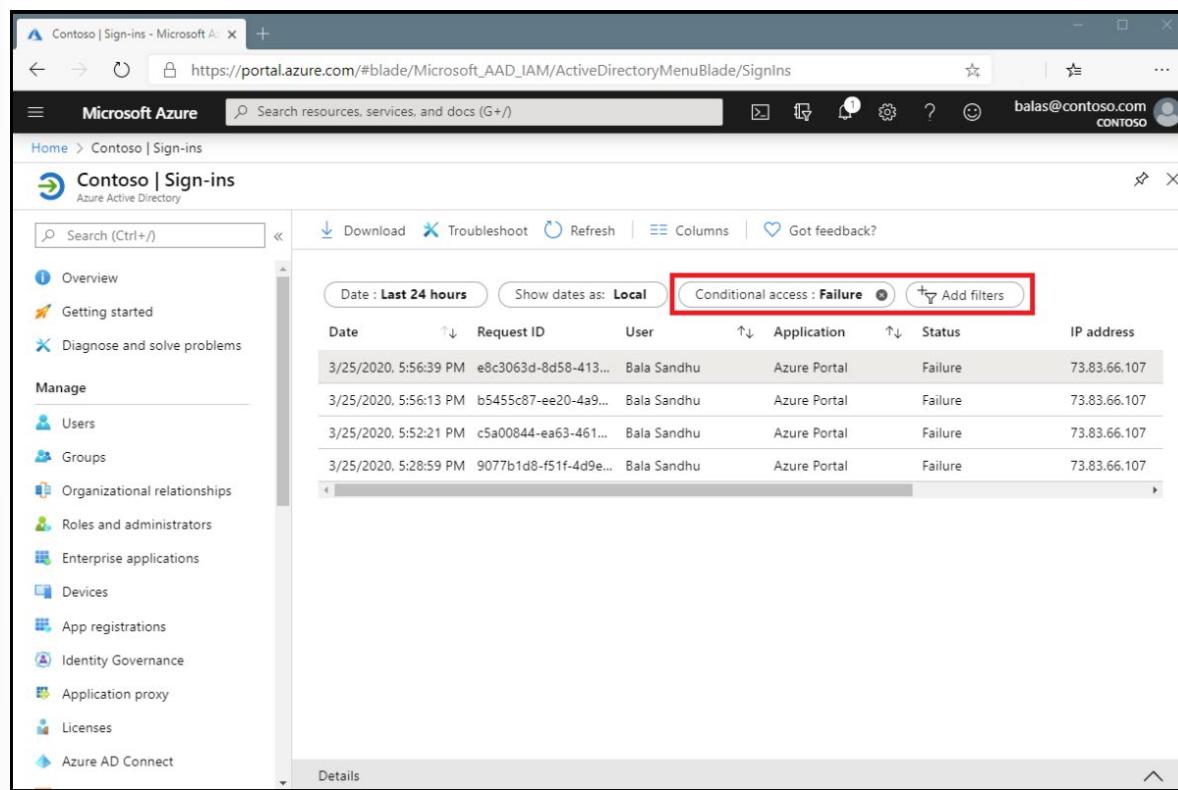
The second method to get detailed information about the sign-in interruption is to review the Azure AD sign-in events to see which Conditional Access policy or policies were applied and why.

Find more information about the problem by clicking **More Details** in the initial error page. Clicking **More Details** will reveal troubleshooting information that's helpful when searching the Azure AD sign-in events for the specific failure event the user saw or when opening a support incident with Microsoft.



To find out which Conditional Access policy or policies applied and why, do the following:

1. Sign in to the Azure portal as a Global Administrator, Security Administrator, or Global Reader.
2. Browse to **Azure Active Directory - Sign-ins**.
3. Find the event for the sign-in to review. Add or remove filters and columns to filter out unnecessary information.
 1. Add filters to narrow the scope:
 1. Correlation ID when you have a specific event to investigate.
 2. Conditional access to see policy failure and success. Scope your filter to show only failures to limit results.
 3. Username to see information related to specific users.
 4. Date scoped to the time frame in question.



The screenshot shows the Microsoft Azure Sign-ins blade for the Contoso tenant. The left sidebar lists various management options like Users, Groups, and Enterprise applications. The main area displays a table of sign-in events. The table has columns for Date, Request ID, User, Application, Status, and IP address. Four rows of data are visible, all showing a failure status. At the top of the table area, there are filters: 'Date : Last 24 hours', 'Show dates as: Local', and a button labeled 'Conditional access : Failure' which is highlighted with a red box. Below the table is a 'Details' link.

Date	Request ID	User	Application	Status	IP address
3/25/2020, 5:56:39 PM	e8c3063d-8d58-413...	Bala Sandhu	Azure Portal	Failure	73.83.66.107
3/25/2020, 5:56:13 PM	b5455c87-ee20-4a9...	Bala Sandhu	Azure Portal	Failure	73.83.66.107
3/25/2020, 5:52:21 PM	c5a00844-ea63-461...	Bala Sandhu	Azure Portal	Failure	73.83.66.107
3/25/2020, 5:28:59 PM	9077b1d8-f51f-4d9e...	Bala Sandhu	Azure Portal	Failure	73.83.66.107

4. Once the sign-in event that corresponds to the user's sign-in failure has been found select the **Conditional Access** tab, the tab will show the specific policy or policies that resulted in the sign-in interruption.
 1. Information in the **Troubleshooting and support** tab may provide a clear reason as to why a sign-in failed, such as a device that did not meet compliance requirements.
 2. To investigate further, drill down into the configuration of the policies by clicking on the Policy Name. Clicking the Policy Name will show the policy configuration user interface for the selected policy for review and editing.
 3. The client user and device details that were used for the Conditional Access policy assessment are also available in the **Basic Info**, **Location**, **Device Info**, **Authentication Details**, and **Additional Details** tabs of the sign-in event.

Policy details

Selecting the ellipsis on the right side of the policy in a sign-in event brings up policy details. This gives administrators additional information about why a policy was successfully applied or not.

Basic info	Location	Device info	Authentication Details	Conditional Access	Report-only	Additional Details
Policy Name	Grant Controls		Session Controls	Result		
Common Policy - Require MFA for administrators	require multi-factor authentication			Failure		
Common Policy - Require MFA for Azure manager	require multi-factor authentication			Not Applied		
Common Policy - Block legacy authentication	block			Not Applied		
Common Policy - Require trusted location for MFA				Disabled		
Common Policy - Require compliant devices	require compliant device			Not Applied		
MFA Pilot	require multi-factor authentication			Not Applied		
Approved Client App - Exchange Online	require approved app			Not Applied		
Approved Client App - Exchange Online - 02	require approved app			Not Applied		
Conditional-Access-Test-Policy	require multi-factor authentication			Not Applied		
Camp2020	require multi-factor authentication			Not Applied		

A sign-in can also be interrupted (e.g. blocked, MFA challenged) because of a user risk policy or sign-in risk policy. Currently, this tab only lists Conditional Access policies.

Policy details (Preview)

Policy: Common Policy - Require MFA for administrators
Policy state: Enabled
Result: Failure

Assignments

User Baka Sandhu Satisfied

Application frontend-app-camp2020 Satisfied

Conditions

Sign-in risk None Not configured

Device Platform Windows 10 Not configured

Location Renton, US
73.254.183.78 Not configured

Client app Browser Not configured

Device state None Not configured

Access controls

Grant Controls Not satisfied

The left side provides details collected at sign-in, and the right side provides details of whether those details satisfy the requirements of the applied Conditional Access policies. Conditional Access policies only apply when all conditions are satisfied or not configured.

If the information in the event isn't enough to understand the sign-in results or adjust the policy to get desired results, then a support incident may be opened. Navigate to that sign-in event's **Troubleshooting and support** tab and select **Create a new support request**.

The screenshot shows the Microsoft Cloud App Security interface. On the left, there's a summary of a sign-in event: Status is Failure, Sign-in error code is 53000, and the Failure reason is "Conditional Access policy requires a compliant device, and the device is not compliant. Have the user enroll their device with an approved MDM provider like Intune." Below that, Additional Details show "MFA completed in Azure AD". On the right, under the "Troubleshooting and support" tab, there's a section for creating a support request. It includes steps: 1. Create a new support request (with a link), 2. Set Issue type to 'Technical' and Service to 'Azure Active Directory', 3. Select 'Problem Type' and 'Category', and 4. Paste Request ID and Date (UTC). The Request ID is e8c3063d-8d58-4130-bd83-21f173d60100 and the Date (UTC) is 2020-03-26T00:56:39.063Z.

When submitting the incident, provide the request ID and time and date from the sign-in event in the incident submission details. This information will allow Microsoft support to find the event you're concerned about.

Implement Application Controls and Application Protection

Conditional Access App Control enables user app access and sessions to be monitored and controlled in real time based on access and session policies. Access and session policies are used within the Cloud App Security portal to further refine filters and set actions to be taken on a user.

Conditional Access App Control

The screenshot shows the Microsoft Cloud App Security interface for creating a new Conditional Access policy. The policy name is "Conditional Access Documentation". Under the "Session" tab, there's a section for "Session controls" which says "Session controls enable limited experiences within a cloud app. Select the session usage requirements." Below this, there are several checkboxes: "Use app enforced restrictions", "Use Conditional Access App Control" (which is highlighted with a green border), "Sign-in frequency", and "Persistent browser session".

Conditional Access App Control uses a reverse proxy architecture and is uniquely integrated with Azure AD Conditional Access. Azure AD 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 Microsoft Cloud App Security where you can protect data with Conditional Access App Control by applying access and session controls.

With the access and session policies, you can:

- **Prevent data exfiltration:** You can block the download, cut, copy, and print of sensitive documents on, for example, unmanaged devices.
- **Protect on download:** Instead of blocking the download of sensitive documents, you can require documents to be labeled and protected with Azure Information Protection. This action ensures the document is protected and user access is restricted in a potentially risky session.
- **Prevent upload of unlabeled files:** Before a sensitive file is uploaded, distributed, and used by others, it's important to make sure that the file has the right label and protection. You can ensure that unlabeled files with sensitive content are blocked from being uploaded until the user classifies the content.
- **Monitor user sessions for compliance:** Risky users are monitored when they sign into apps and their actions are logged from within the session. You can investigate and analyze user behavior to understand where, and under what conditions, session policies should be applied in the future.
- **Block access:** You can granularly block access for specific apps and users depending on several risk factors. For example, you can block them if they are using client certificates as a form of device management.
- **Block custom activities:** Some apps have unique scenarios that carry risk, for example, sending messages with sensitive content in apps like Microsoft Teams or Slack. In these kinds of scenarios, you can scan messages for sensitive content and block them in real time.

How to: Require app protection policy and an approved client app for cloud app access with Conditional Access

People regularly use their mobile devices for both personal and work tasks. While making sure staff can be productive, organizations also want to prevent data loss from potentially unsecure applications. With Conditional Access, organizations can restrict access to approved (modern authentication-capable) client apps.

This section presents two scenarios to configure Conditional Access policies for resources like Microsoft 365, Exchange Online, and SharePoint Online.

Note - In order to require approved client apps for iOS and Android devices, these devices must first register in Azure AD.

Scenario 1: Microsoft 365 apps require an approved client app

In this scenario, Contoso has decided that users using mobile devices can access all Microsoft 365 services as long as they use approved client apps, like Outlook mobile, OneDrive, and Microsoft Teams. All of their users already sign in with Azure AD credentials and have licenses assigned to them that include Azure AD Premium P1 or P2 and Microsoft Intune.

Organizations must complete the following three steps in order to require the use of an approved client app on mobile devices.

Step 1: Policy for Android and iOS based modern authentication clients requiring the use of an approved client application when accessing Exchange Online.

1. Sign in to the **Azure portal** as a Global Administrator, Security Administrator, or Conditional Access Administrator.
2. Browse to **Azure Active Directory - Security - Conditional Access**.
3. Select **New policy**.
4. Give your policy a name. We recommend that organizations create a meaningful standard for the names of their policies.
5. Under **Assignments**, select **Users and groups**.
 1. Under **Include**, select **All users** or the specific **Users and groups** you wish to apply this policy to.
 2. Select **Done**.
6. Under **Cloud apps or actions - Include**, select **Office 365**.
7. Under **Conditions**, select **Device platforms**.
 1. Set **Configure** to **Yes**.
 2. Include **Android** and **iOS**.
8. Under **Conditions**, select **Client apps (preview)**.
 1. Set **Configure** to **Yes**.
 2. Select **Mobile apps and desktop clients** and **Modern authentication clients**.
9. Under **Access controls - Grant**, select **Grant access**, **Require approved client app**, and select **Select**.
10. Confirm your settings and set **Enable policy** to **On**.
11. Select **Create** to create and enable your policy.

Step 2: Configure an Azure AD Conditional Access policy for Exchange Online with ActiveSync (EAS).

1. Browse to **Azure Active Directory - Security - Conditional Access**.
2. Select **New policy**.
3. Give your policy a name. We recommend that organizations create a meaningful standard for the names of their policies.
4. Under **Assignments**, select **Users and groups**.
 1. Under **Include**, select **All users** or the specific **Users and groups** you wish to apply this policy to.
 2. Select **Done**.
5. Under **Cloud apps or actions - Include**, select **Office 365 Exchange Online**.
6. Under **Conditions**:
 1. **Client apps (preview)**:
 1. Set **Configure** to **Yes**.
 2. Select **Mobile apps and desktop clients** and **Exchange ActiveSync clients**.
7. Under **Access controls - Grant**, select **Grant access**, **Require approved client app**, and select **Select**.

8. Confirm your settings and set **Enable policy** to **On**.
9. Select **Create** to create and enable your policy.

Step 3: Configure Intune app protection policy for iOS and Android client applications.

Review the article **How to create and assign app protection policies⁸** for steps to create app protection policies for Android and iOS.

Scenario 2: Exchange Online and SharePoint Online require an approved client app

In this scenario, Contoso has decided that users may only access email and SharePoint data on mobile devices as long as they use an approved client app like Outlook mobile. All of their users already sign in with Azure AD credentials and have licenses assigned to them that include Azure AD Premium P1 or P2 and Microsoft Intune.

Organizations must complete the following three steps in order to require the use of an approved client app on mobile devices and Exchange ActiveSync clients.

Step 1: Policy for Android and iOS based modern authentication clients requiring the use of an approved client application when accessing Exchange Online and SharePoint Online.

1. Sign in to the **Azure portal** as a Global Administrator, Security Administrator, or Conditional Access Administrator.
2. Browse to **Azure Active Directory - Security - Conditional Access**.
3. Select **New policy**.
4. Give your policy a name. We recommend that organizations create a meaningful standard for the names of their policies.
5. Under **Assignments**, select **Users and groups**.
 1. Under **Include**, select **All users** or the specific **Users and groups** you wish to apply this policy to.
 2. Select **Done**.
6. Under **Cloud apps or actions - Include**, select **Office 365 Exchange Online** and **Office 365 SharePoint Online**.
7. Under **Conditions**, select **Device platforms**.
 1. Set **Configure** to **Yes**.
 2. Include **Android** and **iOS**.
8. Under **Conditions**, select **Client apps (preview)**.
 1. Set **Configure** to **Yes**.
 2. Select **Mobile apps and desktop clients** and **Modern authentication clients**.
9. Under **Access controls - Grant**, select **Grant access**, **Require approved client app**, and select **Select**.
10. Confirm your settings and set **Enable policy** to **On**.
11. Select **Create** to create and enable your policy.

⁸ <https://docs.microsoft.com/intune/apps/app-protection-policies>

Step 2: Policy for Exchange ActiveSync clients requiring the use of an approved client app.

1. Browse to **Azure Active Directory - Security - Conditional Access**.
2. Select **New policy**.
3. Give your policy a name. We recommend that organizations create a meaningful standard for the names of their policies.
4. Under **Assignments**, select **Users and groups**.
 1. Under **Include**, select **All users** or the specific **Users and groups** you wish to apply this policy to.
 2. Select **Done**.
5. Under **Cloud apps or actions - Include**, select **Office 365 Exchange Online**.
6. Under **Conditions**:
 1. **Client apps (preview)**:
 1. Set **Configure** to **Yes**.
 2. Select **Mobile apps and desktop clients** and **Exchange ActiveSync clients**.
7. Under **Access controls - Grant**, select **Grant access**, **Require approved client app**, and select **Select**.
8. Confirm your settings and set **Enable policy** to **On**.
9. Select **Create** to create and enable your policy.

Step 3: Configure Intune app protection policy for iOS and Android client applications.

Review the article **How to create and assign app protection policies**⁹ for steps to create app protection policies for Android and iOS.

App protection policies overview

App protection policies (APP) are rules that ensure an organization's data remains safe or contained in a managed app. A policy can be a rule that is enforced when the user attempts to access or move "corporate" data, or a set of actions that are prohibited or monitored when the user is inside the app. A managed app has app protection policies applied to it, and it can be managed by Intune.

Mobile Application Management (MAM) app protection policies allow you to manage and protect your organization's data within an application. With **MAM without enrollment** (MAM-WE), a work or school-related app that contains sensitive data can be managed on almost any device, including personal devices in **bring-your-own-device** (BYOD) scenarios. Many productivity apps, such as the Microsoft Office apps, can be managed by Intune MAM.

How you can protect app data

Your employees use mobile devices for both personal and work tasks. While making sure your employees can be productive, you want to prevent data loss—intentional and unintentional. You'll also want to protect company data that is accessed from devices that you do not manage.

You can use Intune app protection policies **independent of any mobile-device management (MDM) solution**. This independence helps you protect your company's data with or without enrolling devices in a device management solution. By implementing **app-level policies**, you can restrict access to company resources and keep data within the purview of your IT department.

⁹ <https://docs.microsoft.com/intune/apps/app-protection-policies>

App protection policies on devices

App protection policies can be configured for apps that run on devices that are:

- **Enrolled in Microsoft Intune:** These devices are typically corporate owned.
- **Enrolled in a third-party MDM solution:** These devices are typically corporate owned.
Note - Mobile app management policies should not be used with third-party mobile app management or secure container solutions.
- **Not enrolled in any mobile device management solution:** These devices are typically employee-owned devices that aren't managed or enrolled in Intune or other MDM solutions.
Important - You can create mobile app management policies for Office mobile apps that connect to Microsoft 365 services. You can also protect access to Exchange on-premises mailboxes by creating Intune app protection policies for Outlook for iOS/iPadOS and Android enabled with hybrid Modern Authentication. Before using this feature, make sure you meet the Outlook for iOS/iPadOS and Android requirements. App protection policies are not supported for other apps that connect to on-premises Exchange or SharePoint services.

Benefits of using app protection policies

The important benefits of using app protection policies are the following:

- **Protecting your company data at the app level.** Because mobile app management doesn't require device management, you can protect company data on both managed and unmanaged devices. The management is centered on the user identity, which removes the requirement for device management.
- **End-user productivity isn't affected and policies don't apply when using the app in a personal context.** The policies are applied only in a work context, which gives you the ability to protect company data without touching personal data.
- **App protection policies ensure that the app-layer protections are in place.** For example, you can:
 - Require a PIN to open an app in a work context.
 - Control the sharing of data between apps.
 - Prevent the saving of company app data to a personal storage location.
- **MDM, in addition to MAM, ensures that the device is protected.** For example, you can require a PIN to access the device, or you can deploy managed apps to the device. You can also deploy apps to devices through your MDM solution to give you more control over app management.

There are additional benefits to using MDM with app protection policies, and companies can use app protection policies with and without MDM at the same time. For example, consider an employee who uses a phone issued by the company, as well as their personal tablet. The company phone is enrolled in MDM and protected by app protection policies, while the personal device is protected by app protection policies only.

If you apply a MAM policy to the user without setting the device state, the user will get the MAM policy on both the BYOD device and the Intune-managed device. You can also apply a MAM policy based on the managed state. So when you create an app protection policy, next to **Target to all app types**, you'd select **No**. Then do any of the following:

- Apply a less strict MAM policy to Intune managed devices, and apply a more restrictive MAM policy to non MDM-enrolled devices.

- Apply a MAM policy to unenrolled devices only.

Implement Sessions Management

In complex deployments, organizations might have a need to restrict authentication sessions. Some scenarios might include:

- Resource access from an unmanaged or shared device.
- Access to sensitive information from an external network.
- High impact users.
- Critical business applications.

Conditional Access controls allow you to create policies that target specific use cases within your organization without affecting all users.

Before diving into details on how to configure the policy, let's examine the default configuration.

User sign-in frequency

Sign-in frequency defines the time period before a user is asked to sign in again when attempting to access a resource.

The Azure AD default configuration for user sign-in frequency is a rolling window of 90 days. Asking users for credentials often seems like a sensible thing to do, but it can backfire: Users who are trained to enter their credentials without thinking can unintentionally supply them to a malicious credential prompt.

It might sound alarming to not ask for a user to sign back in; in reality any violation of IT policies will revoke the session. Some examples include a password change, an incompliant device, or an account disable. You can also explicitly revoke users' sessions using PowerShell. The Azure AD default configuration comes down to "don't ask users to provide their credentials if the security posture of their sessions has not changed."

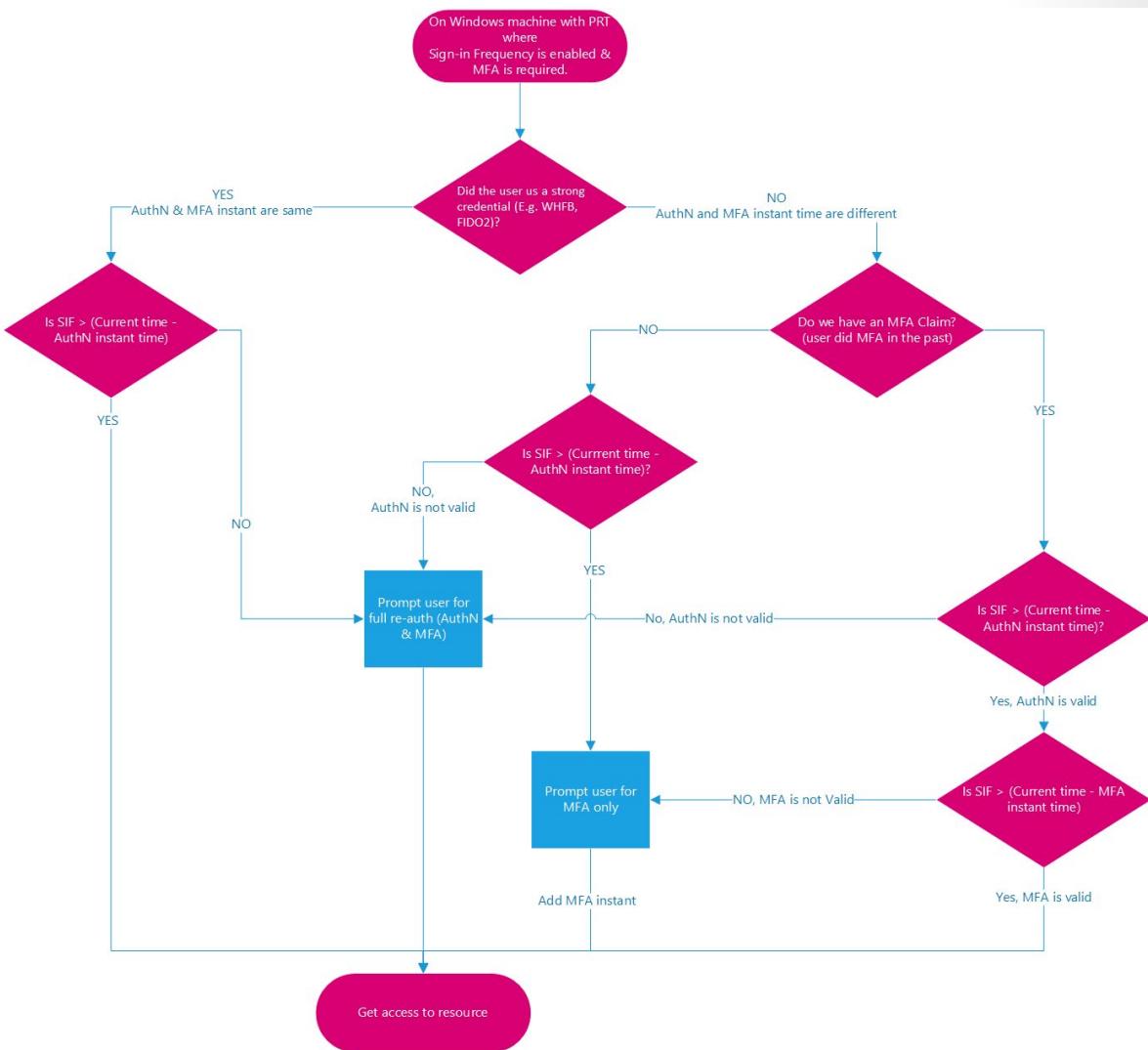
The sign-in frequency setting works with apps that have implemented OAuth2 or OIDC protocols according to the standards. Most Microsoft native apps for Windows, Mac, and mobile, including the following web applications, comply with the setting.

- Word, Excel, PowerPoint Online
- OneNote Online
- Office.com
- Microsoft 365 Admin portal
- Exchange Online
- SharePoint and OneDrive
- Teams web client
- Dynamics CRM Online
- Azure portal

The sign-in frequency setting works with SAML applications as well, as long as they do not drop their own cookies and are redirected back to Azure AD for authentication on a regular basis.

User sign-in frequency and multifactor authentication

Sign-in frequency previously applied only to the first factor authentication on devices that were Azure AD joined, Hybrid Azure AD joined, and Azure AD registered. There was no easy way for our customers to re-enforce multifactor authentication (MFA) on those devices. Based on customer feedback, sign-in frequency will apply for MFA as well.



User sign-in frequency and device identities

If you have Azure AD joined, hybrid Azure AD joined, or Azure AD registered devices, when a user unlocks their device or signs in interactively, this event will satisfy the sign-in frequency policy as well. In the following two examples user sign-in frequency is set to one hour:

Example 1:

- At 00:00, a user signs in to their Windows 10 Azure AD joined device and starts work on a document stored on SharePoint Online.
- The user continues working on the same document on their device for an hour.

- At 01:00, the user is prompted to sign in again based on the sign-in frequency requirement in the Conditional Access policy configured by their administrator.

Example 2:

- At 00:00, a user signs in to their Windows 10 Azure AD joined device and starts work on a document stored on SharePoint Online.
- At 00:30, the user gets up and takes a break, locking their device.
- At 00:45, the user returns from their break and unlocks the device.
- At 01:45, the user is prompted to sign in again based on the sign-in frequency requirement in the Conditional Access policy configured by their administrator since the last sign-in happened at 00:45.

Persistence of browsing sessions

A persistent browser session allows users to remain signed in after closing and reopening their browser window. The Azure AD default for browser session persistence allows users on personal devices to choose whether to persist the session by showing a "Stay signed in?" prompt after successful authentication.

Validation

Use the What-If tool to simulate a login from the user to the target application and other conditions based on how you configured your policy. The authentication session management controls show up in the result of the tool.

The screenshot shows the 'What If' tool interface for testing conditional access policies. It includes fields for User (bala@contoso.com), Cloud apps (selected 1 app), IP address, Country, Device platform, Client apps, Device state, Sign-in risk, and a 'What If' button. The 'Evaluation result' section shows 'Policies that will apply' and 'Policies that will not apply'. The 'SESSION CONTROLS' section, which is highlighted with a red box, contains the setting 'Persistent browser session (preview) - N... ...'.

Policy deployment

To make sure that your policy works as expected, the recommended best practice is to test it before rolling it out into production. Ideally, use a test tenant to verify whether your new policy works as intended.

Configure Smart Lockout Thresholds

Smart lockout helps lock out bad actors who try to guess your users' passwords or use brute-force methods to get in. Smart lockout can recognize sign-ins that come from valid users and treat them differently than ones of attackers and other unknown sources. Attackers get locked out, while your users continue to access their accounts and be productive.

How smart lockout works

By default, smart lockout locks the account from sign-in attempts for one minute after 10 failed attempts. The account locks again after each subsequent failed sign-in attempt, for one minute at first and then longer in subsequent attempts. To minimize the ways an attacker could work around this behavior, we don't disclose the rate at which the lockout period grows over additional unsuccessful sign-in attempts.

Smart lockout tracks the last three bad password hashes to avoid incrementing the lockout counter for the same password. If someone enters the same bad password multiple times, this behavior won't cause the account to lock out.

Federated deployments that use AD FS 2016 and AD FS 2019 can enable similar benefits using AD FS Extranet Lockout and Extranet Smart Lockout.

Smart lockout is always on, for all Azure AD customers, with these default settings that offer the right mix of security and usability. Customization of the smart lockout settings, with values specific to your organization, requires Azure AD Premium P1 or higher licenses for your users.

Using smart lockout doesn't guarantee that a genuine user is never locked out. When smart lockout locks a user account, we try our best to not lock out the genuine user. The lockout service attempts to ensure that bad actors can't gain access to a genuine user account. The following considerations apply:

- Each Azure AD data center tracks lockouts independently. A user has (`threshold_limit - datacenter_count`) number of attempts, if the user hits each data center.
- Smart lockout uses familiar location vs. unfamiliar location to differentiate between a bad actor and the genuine user. Unfamiliar and familiar locations both have separate lockout counters.

Smart lockout can be integrated with hybrid deployments that use password hash sync or pass-through authentication to protect on-premises Active Directory Domain Services (AD DS) accounts from being locked out by attackers. By setting smart lockout policies in Azure AD appropriately, attacks can be filtered out before they reach on-premises AD DS.

When using pass-through authentication, the following considerations apply:

- The Azure AD lockout threshold is **less** than the AD DS account lockout threshold. Set the values so that the AD DS account lockout threshold is at least two or three times longer than the Azure AD lockout threshold.
- The Azure AD lockout duration must be set longer than the AD DS reset account lockout counter after duration. The Azure AD duration is set in seconds, while the AD duration is set in minutes.

For example, if you want your Azure AD counter to be higher than AD DS, then Azure AD would be 120 seconds (2 minutes) while your on-premises AD is set to 1 minute (60 seconds).

Manage Azure AD Identity Protection

Introduction

Protecting a user's identity by monitoring their usage and sign-in patterns ensures a secure cloud solution. Explore how to design and implement Azure Active Directory (Azure AD) Identity Protection.

	<p>In this video, get a high-level overview of Identity Protection, a feature of Azure Active Directory. You'll learn about different types of detections, risks, and risk policies that exist in Identity Protection. The video explains the benefits of the risk policies, recent UX enhancements, powerful APIs, improved risk assessment, and overall alignment along risky users and risky sign-ins.</p>
-----------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



<https://www.microsoft.com/videoplayer/embed/RE4MxmC>

Learning objectives

In this module, you will:

- Review Identity Protection basics.
- Implement and manage a user risk policy.
- Implement and manage sign-in risk policies.
- Implement and manage multifactor authentication (MFA) registration policy.
- Monitor, investigate, and remediate elevated risky users.

Review Identity Protection Basics

Identity Protection is a tool that allows organizations to accomplish three key tasks:

- Automate the detection and remediation of identity-based risks.
- Investigate risks using data in the portal.
- Export risk detection data to third-party utilities for further analysis.

Identity Protection uses the knowledge Microsoft has gained from its position in organizations with Azure AD, the consumer space with Microsoft Accounts, and in gaming with Xbox to protect your users. Microsoft analyzes 6.5 trillion signals per day to identify and protect customers from threats.

The signals generated by and fed to Identity Protection can be further fed into tools like Conditional Access to make access decisions or fed back to a security information and event management (SIEM) tool for further investigation based on your organization's enforced policies.

Risk detection and remediation

Identity Protection identifies risks in the following classifications:

Risk detection type	Description
Anonymous IP address	Sign in from an anonymous IP address (for example: Tor browser, anonymizer VPNs)
Atypical travel	Sign in from an atypical location based on the user's recent sign-ins.
Malware-linked IP address	Sign in from a malware-linked IP address.
Unfamiliar sign-in properties	Sign in with properties we've not seen recently for the given user.
Leaked credentials	Indicates that the user's valid credentials have been leaked.
Password spray	Indicates that multiple usernames are being attacked using common passwords in a unified brute-force manner.
Azure AD threat intelligence	Microsoft internal and external threat intelligence sources have identified a known attack pattern.
New country	This detection is discovered by Microsoft Cloud App Security (MCAS).
Activity from anonymous IP address	This detection is discovered by MCAS.
Suspicious inbox forwarding	This detection is discovered by MCAS.

Permissions

Identity Protection requires users be a Security Reader, Security Operator, Security Administrator, Global Reader, or Global Administrator in order to access.

Role	Can do	Can't do
Global Administrator	Full access to Identity Protection	
Security Administrator	Full access to Identity Protection	Reset password for a user
Security Operator	View all Identity Protection reports and Overview blade Dismiss user risk, confirm safe sign-in, confirm compromise	Configure or change policies: Reset password for a user and Configure alerts
Security Reader	View all Identity Protection reports and Overview blade	Configure or change policies and Reset password for a user and Configure alerts and Give feedback on detections

Currently, the Security Operator role cannot access the Risky sign-ins report.

Conditional Access Administrators can also create policies that factor in sign-in risk as a condition.

License requirements

Using this feature requires an Azure AD Premium P2 license.

Capability	Details	Azure AD Free / Microsoft 365 Apps	Azure AD Premium P1	Azure AD Premium P2
Risk policies	User risk policy (via Identity Protection)	No	No	Yes
Risk policies	Sign-in risk policy (via Identity Protection or Conditional Access)	No	No	Yes
Security reports	Overview	No	No	Yes
Security reports	Risky users	Limited information. Only users with medium and high risk are shown. No details drawer or risk history.	Limited information. Only users with medium and high risk are shown. No details drawer or risk history.	Full access
Security reports	Risky sign-ins	Limited information. No risk detail or risk level is shown.	Limited information. No risk detail or risk level is shown.	Full access
Security reports	Risk detections	No	Limited information. No details drawer.	Full access
Notifications	Users at risk detected alerts	No	No	Yes
Notifications	Weekly digest	No	No	Yes
	MFA registration policy	No	No	Yes

Implement and Manager User Risk Policy

There are two risk policies that can be enabled in the directory:

- Sign-in risk policy: The sign-in risk policy detects suspicious actions that come along with the sign-in. It is focused on the sign-in activity itself and analyzes the probability that the sign-in may not have been performed by the user.
- User risk policy: The user risk policy detects the probability that a user account has been compromised by detecting risk events that are atypical of a user's behavior.

New risky users detected

Date	Count
08/11	1
08/18	1
08/25	3
09/01	1

New risky sign-ins detected

Type	Count
Unprotected	111
Protected	26

High risk users: 7

Medium risk users: 12

Identity Secure Score: 34 / 223

Both policies work to automate the response to risk detections in your environment and allow users to self-remediate when risk is detected.

	In this video, learn how to deploy Azure AD Identity Protection by configuring risk-based policies (user risk and sign-in risk) in your organization. You'll also learn best practices on how to gradually roll out these policies and MFA registration in your organization.
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



<https://www.microsoft.com/videoplayer/embed/RE4MxmD>

Prerequisites

If your organization wants to allow users to self-remediate when risks are detected, users must be registered for both self-service password reset and Azure AD Multi-Factor Authentication. We recommend enabling the combined security information registration experience. Allowing users to self-remediate gets them back to a productive state more quickly without requiring administrator intervention. Administrators can still see these events and investigate them after the fact.

Choosing acceptable risk levels

Organizations must decide the level of risk they are willing to accept, balancing user experience and security posture.

Microsoft recommendation is to set the user risk policy threshold to **High** and the sign-in risk policy to **Medium and above**.

Choosing a **High** threshold reduces the number of times a policy is triggered and minimizes the impact to users. However, it excludes **Low** and **Medium** risk detections from the policy, which may not block an attacker from exploiting a compromised identity. Selecting a **Low** threshold introduces additional user interrupts but increased security posture.

Exclusions

All of the policies allow for excluding users such as your emergency access or break-glass administrator accounts. Organizations may determine they need to exclude other accounts from specific policies based on the way the accounts are used. All exclusions should be reviewed regularly to see if they are still applicable.

Configured trusted network locations are used by Identity Protection in some risk detections to reduce false positives.

Monitor and remediate elevated risky users

Investigate risk

Identity Protection provides organizations with three reports they can use to investigate identity risks in their environment: **risky users**, **risky sign-ins**, and **risk detections**. Investigating events is key to better understanding and identifying any weak points in your security strategy.

All three reports allow for downloading of events in .CSV format for further analysis outside of the Azure portal. The risky users and risky sign-ins reports allow for downloading the most recent 2,500 entries, while the risk detections report allows for downloading the most recent 5,000 records.

Organizations can take advantage of the Microsoft Graph API integrations to aggregate data with other sources they may have access to as an organization.

You can find the three reports in the **Azure portal - Azure Active Directory - Security**.

Navigating the reports

Each report launches with a list of all detections for the period shown at the top of the report. Each report allows for the addition or removal of columns based on administrator preference. Administrators can choose to download the data in .CSV or .JSON format. Reports can be filtered using the filters across the top of the report.

Selecting individual entries may enable additional entries at the top of the report, such as the ability to confirm a sign-in as compromised or safe, confirm a user as compromised, or dismiss user risk.

Selecting individual entries expands a details window below the detections. The details view allows administrators to investigate and perform actions on each detection.

DATE	USER	APPLICATION	STATUS	IP ADDRESS	LOCATION	RISK STATE	REQUEST ID	CONDITIONAL ACCE...
9/10/2019, 3:26:10 PM	Alain Charon	Azure Portal	Success	131.107.174.133	Redmond, Washingt...	At risk	5e29879d-e7b6-48...	notApplied
9/10/2019, 3:02:56 P...	Alain Charon	Azure Portal	Interrupted	131.107.147.147	Redmond, Washingt...	At risk	cfd5e5c2-2478-4ee...	notApplied
9/10/2019, 9:15:42 A...	Alain Charon	Microsoft Office 36...	Interrupted	131.107.159.177	Redmond, Washingt...	At risk	29e08ac8-4e6e-4e9...	notApplied
9/9/2019, 11:01:05 PM	Alain Charon	Azure Portal	Interrupted	131.107.159.4	Redmond, Washingt...	At risk	75442753-71d3-428...	notApplied
9/9/2019, 9:48:30 PM	Alain Charon	Azure Portal	Interrupted	50.47.110.56	Redmond, Washingt...	At risk	8a7d5385-432a-48...	notApplied

Details

View user's risk report	View user's sign-ins	View user's risky sign-ins	View user's risk detections	View sign-in's risk detections	Confirm sign-in compromised	Confirm sign-in safe
Basic Info Risk info Device info MFA info Conditional Access						
Request ID	29e08ac8-xxxx-xxxx-4a65c26bab00	IP address	131.107.159.177			
Correlation ID	44ff76a8-xxxx-xxxx-41bea9d9be5d	Location	Redmond, Washington, US			
User	Alain Charon	Date	9/10/2019, 9:15:42 AM			
Username	alain.charon@contoso.com	Status	Interrupted			
User ID	bab8aed7-xxxx-xxxx-49c3688bf05e	Sign-in error code	50140			
Application	Microsoft Office 365 Portal	Failure reason	This error occurred due to 'Keep me signed in' interrupt when the user was signing-in.			
Application ID	00000006-0000-0ff1-ce00-000000000000	Client app	Browser			
Resource	Windows Azure Active Directory					
Resource ID	00000002-0000-0000-c000-000000000000					

Risky users

With the information provided by the risky users report, administrators can find:

- Which users are at risk, have had risk remediated, or have had risk dismissed?
- Details about detections.
- History of all risky sign-ins.
- Risk history.

Administrators can then choose to take action on these events. They can choose to:

- Reset the user password.
- Confirm user compromise.
- Dismiss user risk.
- Block user from signing in.
- Investigate further using Azure ATP.

Risky sign-ins

The risky sign-ins report contains filterable data for up to the past 30 days (one month).

With the information provided by the risky sign-ins report, administrators can find:

- Which sign-ins are classified as at risk, confirmed compromised, confirmed safe, dismissed, or remediated.
- Real-time and aggregate risk levels associated with sign-in attempts.
- Detection types triggered.
- Conditional Access policies applied.
- MFA details.
- Device information.
- Application information.
- Location information.

Administrators can then choose to take action on these events. Administrators can choose to:

- Confirm sign-in compromise.
- Confirm sign-in safe.

Note - Identity Protection evaluates risk for all authentication flows, whether it be interactive or non-interactive. However, the sign-in report shows only the interactive sign-ins. You may see risky sign-ins that occurred on non-interactive sign-ins, but the sign-in will not show up in the Azure AD sign-ins report.

Risk detections

The risk detections report contains filterable data for up to the past 90 days (three months).

With the information provided by the risk detections report, administrators can find:

- Information about each risk detection including type.
- Other risks triggered at the same time.
- Sign-in attempt location.

Administrators can then choose to return to the user's risk or sign-ins report to take actions based on information gathered.

The risk detection report also provides a clickable link to the detection in the Microsoft Cloud App Security (MCAS) portal where you can view additional logs and alerts.

Note - Our system may detect that the risk event that contributed to the risk user risk score was a false positive or that the user risk was remediated with policy enforcement such as completing an MFA prompt

or secure password change. Therefore, our system will dismiss the risk state, and a risk detail of "AI confirmed sign-in safe" will surface and no longer contribute to the user's risk.

Remediate risks and unlock users

After completing your investigation, you will want to take action to remediate the risk or unlock users. Organizations also have the option to enable automated remediation using their risk policies. Organizations should try to close all risk detections that they are presented with in a time period your organization is comfortable with. Microsoft recommends closing events as soon as possible because time matters when working with risk.

Remediation

All active risk detections contribute to the calculation of a value called *user risk level*. The user risk level is an indicator (low, medium, high) for the probability that an account has been compromised. As an administrator, you want to get all risk detections closed, so that the affected users are no longer at risk.

Some risk detections may be marked by Identity Protection as "Closed (system)" because the events were no longer determined to be risky.

Administrators have the following options to remediate:

- Self-remediation with risk policy.
- Manual password reset.
- Dismiss user risk.
- Close individual risk detections manually.

Self-remediation with risk policy

If you allow users to self-remediate, with Azure AD Multi-Factor Authentication (MFA) and self-service password reset (SSPR) in your risk policies, they can unblock themselves when risk is detected. These detections are then considered closed. Users must have previously registered for Azure AD MFA and SSPR in order to use when risk is detected.

Some detections may not raise risk to the level where a user self-remediation would be required, but administrators should still evaluate these detections. Administrators may determine that additional measures are necessary, such as blocking access from locations or lowering the acceptable risk in their policies.

Manual password reset

If requiring a password reset using a user risk policy is not an option, administrators can close all risk detections for a user with a manual password reset.

Administrators are given two options when resetting a password for their users:

Generate a temporary password - By generating a temporary password, you can immediately bring an identity back into a safe state. This method requires contacting the affected users since they need to know what the temporary password is. Because the password is temporary, the user is prompted to change the password to something new during the next sign-in.

Require the user to reset password - Requiring the users to reset passwords enables self-recovery without contacting help desk or an administrator. This method only applies to users who are registered for Azure AD MFA and SSPR. For users who have not been registered, this option isn't available.

Dismiss user risk

If a password reset is not an option for you because, for example, the user has been deleted, you can choose to dismiss user risk detections.

When you click **Dismiss user risk**, all events are closed and the affected user is no longer at risk. However, because this method doesn't have an impact on the existing password, it doesn't bring the related identity back into a safe state.

Close individual risk detections manually

By closing individual risk detections manually, you can lower the user risk level. Typically, risk detections are closed manually in response to a related investigation, such as when talking to a user reveals that an active risk detection is not required anymore.

When closing risk detections manually, you can choose to take any of the following actions to change the status of a risk detection:

- Confirm user compromised.
- Dismiss user risk.
- Confirm sign-in safe.
- Confirm sign-in compromised.

Unblocking users

An administrator may choose to block a sign-in based on their risk policy or investigations. A block may occur based on either sign-in or user risk.

Unblocking based on user risk

To unblock an account blocked due to user risk, administrators have the following options:

- **Reset password** - You can reset the user's password.
- **Dismiss user risk** - The user risk policy blocks a user if the configured user risk level for blocking access has been reached. You can reduce a user's risk level by dismissing user risk or manually closing reported risk detections.
- **Exclude the user from policy** - If you think that the current configuration of your sign-in policy is causing issues for specific users, you can exclude the users from it.
- **Disable policy** - If you think that your policy configuration is causing issues for all your users, you can disable the policy.

Unlocking based on sign-in risk

To unblock an account based on sign-in risk, administrators have the following options:

- **Sign in from a familiar location or device** - A common reason for blocked suspicious sign-ins are sign-in attempts from unfamiliar locations or devices. Your users can quickly determine whether this reason is the blocking reason by trying to sign in from a familiar location or device.
- **Exclude the user from policy** - If you think that the current configuration of your sign-in policy is causing issues for specific users, you can exclude the users from it.
- **Disable policy** - If you think that your policy configuration is causing issues for all your users, you can disable the policy.

PowerShell preview

Using the Microsoft Graph PowerShell SDK Preview module, organizations can manage risk using PowerShell. The preview modules and sample code are located in the Azure AD GitHub repo <https://github.com/AzureAD/IdentityProtectionTools>.

Use the Microsoft Graph API

Microsoft Graph is the Microsoft unified API endpoint and the home of Azure Active Directory Identity Protection APIs. There are three APIs that expose information about risky users and sign-ins: riskDetection, riskyUsers, and signIn.

riskDetection allows you to query Microsoft Graph for a list of both user and sign-in linked risk detections and associated information about the detection.

riskyUsers allows you to query Microsoft Graph for information about users that Identity Protection detected as being risky.

signIn allows you to query Microsoft Graph for information on Azure AD sign-ins with specific properties related to risk state, detail, and level.

This section gets you started with connecting to the Microsoft Graph and querying these APIs. For an in-depth introduction, full documentation, and access to the Graph Explorer, see the Microsoft Graph site <https://graph.microsoft.io/> or the specific reference documentation for the riskDetection, riskyUsers, and signIn APIs.

Connect to Microsoft Graph

There are four steps to accessing Identity Protection data through Microsoft Graph: retrieve your domain name, create a new app registration, configure API permissions, and configure a valid credential.

Retrieve your domain name

1. Sign in to the Azure portal.
2. Browse to **Azure Active Directory** then open **Custom domain names**.
3. Take note of the .onmicrosoft.com domain. You will need this information in a later step.

Create a new app registration

1. In the Azure portal, browse to **Azure Active Directory** then open **App registrations**.

2. Select **New registration**.
3. On the **Create** page, perform the following steps:
 1. In the **Name** textbox, type a name for your application (for example: Azure AD Risk Detection API).
 2. Under **Supported account types**, select the type of accounts that will use the APIs.
 3. Select **Register**.
4. Copy the **Application ID**.

Configure API permissions

1. From the **Application** you created, select **API permissions**.
2. On the **Configured permissions** page, in the toolbar on the top, click **Add a permission**.
3. On the **Add API access** page, click **Select an API**.
4. On the **Select an API** page, select **Microsoft Graph**, and then click **Select**.
5. On the **Request API permissions** page:
 1. Select **Application permissions**.
 2. Select the checkboxes next to `IdentityRiskEvent.Read.All` and `IdentityRiskyUser.Read.All`.
 3. Select **Add permissions**.
6. Select **Grant admin consent for domain**.

Configure a valid credential

1. From the **Application** you created, select **Certificates & secrets**.
2. Under **Client secrets**, select **New client secret**.
 1. Give the client secret a **Description** and set the expiration time period according to your organizational policies.
 2. Select **Add**.

Note - If you lose this key, you will have to return to this section and create a new key. Keep this key a secret: Anyone who has it can access your data.

Authenticate to Microsoft Graph and query the Identity Protection risk detections API

At this point, you should have:

- The name of your tenant's domain
- The Application (client) ID
- The client secret or certificate

To authenticate, send a post request to <https://login.microsoft.com> with the following parameters in the body:

- `grant_type: "client_credentials"`

- resource: <https://graph.microsoft.com>
- client_id: **your client ID**
- client_secret: **your key**

If successful, this request returns an authentication token. To call the API, create a header with the following parameter:

```
Authorization="**token_type**      **access_token**"
```

When authenticating, you can find the token type and access token in the returned token.

Send this header as a request to the following API URL: <https://graph.microsoft.com/v1.0/identityProtection/riskDetections>

The response, if successful, is a collection of identity risk detections and associated data in the OData JSON format, which can be parsed and handled as you see fit.

Sample

This sample shows the use of a shared secret to authenticate. In a production environment, storing secrets in code is generally frowned upon. Organizations can use managed identities for Azure resources to secure these credentials.

Here's sample code for authenticating and calling the API using PowerShell. Just add your client ID, the secret key, and the tenant domain.

```
$ClientID      = "**your client ID here**"          # Should be a ~36 hex
character string; insert your info here

$ClientSecret   = "**your client secret here**"        # Should be a ~44
character string; insert your info here

$tenantdomain   = "**your tenant domain here**"        # For example,
contoso.onmicrosoft.com

$loginURL       = "https://login.microsoft.com"

$resource        = "https://graph.microsoft.com"

$body           = @{
    grant_type="client_credentials";
    resource=$resource;
    client_id=$ClientID;
    client_secret=$ClientSecret
}

$oauth           = Invoke-RestMethod -Method Post -Uri $loginURL/$tenantdo-
main/oauth2/token?api-version=1.0 -Body $body

Write-Output $oauth

if ($oauth.access_token -ne $null) {

    $headerParams = @{'Authorization'="$($oauth.token_type) $($oauth.access_token)"}

    $url = "https://graph.microsoft.com/v1.0/identityProtection/riskDe-
```

```
tectors"
Write-Output $url

$myReport = (Invoke-WebRequest -UseBasicParsing -Headers $header-
Params -Uri $url)

foreach ($event in ($myReport.Content | ConvertFrom-Json).value) {

    Write-Output $event

}

} else {

    Write-Host "ERROR: No Access Token"

}
```

Get all of the offline risk detections (riskDetection API)

With Identity Protection sign-in risk policies, you can apply conditions when risk is detected in real time. But what about detections that are discovered offline? To understand what detections occurred offline and, thus, would not have triggered the sign-in risk policy, you can query the riskDetection API.

```
GET https://graph.microsoft.com/v1.0/identityProtection/riskDetections?$fil-
ter=detectionTimingType eq 'offline'
```

Get all of the users who successfully passed an MFA chal- lenge triggered by risky sign-ins policy (riskyUsers API)

To understand the impact Identity Protection risk-based policies have on your organization, you can query all of the users who successfully passed an MFA challenge triggered by a risky sign-ins policy. This information can help you understand which users Identity Protection may have falsely detected as a risk and which of your legitimate users may be performing actions that the AI deems risky.

```
GET https://graph.microsoft.com/v1.0/identityProtection/riskyUsers?$fil-
ter=riskDetail eq 'userPassedMFADrivenByRiskBasedPolicy'
```

Module 2 Review Questions

Module 2 Review Questions

Review Question 1

Which task can a user with the Security Operator role perform?

- Configure alerts
- Confirm safe sign-in
- Reset a password for a user

Review Question 2

There are two risk policies that can be enabled in the directory. One is user risk policy. Which is the other risk policy?

- Mobile device access risk policy
- Sign-in risk policy
- Hybrid identity sign-in risk policy

Review Question 3

In Microsoft Graph, which three APIs expose information about risky users and sign-ins?

- riskDetection, riskyUsers, signIn
- riskDetection, itemActivity, signIn
- riskyUsers, signIn, IdentitySet

Review Question 4

What action does Conditional Access perform?

- It is the component that enforces multifactor authentication policies for access.
- It analyzes signals such as user, device, and location to enforce organizational access policies.
- It monitors and logs all access attempts.

Module 2 Hands-on Exercises

Lab 12: Enable Azure AD multi-factor authentication

To download the most recent version of this lab, please visit the SC-300 [GitHub repository¹⁰](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

To improve security in your organization, you've been directed to enable multi-factor authentication for Azure Active Directory.

Objectives

After you complete this lab, you will be able to:

- Configure Multi-Factor Authentication options
- Setup conditional access rules for MFA
- Configure Azure AD MFA for passwords

Lab setup

- Estimated time: 10 minutes

Lab 13: Configure and deploy self-service password reset

To download the most recent version of this lab, please visit the SC-300 [GitHub repository¹¹](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

The company has decided to empower the employees and enable self-service password reset. You must configure this setting in your organization.

¹⁰ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

¹¹ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

Objectives

After you complete this lab, you will be able to:

- Add a new user
- Enable SSPR
- Register for SSPR
- Test SSPR

Lab setup

- Estimated time: 15 minutes

Lab 14: Working with security defaults

To download the most recent version of this lab, please visit the SC-300 [GitHub repository¹²](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

You must configure the Azure Active Directory security default settings in your organization.

Objectives

After you complete this lab, you will be able to:

- Enabling security defaults
- Disabling security defaults

Lab setup

- Estimated time: 5 minutes

Lab 15: Implement and test a conditional access policy

To download the most recent version of this lab, please visit the SC-300 [GitHub repository¹³](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

¹² <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

¹³ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

Lab scenario

Your organization needs to be able to limit user access to its internal applications. You must deploy an Azure Active Directory conditional access policy.

Objectives

After you complete this lab, you will be able to:

- Create a conditional access policy
- Test the conditional access policy

Lab setup

- Estimated time: 10 minutes

Lab 16: Configure authentication session controls

To download the most recent version of this lab, please visit the SC-300 [GitHub repository¹⁴](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

As part of your company's larger security configuration, you must test a conditional access policy that can be used to control sign in frequency.

Objectives

After you complete this lab, you will be able to:

- Configure sign in frequency controls using a conditional access policy

Lab setup

- Estimated time: 10 minutes

Lab 17: Manage Azure AD smart lockout values

To download the most recent version of this lab, please visit the SC-300 [GitHub repository¹⁵](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version

¹⁴ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

¹⁵ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

You must configure the additional password protection settings for your organization.

Objectives

After you complete this lab, you will be able to:

- Manage Azure AD smart lockout values

Lab setup

- Estimated time: 5 minutes

Lab 18: Enable sign in and user risk policies

To download the most recent version of this lab, please visit the SC-300 [GitHub repository¹⁶](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

As an additional layer of security, you need to enable and configure your Azure AD organization's sign in and user risk policies.

Objectives

After you complete this lab, you will be able to:

- Enable User risk policy
- Enable Sign-in risk policy

Lab setup

- Estimated time: 10 minutes

Lab 19: Configure an Azure AD multi-factor authentication registration policy

To download the most recent version of this lab, please visit the SC-300 [GitHub repository¹⁷](#).

¹⁶ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

¹⁷ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

Azure AD multi-factor authentication provides a means to verify who you are using more than just a username and password. It provides a second layer of security to user sign-ins. For users to be able to respond to MFA prompts, they must first register for Azure AD Multi-Factor Authentication. You must configure your Azure AD organization's MFA registration policy to be assigned to all users.

Objectives

After you complete this lab, you will be able to:

- Policy configuration

Lab setup

- Estimated time: 5 minutes

Module 2 Summary

Summary for Module 2

During this module you saw how companies can use their Azure AD and on-premises AD systems to run their business and protect their users and resources. We saw how you can manage how your users authenticate and then use conditional access to ensure they have secure access only to the resources they need. Finally, we looked at how identity protection helps to protect your users.

Plan and Implement Azure AD Multifactor Authentication

During this lesson you explored how that:

Using Azure AD Multi-Factor Authentication, you can ensure that when users sign in to access your confidential systems and data, they are who they say they are. Azure AD allows you to create policies to ensure that specific apps are protected, while allowing more public systems to remain easier to get to. Skills learned:

- Learn about Azure AD Multi-Factor Authentication (MFA)
- Create a plan to deploy Azure AD MFA
- Turn on Azure AD MFA for users and specific apps

Manage User Authentication

During this lesson you explored how to:

- Administer authentication methods (FIDO2/Passwordless).
- Implement an authentication solution based on Windows Hello for Business.
- Configure and deploy self-service password reset.
- Deploy and manage password protection.
- Implement and manage tenant restrictions.

Plan, Implement, and Administer Conditional Access

During this lesson you explored how to:

- Plan and implement security defaults.
- Plan Conditional Access policies.
- Implement Conditional Access policy controls and assignments (targeting, applications, and conditions).
- Test and troubleshoot Conditional Access policies.
- Implement application controls.
- Implement session management.
- Configure smart lockout thresholds.

Mand Azure AD Identity Protection

During this lesson you explored how to:

- Review Identity Protection basics.
- Implement and manage a user risk policy.
- Implement and manage sign-in risk policies.
- Implement and manage MFA registration policy.
- Monitor, investigate, and remediate elevated risky users.

Supplemental Resources

Use these resources to discover more:

- **Identity Protection policies¹⁸**
- **What is Azure Active Directory Identity Protection?¹⁹**
- **Planning a cloud-based Azure AD Multi-Factor Authentication deployment²⁰**
- **Deploy Azure AD self-service password reset²¹**
- **Enable combined security information registration in Azure Active Directory²²**
- **Create a resilient access control management strategy in Azure AD²³**
- **Microsoft Graph REST API beta²⁴**
- **Windows Hello for Business overview²⁵**
- **Microsoft Authenticator app²⁶**
- **Passwordless authentication options for Azure Active Directory²⁷**
- **Authentication methods in Azure Active Directory - OATH tokens²⁸**
- **Configure and enable users for SMS-based authentication using Azure Active Directory (preview)²⁹**
- **Authentication methods in Azure Active Directory - phone options³⁰**
- **Conceptual overview of Azure AD Password Protection³¹**
- **Enable on-premises Azure Active Directory Password Protection³²**
- **Step-By-Step: Implementing Azure AD Password Protection On-Premises³³**

¹⁸ <https://docs.microsoft.com/azure/active-directory/identity-protection/concept-identity-protection-policies>

¹⁹ <https://docs.microsoft.com/azure/active-directory/active-directory-identityprotection>

²⁰ <https://docs.microsoft.com/azure/active-directory/authentication/howto-mfa-getstarted>

²¹ <https://docs.microsoft.com/azure/active-directory/authentication/howto-sspr-deployment>

²² <https://docs.microsoft.com/azure/active-directory/authentication/howto-registration-mfa-sspr-combined>

²³ <https://docs.microsoft.com/azure/active-directory/authentication/concept-resilient-controls>

²⁴ <https://docs.microsoft.com/graph/api/resources/authenticationmethods-overview>

²⁵ <https://docs.microsoft.com/windows/security/identity-protection/hello-for-business/hello-overview>

²⁶ <https://docs.microsoft.com/azure/active-directory/authentication/concept-authentication-authenticator-app>

²⁷ <https://docs.microsoft.com/azure/active-directory/authentication/concept-authentication-passwordless>

²⁸ <https://docs.microsoft.com/azure/active-directory/authentication/concept-authentication-oath-tokens>

²⁹ <https://docs.microsoft.com/azure/active-directory/authentication/howto-authentication-sms-signin>

³⁰ <https://docs.microsoft.com/azure/active-directory/authentication/concept-authentication-phone-options>

³¹ <https://docs.microsoft.com/azure/active-directory/authentication/concept-password-ban-bad-on-premises>

³² <https://docs.microsoft.com/azure/active-directory/authentication/howto-password-ban-bad-on-premises-operations>

³³ <https://techcommunity.microsoft.com/t5/itops-talk-blog/step-by-step-implementing-azure-ad-password-protection-on/ba-p/563342>

- **What is Conditional Access?**³⁴
- **Using the location condition in a Conditional Access policy**³⁵
- **Use compliance policies to set rules for devices you manage with Intune**³⁶
- **Introducing security defaults**³⁷
- **Plan a Conditional Access deployment**³⁸
- **Enabling combined security information registration in Azure Active Directory**³⁹
- **Manage emergency access accounts in Azure AD**⁴⁰
- **How To: Configure and enable risk policies**⁴¹
- **What are managed identities for Azure resources?**⁴²
- **Remediate risks and unblock users**⁴³
- **Azure Active Directory Identity Protection notifications**⁴⁴

³⁴ <https://youtu.be/ffMAw2IV07A>

³⁵ <https://docs.microsoft.com/azure/active-directory/conditional-access/location-condition>

³⁶ <https://docs.microsoft.com/mem/intune/protect/device-compliance-get-started>

³⁷ <https://techcommunity.microsoft.com/t5/azure-active-directory-identity/introducing-security-defaults/ba-p/1061414>

³⁸ <https://docs.microsoft.com/azure/active-directory/conditional-access/plan-conditional-access>

³⁹ <https://docs.microsoft.com/azure/active-directory/authentication/howto-registration-mfa-sspr-combined>

⁴⁰ <https://docs.microsoft.com/azure/active-directory/roles/security-emergency-access>

⁴¹ <https://docs.microsoft.com/azure/active-directory/identity-protection/howto-identity-protection-configure-risk-policies>

⁴² <https://docs.microsoft.com/azure/active-directory/managed-identities-azure-resources/overview>

⁴³ <https://docs.microsoft.com/azure/active-directory/identity-protection/howto-identity-protection-remediate-unblock>

⁴⁴ <https://docs.microsoft.com/azure/active-directory/identity-protection/howto-identity-protection-configure-notifications>

Answers

Review Question 1

Which task can a user with the Security Operator role perform?

- Configure alerts
- Confirm safe sign-in
- Reset a password for a user

Explanation

Security Operators can view all Identity Protection reports and the Overview blade, dismiss user risk, confirm safe sign-in, and confirm compromise.

Review Question 2

There are two risk policies that can be enabled in the directory. One is user risk policy. Which is the other risk policy?

- Mobile device access risk policy
- Sign-in risk policy
- Hybrid identity sign-in risk policy

Explanation

Sign-in risk policy: The sign-in risk policy detects suspicious actions that come along with the sign-in. It is focused on the sign-in activity itself and analyzes the probability that the sign-in may not have been performed by the user.

Review Question 3

In Microsoft Graph, which three APIs expose information about risky users and sign-ins?

- riskDetection, riskyUsers, signIn
- riskDetection, itemActivity, signIn
- riskyUsers, signIn, IdentitySet

Explanation

There are three APIs that expose information about risky users and sign-ins. The first API, riskDetection, allows you to query Microsoft Graph for a list of both user and sign-in linked risk detections and associated information about the detection. The second API, riskyUsers, allows you to query Microsoft Graph for information about users that Identity Protection detected as being risky. The third API, signIn, allows you to query Microsoft Graph for information on Azure AD sign-ins with specific properties related to risk state, detail, and level.

Review Question 4

What action does Conditional Access perform?

- It is the component that enforces multifactor authentication policies for access.
- It analyzes signals such as user, device, and location to enforce organizational access policies.
- It monitors and logs all access attempts.

Explanation

Conditional Access is the tool used by Azure Active Directory to bring signals together, make decisions, and enforce organizational policies. Conditional Access is at the heart of the new identity-driven control plane.

Module 3 Implement Access Management for Apps

Learning Objectives

Learning Objectives

After completing this module, you'll be able to:

- Plan, implement, and monitor the integration of Enterprise Apps for Single Sign-On (SSO)
 - implement and configure consent settings
 - discover apps by using MCAS or ADFS app report
 - design and implement access management for apps
 - design and implement app management roles
 - monitor and audit access / Sign-Ons to Azure Active Directory integrated enterprise applications
 - implement token customizations
 - integrate on-premises apps by using Azure AD application proxy
 - integrate custom SaaS apps for SSO
 - configure pre-integrated (gallery) SaaS apps
 - implement application user provisioning
- Implement app registrations
 - plan your line of business application registration strategy
 - implement application registrations
 - configure application permissions
 - implement application authorization

- plan and configure multi-tier application permissions

Plan and Design Integration of Enterprise Apps for SSO

Introduction

In this module, you will discover apps. You will also design and implement access management and app management roles. In addition, you will configure pre-integrated (gallery) SaaS apps.

Learning objectives

In this module, you will:

- Discover apps by using MCAS or ADFS app report.
- Design and implement access management for apps.
- Design and implement app management roles.
- Configure pre-integrated (gallery) SaaS apps.

Prerequisites

none

Discover Apps using MCAS and App Report

To start learning how to protect cloud apps, you first need to learn what Cloud Access Security Broker (CASB) is. Then, learn what the Microsoft implementation of CASB is.

CASB - Cloud Access Security Broker - An on-premises or cloud-based security policy enforcement point, placed between cloud service consumers and cloud service providers to combine and interject enterprise security policies as the cloud-based resources are accessed.

MCAS - Microsoft Cloud App Security - Microsoft implementation of a CASB service to protect data, services, and applications with enterprise policies. It provides supplemental reporting and analytics services

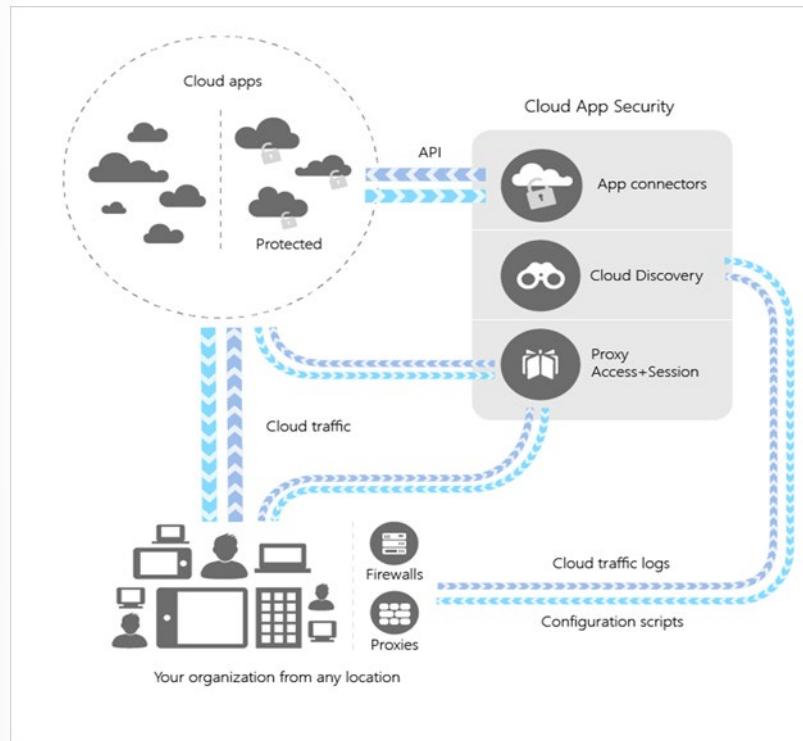
Microsoft Cloud App Security

Moving to the cloud increases flexibility for employees and IT alike. However, it also introduces new challenges and complexities for keeping your organization secure. 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 Cloud App Security is a Cloud Access Security Broker (CASB) that supports various deployment modes, including log collection, API connectors, and reverse proxy. It provides rich visibility, control over data travel, and sophisticated analytics to identify and combat cyberthreats across all your Microsoft and third-party cloud services. Microsoft Cloud App Security natively integrates with leading Microsoft solutions and is designed with security professionals in mind. It provides simple deployment, centralized management, and innovative automation capabilities. Microsoft Cloud App Security is a comprehensive cross-SaaS solution bringing deep visibility, strong data controls, and enhanced threat protection to your cloud apps. Cloud Discovery, a feature of Cloud App Security, enables you to gain visibility into Shadow IT by discovering cloud apps in use.

Architecture

Cloud App Security integrates visibility with your cloud by:

- Using Cloud Discovery to map and identify your cloud environment and the cloud apps your organization is using.
- Sanctioning and de-authorizing apps in your cloud.
- Using easy-to-deploy app connectors that take advantage of provider APIs, for visibility and governance of apps that you connect to.
- Using Conditional Access App Control protection to get real-time visibility and control over access and activities within your cloud apps.
- Helping you have continuous control by setting and continually fine-tuning policies.



Cloud Discovery

Cloud Discovery uses your traffic logs to dynamically discover and analyze the cloud apps your organization is using. To create a snapshot report of your organization's cloud use, manually upload log files from your firewalls or proxies for analysis. To set up continuous reports, use Cloud App Security log collectors to periodically forward your logs.

Review the Cloud Discovery Dashboard

The first thing you should do to get a general picture of your Cloud Discovery apps is review the following information in the Cloud Discovery Dashboard:

- First look at the overall cloud app use in your organization in the High-level usage overview.
- Then, dive one level deeper to see which are the top categories used in your org for each of the different use parameters. You can see how much of this usage is by Sanction apps.
- Go even deeper and see all the apps in a specific category in the Discovered apps tab.
- You can see the top users and source IP addresses to identify which users are the most dominant users of cloud apps in your organization.
- Check how the discovered apps spread according to geographic location (according to their HQ) in the App Headquarters map.
- Finally, don't forget to review the risk score of the discovered app in the App risk overview. Check the discovery alerts status to see how many open alerts should you investigate.

Filtering Discovered Apps

- **App tag** - Select whether the app was sanctioned or unsanctioned or not tagged. Additionally, you can create a custom tag for your app and then use it to filter for specific types of apps.
- **Apps and domains** - Enables you to search for specific apps or apps used in specific domains.
- **Categories** - The categories filter, located on the left of the page, enables you to search for types of apps according to app categories. Example categories include social network apps, cloud storage apps, and hosting services. You can select multiple categories at a time, or a single category, then apply the basic and advanced filters on top.

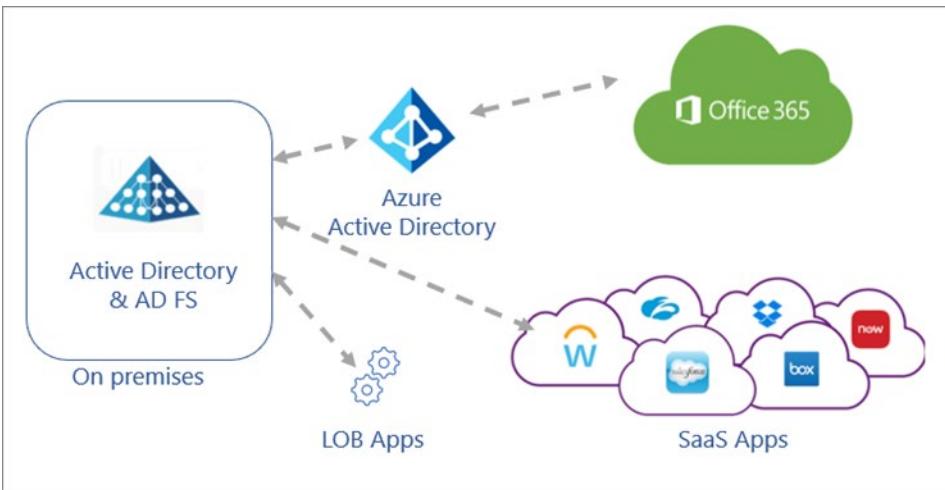
- **Compliance risk factor** - Lets you search for a specific standards, certification, and compliance that the app may comply with (HIPAA, ISO 27001, SOC 2, PCI-DSS, and more.).
- **General risk factor** - Lets you search for general risk factors such as consumer popularity, data center locale, and more.
- **Risk score** - Lets you filter apps by risk score so that you can focus on, for example, reviewing only highly risky apps. You can also override the risk score set by Cloud App Security. For more information, see Working with the risk score.
- **Security risk factor** - Enables you to filter based on specific security measures (such as Encryption at rest, multifactor authentication, etc.).
- **Usage** - Lets you filter based on the usage statistics of this app. Usage such as apps with less than or more than a specified number of data uploads, apps with more than or less than a specified number of Users.
- **Legal risk factor** - Lets you filter based on all the regulations and policies that are in-place to ensure data protection and privacy of the app's users. Examples include GDPR ready cloud apps, DMCA, and data retention policy.

Sanctioning and unsanctioning an app

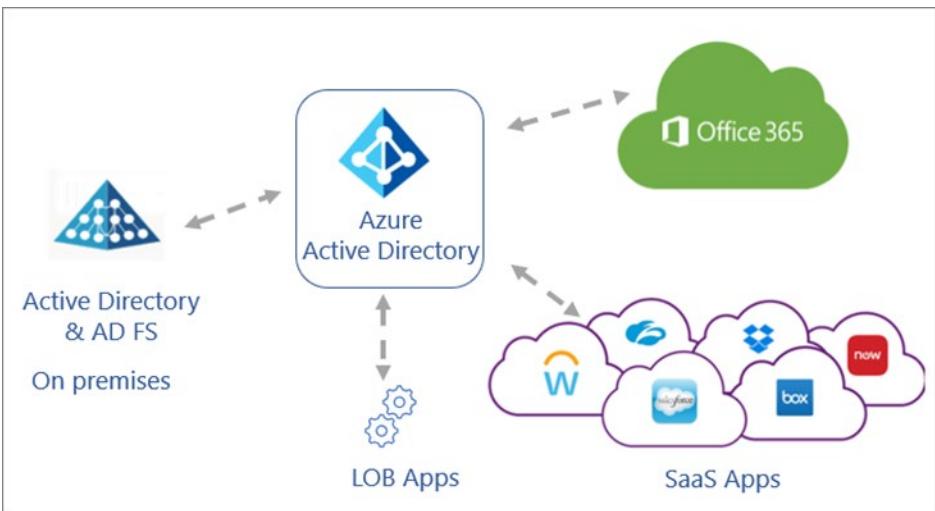
You can use Cloud App Security to sanction or unsanction apps in your organization by using the **Cloud app catalog**. The Microsoft team of analysts has an extensive and continuously growing catalog of more than 16,000 cloud apps that are ranked and scored based on industry standards. Use the Cloud app catalog to rate the risk for your cloud apps based on regulatory certifications, industry standards, and best practices. Then, customize the scores and weights of various parameters to your organization's needs. Based on these scores, Cloud App Security lets you know how risky an app is. Scoring is based on more than 80 risk factors that might affect your environment.

Active Directory Federation Services

If you have an on-premises directory that contains user accounts, you likely have many applications to which users authenticate. Each of these apps is configured for users to access using their identities. Users may also authenticate directly with your on-premises Active Directory. Active Directory Federation Services (AD FS) is a standards-based on-premises identity service. AD FS extends the ability to use single sign-on (SSO) functionality between trusted business partners without requiring users to sign in separately to each application. This is known as federation. Many organizations have software as a service (SaaS) or custom line-of-business (LOB) apps federated directly to AD FS, alongside Microsoft 365 and Azure AD-based apps.



To increase application security, your goal is to have a single set of access controls and policies across your on-premises and cloud environments.



Many organizations use AD FS to provide SSO to cloud applications. Moving your AD FS applications to Azure AD for authentication provides significant benefits, especially in terms of cost management, risk management, productivity, compliance, and governance. But understanding which applications are compatible with Azure AD and identifying specific migration steps can be time consuming.

Sometimes the organization may be using alternate on-premises or cloud identity providers, such as SiteMinder, Oracle Access Manager, PingFederate, etc. Most of them are on-premises installations. Some cloud providers, such as Okta and OneLogin, offer similar services.

The AD FS application activity report in the Azure portal enables you to quickly identify which applications you can migrate to Azure AD. It assesses all AD FS applications for compatibility with Azure AD, checks for any issues, and gives guidance on preparing individual applications for migration. With the AD FS application activity report, you can discover AD FS applications and scope your migration. The AD FS application activity report lists all AD FS applications in your organization that have had an active user login in the last 30 days. The activity data is available to users who are assigned any of these admin roles: global administrator, report reader, security reader, application administrator, or cloud application administrator.

Types of apps to migrate

Migrating all your application authentication to Azure AD is optimal, as it gives you a single control plane for identity and access management.

There are two types of applications to migrate:

1. SaaS applications, which are generally procured by the organization.
2. Line-of-business applications, which are developed by the organization and not meant to be used by other companies.

Your applications may use modern or legacy protocols for authentication. Most SaaS applications use modern authentication protocols and provide guidance on how to enable SSO. Consider first migrating applications that use modern authentication protocols (such as SAML and Open ID Connect). These apps can be reconfigured to authenticate with Azure AD via either a built-in connector in our App Gallery, or by registering the application in Azure AD. Integrate apps using older protocols by using **Application Proxy**¹ and/or Azure AD Domain Services.

Discover AD FS applications that can be migrated

The AD FS application activity report is available in the Azure portal under Azure AD **Usage & insights** reporting. The AD FS application activity report analyzes each AD FS application to determine whether it can be migrated as-is or additional review is needed.

1. Sign in to the Azure portal with an admin role that has access to AD FS application activity data (global administrator, report reader, security reader, application administrator, or cloud application administrator).
2. Select **Azure Active Directory**, and then select **Enterprise applications**.
3. Under **Activity**, select **Usage & insights (Preview)**, and then select **AD FS application activity** to open a list of all AD FS applications in your organization.

Application Identifier	Unique User Count	Migration status
CL clearforce_sp_ms	17	⚠ Additional steps required
CV cvent	6	⚠ Additional steps required
FI financialknowledge.net	26	⚠ Additional steps required
GE getabstract	79	⚠ Additional steps required
OA http://identity.office.net	31	⚠ Additional steps required
AR http://mssource.sourcing.ariba.com	32	⚠ Additional steps required
WS http://nebulaservices.tenantsite	31	⚠ Additional steps required
EB http://shibboleth.ebscohost.com	1	⚠ Additional steps required
AA https://appmapadmintool.cloudapp.net	1	⚠ Additional steps required
AP https://apportal.osdfinfra.net	2294	⚠ Additional steps required
L https://authorcms.skype.net	2	⚠ Additional steps required
AX https://axo.citsamex.com	4	⚠ Additional steps required

¹ <https://docs.microsoft.com/azure/active-directory/manage-apps/what-is-application-proxy>

4. For each application in the AD FS application activity list, view the **Migration status**:
 - **Ready to migrate** means the AD FS application configuration is fully supported in Azure AD and can be migrated as-is.
 - **Needs review** means some of the application's settings can be migrated to Azure AD, but you'll need to review the settings that can't be migrated as-is.
 - **Additional steps required** means Azure AD doesn't support some of the application's settings, so the application can't be migrated in its current state.

Design and Implement App Management Roles

This unit describes how to use permissions granted by custom roles in Azure AD to address your application management needs. In Azure AD, you can delegate application creation and management permissions by:

- Restricting who can create applications and manage the applications they create.
- Assigning one or more owners to an application. This is a simple way to grant someone the ability to manage all aspects of Azure AD configuration for a specific application.
- Assigning a built-in administrative role that grants access to manage configuration in Azure AD for all applications. This is the recommended way to grant IT experts access to manage broad application configuration permissions without granting access to manage other parts of Azure AD not related to application configuration.
- Creating a custom role defining specific permissions and assigning it to someone either to the scope of a single application as a limited owner, or at the directory scope (all applications) as a limited administrator.

It's important to consider granting access using one of the above methods for two reasons. First, delegating the ability to perform administrative tasks reduces global administrator overhead. Second, using limited permissions improves your security posture and reduces the potential for unauthorized access.

Restrict who can create applications

In Azure AD, all users can register application registrations and manage all aspects of applications they create. Everyone also has the ability to consent to apps accessing company data on their behalf. You can choose to selectively grant those permissions by setting the global switches to "No" and adding the selected users to the Application Developer role.

To disable the default ability to create application registrations or consent to applications

1. Sign in to your Azure AD organization with an account that's eligible for the Global Administrator role in your Azure AD organization.
2. Set one or both of the following:
 - On the **User settings** page for your organization, set the **Users can register applications** setting to No. This will disable the default ability for users to create application registrations.
 - On the User settings for enterprise applications, set the **Users can consent to applications accessing company data on their behalf** setting to No. This will disable the default ability for users to consent to applications accessing company data on their behalf.

Grant individual permissions to create and consent to applications when the default ability is disabled

Assign the Application Developer role to grant the ability to create application registrations when the **Users can register applications** setting is set to No. This role also grants permission to consent on one's own behalf when the **Users can consent to apps accessing company data on their behalf** setting is set to No. As a system behavior, when a user creates a new application registration, they are automatically added as the first owner. Ownership permissions give the user the ability to manage all aspects of an application registration or enterprise application that they own.

Assign application owners

Assigning owners is a simple way to grant the ability to manage all aspects of Azure AD configuration for a specific application registration or enterprise application. As a system behavior, when a user creates a new application registration, they are automatically added as the first owner. Ownership permissions give the user the ability to manage all aspects of an application registration or enterprise application that they own. The original owner can be removed and additional owners can be added.

Enterprise application owners

As an owner, a user can manage the organization-specific configuration of the enterprise application, such as the SSO configuration, provisioning, and user assignments. An owner can also add or remove other owners. Unlike Global Administrators, owners can manage only the enterprise applications they own.

In some cases, enterprise applications created from the application gallery include both an enterprise application and an application registration. When this is true, adding an owner to the enterprise application automatically adds the owner to the corresponding application registration as an owner.

To assign an owner to an enterprise application

1. Sign in to your Azure AD organization with an account that's eligible for the Application Administrator or Cloud Application Administrator for the organization.
2. On the **App registrations** page for the organization, select an app to open the Overview page for the app.
3. Select **Owners** to see the list of the owners for the app.
4. Select **Add** to select one or more owners to add to the app.

Important - Users and service principals can be owners of application registrations. Only users can be owners of enterprise applications. Groups cannot be assigned as owners of either. Owners can add credentials to an application and use those credentials to impersonate the application's identity. 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 or service principal. Depending on the application's permissions, an application owner could potentially create or update users or other objects while impersonating the application.

Assign built-in application admin roles

Azure AD has a set of built-in admin roles for granting access to manage configuration in Azure AD for all applications. These roles are the recommended way to grant IT experts access to manage broad applica-

tion configuration permissions without granting access to manage other parts of Azure AD not related to application configuration.

- **Application Administrator:** Users in this role can create and manage all aspects of enterprise applications, application registrations, and application proxy settings. This role also grants the ability to consent to delegated permissions and application permissions, excluding Microsoft Graph. Users assigned to this role are not added as owners when creating new application registrations or enterprise applications.
- **Cloud Application Administrator:** Users in this role have the same permissions as the Application Administrator role, excluding the ability to manage application proxy. Users assigned to this role are not added as owners when creating new application registrations or enterprise applications.

Important - Application Administrators and Cloud Application Administrators can add credentials to an application and use those credentials to impersonate the application's identity. The application may have permissions that are an elevation of privilege over the admin role's permissions. Depending on the application's permissions, an admin in one of these roles could potentially create or update users or other objects while impersonating the application. Neither role grants the ability to manage Conditional Access settings.

Create and assign a custom role

Creating custom roles and assigning custom roles are separate steps:

- Create a custom **role definition** and add permissions to it from a preset list. These are the same permissions used in the built-in roles.
- Create a **role assignment** to assign the custom role.

This separation enables you to create a single role definition and then assign it many times at different **scopes**. A custom role can be assigned at organization-wide scope or at the scope of a single Azure AD object. An example of an object scope is a single app registration. When using different scopes, however, the same role definition can be assigned to one person over all of the app registrations in the organization, and then to another person over only a single app or specific app registrations.

Tips when creating and using custom roles for delegating application management:

- Custom roles only grant access in the most current app registration blades of the Azure AD portal. They do not grant access in the legacy app registrations blades.
- Custom roles do not grant access to the Azure AD portal when the **Restrict access to Azure AD administration** portal user setting is set to Yes.
- For app registrations the user has access to, role assignments only show up in the **All applications** tab on the **App registration** page. They do not show up in the **Owned applications** tab.

Configure Pre-Integrated Gallery SaaS Apps

As you know, Azure AD has a gallery that contains thousands of pre-integrated applications. Many of the applications your organization uses are probably already in the gallery. If an app is added to your Azure AD tenant, you can configure properties for the app, manage user access to the app, and configure SSO so users can sign in to the app with their Azure AD credentials. This unit will show you how to configure properties for the app.

Configure app properties

To edit the application properties:

1. In the Azure AD portal, select **Enterprise applications**. Then find and select the application you want to configure.
2. In the **Manage** section, select **Properties** to open the **Properties** pane for editing.
3. Take a moment to understand the options available. The options available will depend on how the app is integrated with Azure AD. For example, an app that uses SAML-based SSO will have fields such as **User access URL** whereas an app that uses OIDC-based SSO will not. Apps added through **Azure Active Directory - App registrations** are by default OIDC-based apps, while apps added through **Azure Active Directory - Enterprise applications** might use any of a number of SSO standards. All apps will have fields for configuring when an app appears and can be used. These fields are:
 - **Enabled for users to sign in?** determines whether users assigned to the application can sign in.
 - **User assignment required?** determines whether users who aren't assigned to the application can sign in.
 - **Visible to users?** determines whether users assigned to an app can see it in **My Apps²** and Microsoft 365 app launcher. (See the waffle menu in the upper-left corner of a Microsoft 365 website.)

Tip - Assigning users happens on the Users and groups section of navigation.

The three options can be toggled independently of each other, and the resulting behavior is not always obvious. This table might help:

Enabled for users to sign in?	User assignment required?	Visible to users?	Behavior for users who have either been assigned to the app or not.
Yes	Yes	Yes	Assigned users can see the app and sign in. Unassigned users cannot see the app and cannot sign in.
Yes	Yes	No	Assigned users cannot see the app but they can sign in. Unassigned users cannot see the app and cannot sign in.
Yes	No	Yes	Assigned users can see the app and sign in. Unassigned users cannot see the app but can sign in.

² <https://myapps.microsoft.com/>

Enabled for users to sign in?	User assignment required?	Visible to users?	Behavior for users who have either been assigned to the app or not.
Yes	No	No	Assigned users cannot see the app but can sign in.Unassigned users cannot see the app but can sign in.
No	Yes	Yes	Assigned users cannot see the app and cannot sign in.Unassigned users cannot see the app and cannot sign in.
No	Yes	No	Assigned users cannot see the app and cannot sign in.Unassigned users cannot see the app and cannot sign in.
No	No	Yes	Assigned users cannot see the app and cannot sign in.Unassigned users cannot see the app and cannot sign in.
No	No	No	Assigned users cannot see the app and cannot sign in.Unassigned users cannot see the app and cannot sign in.

- When you're finished, select **Save**.

Use a custom logo

- To use a custom logo:
- Create a logo that's 215 by 215 pixels and save it in .png format.
- In the Azure AD portal, select **Enterprise applications**. Then find and select the application you want to configure.
- In the **Manage** section, select **Properties** to open the **Properties** pane for editing.
- Select the icon to upload the logo.
- When you're finished, select **Save**.

The screenshot shows the Azure AD portal's 'Enterprise Applications' section. On the left, a navigation pane lists various management sections like Overview, Deployment Plan, Diagnose and solve problems, Manage, Properties (which is selected and highlighted with a red box), Owners, Users and groups, Single sign-on, Provisioning, Self-service, Security, Conditional Access, Permissions, Token encryption (Preview), Activity, Sign-ins, Usage & insights (Preview), Audit logs, Provisioning logs (Preview), and Access reviews. The main pane displays the properties for the application 'GitHub-test.com - Properties'. It includes fields for Name (GitHub-test.com), Homepage URL (https://github.com/business), Logo (a thumbnail labeled 'new logo' with a file path 'new-logo.png' highlighted with a red box), User access URL, Application ID, Object ID, Terms of Service Url, Privacy Statement Url, Reply Url, User assignment required? (Yes), and Visible to users? (Yes).

Note - The thumbnail displayed on this Properties pane doesn't update right away. You can close and reopen the Properties pane to see the updated icon.

Add notes

You can use the notes field to add any information that is relevant for the management of the application.

1. In the Azure AD portal, select **Enterprise applications**. Then find and select the application you want to configure.
2. In the **Manage** section, select **Properties** to open the **Properties** pane for editing.
3. Update the Notes field, select **Save**.

GitHub.com | Properties

Enterprise Application

Save Discard Delete Got feedback?

Name: GitHub Enterprise Cloud - Organization

Homepage URL: <https://github.com/business>

Logo:



User access URL: <https://myapps.microsoft.com/signin/GitHub%20Enterprise%20Cloud...>

Application ID: dee2cdc2-7505-4746-9758-219fec5e3da0

Object ID: 5231fe30-5d18-4a45-98bd-929c903d24dd

Terms of Service Url: Publisher did not provide this information

Privacy Statement Url: Publisher did not provide this information

Reply URL: Publisher did not provide this information

User assignment required? Yes No

Visible to users? Yes No

Notes:

Implement and Monitor integration of Enterprise Apps for SSO

Introduction

In this module, you will learn how to implement token customizations and implement and configure consent settings. You will also learn how to integrate on-premises apps by using Azure Active Directory (AD) application proxy, and also integrate custom software as a service (SaaS) apps for single sign-on (SSO). You will learn how to implement application user provisioning and monitor and audit access and sign-on to Azure AD-integrated enterprise applications.

Learning objectives

In this module, you will:

- Implement token customizations
- Implement and configure consent settings
- Integrate on-premises apps by using Azure AD application proxy
- Integrate custom SaaS apps for SSO
- Implement application user provisioning
- Monitor and audit access/sign-on to Azure AD-integrated enterprise applications

Prerequisites

none

Implement Token Customizations

You can specify the lifetime of a token issued by Microsoft identity platform. Additionally, you can set token lifetimes for all apps in your organization, for a multi-tenant (multi-organization) application, or for a specific service principal in your organization. In Azure AD, a policy object represents a set of rules that are enforced on individual applications or on all applications in an organization. Each policy type has a unique structure, with a set of properties that are applied to objects to which they are assigned.

You can designate a policy as the default policy for your organization. The policy is applied to any application in the organization, as long as it is not overridden by a policy with a higher priority. You also can assign a policy to specific applications. The order of priority varies by policy type.

Configure authentication session management with Conditional Access

In complex deployments, organizations might have a need to restrict authentication sessions. These complex scenarios might include:

- Resource access from an unmanaged or shared device.
- Access to sensitive information from an external network.
- High impact users.

- Critical business applications.

Conditional Access controls allow you to create policies that target specific use cases within your organization without affecting all users.

Customize Tokens for Azure AD

Access and ID token lifetime	Refresh token lifetime (days)	Refresh token sliding windows lifetime	Lifetime length (days)
The lifetime of the OAuth 2.0 bearer token and ID tokens	The maximum time period before which a refresh token can be used to acquire a new access token	The refresh token sliding window type	After time period elapses, the user is forced to reauthenticate



Configure optional claims as part of your token

Application developers can use optional claims in their Azure AD applications to specify which claims they want in tokens sent to their application.

You can use optional claims to:

- Select additional claims to include in tokens for your application.
- Change the behavior of certain claims that the Microsoft identity platform returns in tokens.
- Add and access custom claims for your application.

While optional claims are supported in both v1.0 and v2.0 format tokens, as well as SAML tokens, they provide most of their value when moving from v1.0 to v2.0. One of the goals of the Microsoft identity platform is smaller token sizes to ensure optimal performance by clients. As a result, several claims formerly included in the access and ID tokens are no longer present in v2.0 tokens and must be asked for specifically on a per-application basis.

The screenshot shows the 'Token configuration' page for the GitHub application in the Azure portal. The left sidebar lists various management options. The main area shows the 'Optional claims' section, which is currently set to 'ID'. A detailed list of optional claims is provided:

Claim	Description
acct	User's account status in tenant
auth_time	Time when the user last authenticated; See OpenID Con...
ctry	User's country
email	The addressable email for this user, if the user has one
enfpolids	Enforced policy IDs; a list of the policy IDs that were eva...
family_name	Provides the last name, surname, or family name of the ...
fwd	IP address
given_name	Provides the first or "given" name of the user, as set on ...
home_oid	For guest users, the object ID of the user in the user's h...

Implement and Configure Consent Settings

You can integrate your applications with the Microsoft identity platform to allow users to sign in with their work or school account and access the organization's data to deliver rich data-driven experiences.

Before an application can access the organization's data, a user must grant the application permissions to do so. Different permissions allow different levels of access. By default, all users can consent to applications for permissions that do not require administrator consent. For example, by default, a user can consent to allow an app to access their mailbox. However, they cannot consent to allow an app unfettered access to read and write to all files in your organization.

By allowing users to grant apps access to data, users can easily acquire useful applications and be productive. However, in some situations this configuration can represent a risk if it is not carefully monitored and controlled.

Important - To reduce the risk of malicious applications attempting to trick users into granting them access to your organization's data, it is recommended that you allow user consent only for applications that have been published by a **verified publisher**³.

User consent settings

App consent policies describe conditions that must be met before an app can be consented to. These policies might include conditions on the app requesting access, as well as the permissions the app is requesting.

³ <https://docs.microsoft.com/azure/active-directory/develop/publisher-verification-overview>

By choosing which app consent policies apply for all users, you can set limits on when end users are allowed to grant consent to apps and when they will be required to request administrator review and approval.

- **Disable user consent** – Users cannot grant permissions to applications. Users can continue to sign into apps they had previously consented to or that are consented to by administrators on their behalf, but they will not be allowed to consent to new permissions or to new apps on their own. Only users who have been granted a directory role that includes the permission to grant consent will be able to consent to new apps.
- **Users can consent to apps from verified publisher⁴s or your organization, but only for permissions you choose** – All users can only consent to apps that were published by a verified publisher and apps that are registered in your tenant. Users can only consent to the permissions you have classified as “low impact”. You must **classify permissions⁵** to choose which permissions users are allowed to consent to.
- **Users can consent to all apps** – This option allows all users to consent to any permission that does not require administrator consent for any application.
- **Custom app consent policy** – For even more options over the conditions governing when users consent, you can **create custom app consent policies⁶** and configure those to apply for user consent.

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 - Enable the admin consent workflow⁷ to allow users to request an administrator's review and approval of an application that the user is not allowed to consent to—for example, when user consent has been disabled or when an application is requesting permissions that the user is not allowed to grant.

Risk-based step-up consent

Risk-based step-up consent helps reduce user exposure to malicious apps that make illicit consent requests. If Microsoft detects a risky end-user consent request, the request will require a step-up to admin consent instead. This capability is enabled by default, but it will only result in a behavior change when end-user consent is enabled.

When a risky consent request is detected, the consent prompt will display a message indicating that admin approval is needed. If the admin consent request workflow is enabled, the user can send the

⁴ <https://docs.microsoft.com/azure/active-directory/develop/publisher-verification-overview>

⁵ <https://docs.microsoft.com/azure/active-directory/manage-apps/configure-permission-classifications>

⁶ <https://docs.microsoft.com/azure/active-directory/manage-apps/manage-app-consent-policies>

⁷ <https://docs.microsoft.com/azure/active-directory/manage-apps/configure-admin-consent-workflow>

request to an administrator for further review directly from the consent prompt. If it is not enabled, the following message will be displayed:

- **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.

In this case, an audit event will also be logged with a Category of **ApplicationManagement**, an Activity Type of **Consent to application**, and a Status Reason of **Risky application detected**.

Important - Administrators should evaluate all consent requests carefully before approving a request, especially when Microsoft has detected risk.

Integrate On-premises Apps using Azure AD App Proxy

What is Application Proxy?

Application Proxy is a feature of Azure AD that enables users to access on-premises web applications from a remote client. Application Proxy includes both the Application Proxy service that runs in the cloud, and the Application Proxy connector that runs on an on-premises server. Azure AD, the Application Proxy service, and the Application Proxy connector work together to securely pass the user sign-on token from Azure AD to the web application.

The Application Proxy for Azure AD provides secure remote access to on-premises web applications. After a single sign-on to Azure AD, users can access both cloud and on-premises applications through an external URL or an internal application portal. For example, Application Proxy can provide remote access and single sign-on to Remote Desktop, SharePoint, Teams, Tableau, Qlik, and line of business (LOB) applications.

Application Proxy works with:

- Web applications that use **Integrated Windows Authentication**⁸ for authentication.
- Web applications that use form-based or **header-based**⁹ access.
- Web APIs that you want to expose to rich applications on different devices.
- Applications hosted behind a **Remote Desktop Gateway**¹⁰.
- Rich client apps that are integrated with the Microsoft Authentication Library (MSAL).

Application Proxy is recommended for giving remote users access to internal resources. Application Proxy replaces the need for a virtual private network (VPN) or reverse proxy. It is not intended for internal users on the corporate network. These users who unnecessarily use Application Proxy can introduce unexpected and undesirable performance issues.

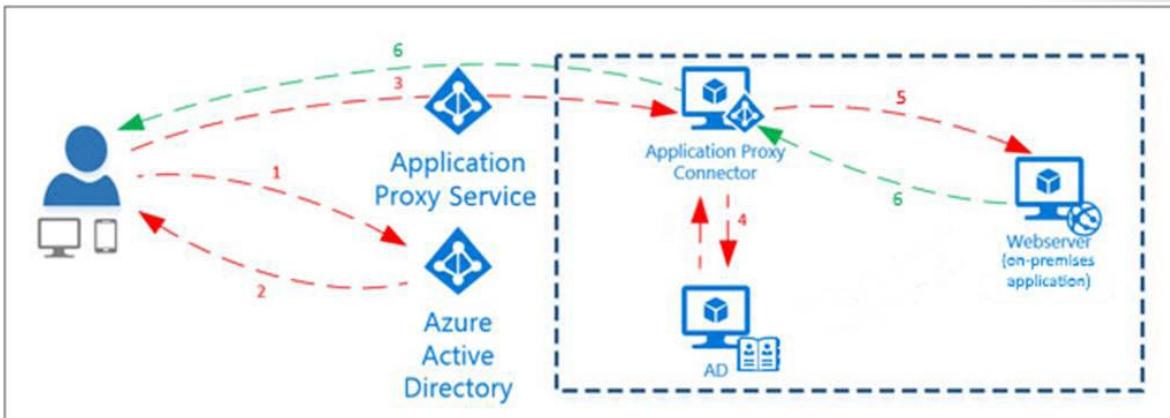
How Application Proxy works

The following diagram shows how Azure AD and Application Proxy work together to provide single sign-on to on-premises applications.

⁸ <https://docs.microsoft.com/azure/active-directory/manage-apps/application-proxy-configure-single-sign-on-with-kcd>

⁹ <https://docs.microsoft.com/azure/active-directory/manage-apps/application-proxy-configure-single-sign-on-with-headers>

¹⁰ <https://docs.microsoft.com/azure/active-directory/manage-apps/application-proxy-integrate-with-remote-desktop-services>

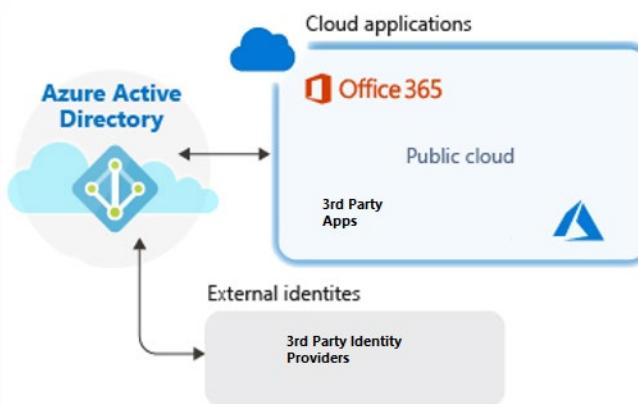


1. After the user has accessed the application through an endpoint, the user is directed to the Azure AD sign-in page.
2. After a successful sign-in, Azure AD sends a token to the user's client device.
3. The client sends the token to the Application Proxy service, which retrieves the user principal name (UPN) and security principal name (SPN) from the token. Application Proxy then sends the request to the Application Proxy connector.
4. If you have configured single sign-on, the connector performs any additional authentication required on behalf of the user.
5. The connector sends the request to the on-premises application.
6. The response is sent through the connector and Application Proxy service to the user.

Add an on-premises application for remote access through Application Proxy in Azure Active Directory

Click and interact with this Interactive Guide to learn more about enabling integrated windows authentication to on-premises applications with Azure AD Application Proxy	Enable Integrated Windows Authentication Interactive Guide (https://mslearn.cloudguides.com/guides/Provide%20secure%20remote%20access%20to%20on-premises%20applications%20with%20Azure%20AD%20Application%20Proxy)
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Integrate Custom SaaS with Single Sign-On



- You can use Azure AD as your identity system for just about any app. Many apps are already pre-configured and can be set up with minimal effort. These pre-configured apps are published in the Azure AD App Gallery.
- You can manually configure most apps for single sign-on if they aren't already in the gallery. Azure AD provides several SSO options. SAML-based SSO and OIDC-based SSO.

Apps can delegate maintenance of their own username and password information to a centralized identity provider. Delegating authentication and authorization to it enables scenarios such as Conditional Access policies that require a user to be in a specific location. The use of **multi-factor authentication**¹¹ (sometimes referred to as two-factor authentication or 2FA), enables a user to sign in once and then be automatically signed in to all of the web apps that share the same centralized directory.

Microsoft identity platform simplifies authorization and authentication for application developers by providing identity as a service, with support for industry-standard protocols such as OAuth 2.0 and OpenID Connect, as well as open-source libraries for different platforms to help you start coding quickly. It allows developers to build applications that sign in all Microsoft identities, get tokens to call Microsoft Graph, other Microsoft APIs, or APIs that developers have built.

The following list is a brief comparison of the various protocols used by Microsoft identity platform.

- **OAuth versus OpenID Connect:** OAuth is used for authorization and OpenID Connect (OIDC) is used for authentication. OpenID Connect is built on top of OAuth 2.0, which means the terminology and flow are similar between the two. You can even authenticate a user using OpenID Connect and get authorization to access a protected resource that the user owns using OAuth 2.0 in one request.
- **OAuth versus SAML:** OAuth is used for authorization and Security Assertion Markup Language (SAML) is used for authentication.
- **OpenID Connect versus SAML:** Both OpenID Connect and SAML are used to authenticate a user and are used to enable single sign-on. SAML authentication is commonly used with identity providers such as Active Directory Federation Services (ADFS) federated to Azure AD and is therefore frequently used in enterprise applications. OpenID Connect is commonly used for apps that are purely in the cloud, such as mobile apps, web sites, and web APIs.

If you have an application that you want to integrate with AzureAD providing the single sign-on experience for your users, please see the article ClaimsXRay in Azure AD with Directory Extension, linked below:

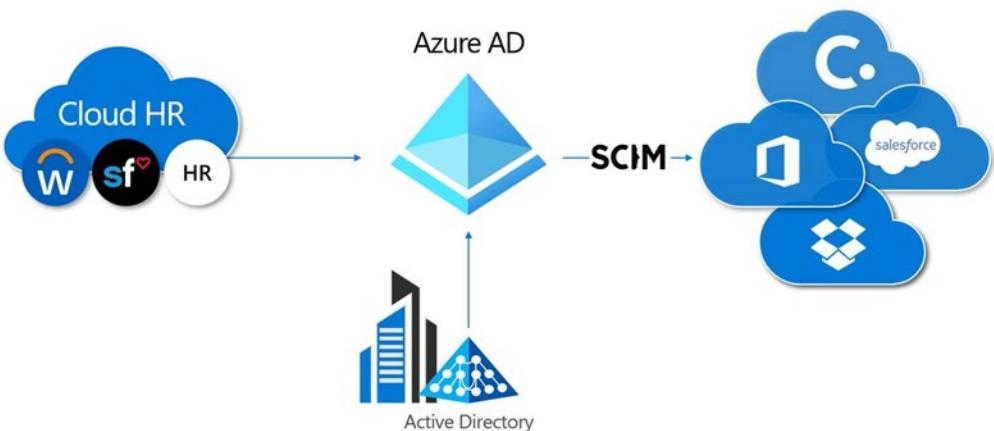
ClaimsXRay in Azure AD with Directory Extension¹²

¹¹ <https://docs.microsoft.com/azure/active-directory/authentication/concept-mfa-howitworks>

¹² <https://techcommunity.microsoft.com/t5/core-infrastructure-and-security/claimsxray-in-azuread-with-directory-extension/ba-p/1505737>

Implement Application User Provisioning

In Azure AD, the term app provisioning refers to automatically creating user identities and roles in the cloud (**SaaS¹³**) applications that users need access to. In addition to creating user identities, automatic provisioning includes the maintenance and removal of user identities as status or roles change. A common scenario is provisioning an Azure AD user into applications like **Dropbox¹⁴**, **Salesforce¹⁵**, **ServiceNow¹⁶**, and more.



This feature lets you to the following actions.

- **Automate provisioning** - Automatically create new accounts in the right systems for new people when they join a team or organization.
- **Automate deprovisioning** - Automatically deactivate accounts in the right systems when people leave a team or organization.
- **Synchronize data between systems** - Ensure that the identities in the apps and systems are kept up to date based on changes in the directory or the human resources system.
- **Provision groups** - Provision groups to applications that support them.
- **Govern access** - Monitor and audit who has been provisioned into the applications.
- **Seamlessly deploy in brown field scenarios** - Match existing identities between systems and allow for easy integration, even when users already exist in the target system.
- **Use rich customization** - Take advantage of customizable attribute mappings that define what user data should flow from the source system to the target system.
- **Get alerts for critical events** - The provisioning service provides alerts for critical events and allows for Log Analytics integration where you can define custom alerts to suite your business needs.

¹³ <https://azure.microsoft.com/overview/what-is-saas/>

¹⁴ <https://docs.microsoft.com/azure/active-directory/saas-apps/dropboxforbusiness-provisioning-tutorial>

¹⁵ <https://docs.microsoft.com/azure/active-directory/saas-apps/salesforce-provisioning-tutorial>

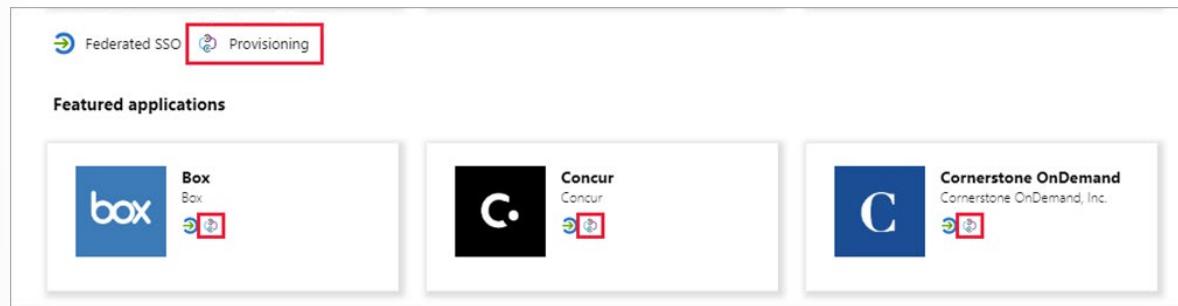
¹⁶ <https://docs.microsoft.com/azure/active-directory/saas-apps/servicenow-provisioning-tutorial>

Manual versus Automatic provisioning

Applications in the Azure AD gallery support either manual or automatic provisioning.

- Manual provisioning means there is no automatic Azure AD provisioning connector for the app yet. User accounts must be created manually. Examples of this include adding users directly into the administrative portal of the app or uploading a spreadsheet with user account details. Consult the documentation provided by the app or contact the app developer to determine what mechanisms are available.
- Automatic means that an Azure AD provisioning connector has been developed for this application. Follow the setup tutorial for setting up provisioning for the application.

In the Azure AD gallery, applications that support automatic provisioning are designated by a **Provisioning** icon.



The provisioning mode supported by an application is also visible on the **Provisioning** tab once you have added the application to your **Enterprise apps**.

System for Cross-domain Identity Management

To help automate provisioning and deprovisioning, apps expose proprietary user and group APIs. However, every app tries to perform the same simple actions, such as creating or updating users, adding users to groups, or deprovisioning users. Yet, all these simple actions are implemented just a little bit differently, using different endpoint paths, different methods to specify user information, and a different schema to represent each element of information.

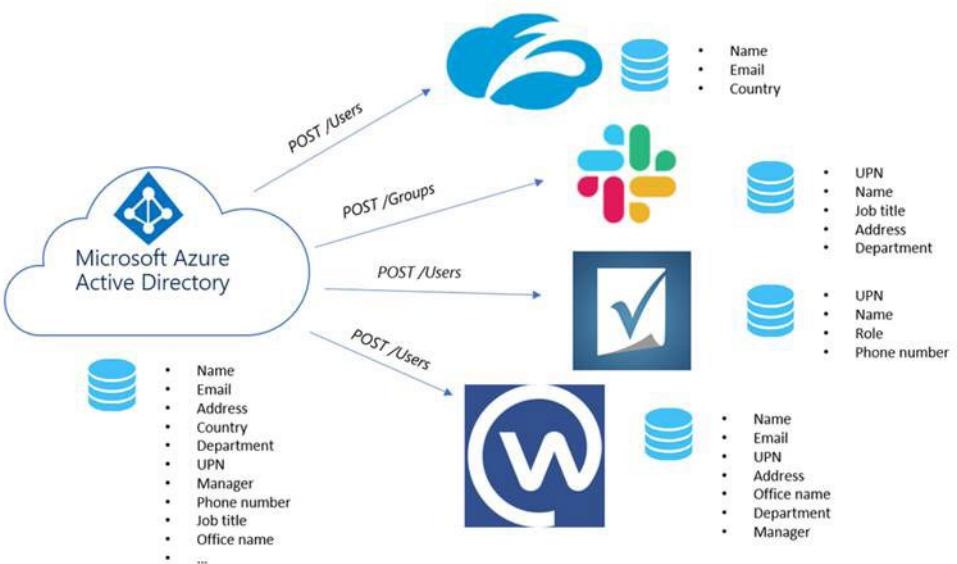
To address these challenges, the System for Cross-domain Identity Management (SCIM) specification provides a common user schema to help users move into, out of, and around apps. SCIM is becoming the standard for provisioning and, when used in conjunction with federation standards like SAML or OpenID Connect, provides administrators an end-to-end, standards-based solution for access management.

Build a System for Cross-domain Identity Management endpoint and configure user provisioning with Azure AD

As an application developer, you can use the System for Cross-Domain Identity Management (SCIM) user management API to enable automatic provisioning of users and groups between your application and Azure AD. The SCIM specification provides a common user schema for provisioning. When used in conjunction with federation standards like SAML or OpenID Connect, SCIM gives administrators an end-to-end, standards-based solution for access management.

SCIM is a standardized definition of two endpoints: a /Users endpoint and a /Groups endpoint. It uses common Representational state transfer (REST) verbs to create, update, and delete objects, and a pre-defined schema for common attributes like group name, username, first name, last name, and email. Apps

that offer a SCIM 2.0 REST API can reduce or eliminate the pain of working with a proprietary user management API. For example, any compliant SCIM client knows how to make an HTTP POST of a JSON object to the /Users endpoint to create a new user entry. Instead of needing a slightly different API for the same basic actions, apps that conform to the SCIM standard can instantly take advantage of pre-existing clients, tools, and code.



The standard user object schema and REST APIs for management defined in SCIM 2.0 allow identity providers and apps to integrate with each other more easily. Application developers that build a SCIM endpoint can integrate with any SCIM-compliant client without having to do custom work, rather than starting from scratch and building the implementation completely on your own, you can rely on a number of open source SCIM libraries published by the SCIM community.

Monitor and Audit Access to Azure AD Integrated Apps

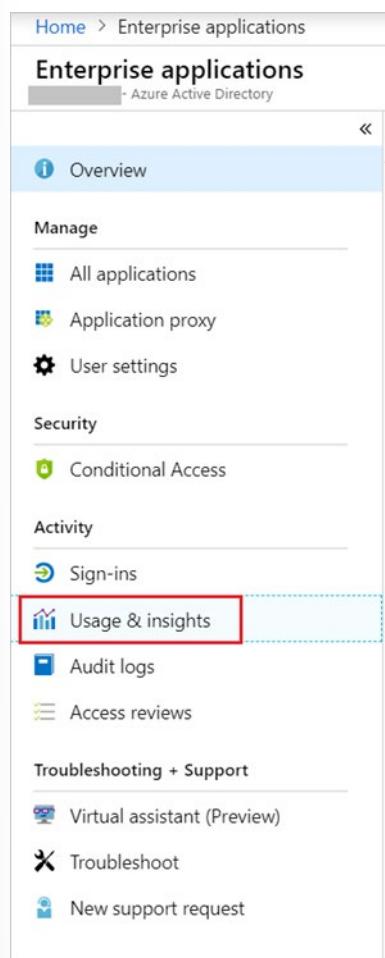
With Azure AD reports, you can get the information needed to determine how your environment is doing. With the usage and insights report, you can get an application-centric view of your sign-in data and find answers to the following questions:

- What are the top used applications in the organization?
- What applications have the most failed sign-ins?
- What are the top sign-in errors for each application?

Access the usage and insights report

1. Navigate to the [Azure portal](https://portal.azure.com/)¹⁷.
2. Select the correct directory, then select **Azure Active Directory** and choose **Enterprise applications**.
3. From the **Activity** section, select **Usage & insights** to open the report.

¹⁷ <https://portal.azure.com/>



Use the report

The usage and insights report shows the list of applications with one or more sign in attempts and allows you to sort by the number of successful sign-ins, failed sign-ins, and the success rate.

Selecting **load more** at the bottom of the list allows you to view additional applications on the page. You can select the date range to view all applications that have been used within the range.

You can also set the focus on a specific application. Select **view sign-in activity** to see the sign in activity over time for the application as well as the top errors.

When you select a day in the application usage graph, you get a detailed list of the sign-in activities for the application.

The screenshot shows the 'Usage & insights - Application activity' section of the Azure portal. It displays a list of active applications with their sign-in statistics. The applications listed are Graph explorer, Azure Portal, Office 365 SharePoint Online, and Azure DevOps. Each application row includes a 'View sign in activity' link, which is highlighted with a red box.

APPLICATION NAME	SUCCESSFUL SIGN-INS	FAILED SIGN-INS	SUCCESS RATE
Graph explorer	314	2	99.37%
Azure Portal	977	422	69.84%
Office 365 SharePoint Online	32	1	96.97%
Azure DevOps	13	0	100.00%

Audit logs

The Azure AD audit logs provide records of system activities for compliance. Users in the Security Administrator, Security Reader, Report Reader, Global Reader or Global Administrator roles can access their data. To access the audit report, select **Audit logs** in the **Monitoring** section of **Azure Active Directory**.

An audit log has a default list view that shows:

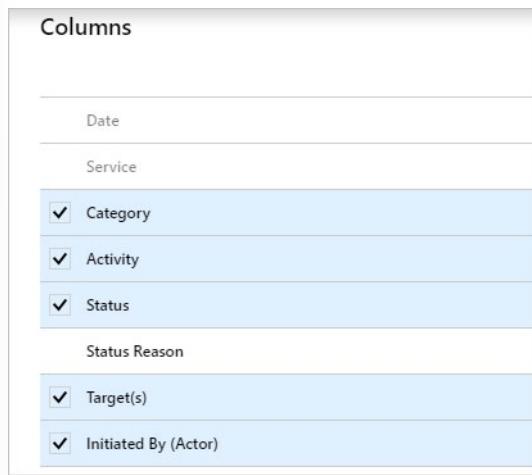
- the date and time of the occurrence
- the service that logged the occurrence
- the category and name of the activity (what)
- the status of the activity (success or failure)
- the target
- the initiator/actor (who) of an activity

DATE	SERVICE	CATEGORY	ACTIVITY	STATUS	TARGET(S)	INITIATED BY (ACTOR)
2/7/2019, 2:16:33 AM	Core Directory	Policy	Delete policy	Success	MFA Registration	admin@aad171.ccscstp.net
2/7/2019, 2:16:33 AM	Core Directory	Policy	Delete policy	Success	02/07/2019 10:15 AM	admin@aad171.ccscstp.net
2/7/2019, 2:15:56 AM	Identity Protection	Policy	Set MFA registration policy	Failure	Test_Test_aad171	admin@aad171.ccscstp.net
2/7/2019, 2:15:56 AM	Core Directory	Policy	Add policy	Failure	MFA Registration	Azure AD Identity Protection

You can customize the list view by clicking **Columns** in the toolbar.



This enables you to display additional fields or remove fields that are already displayed.



Select an item in the list view to get more detailed information.

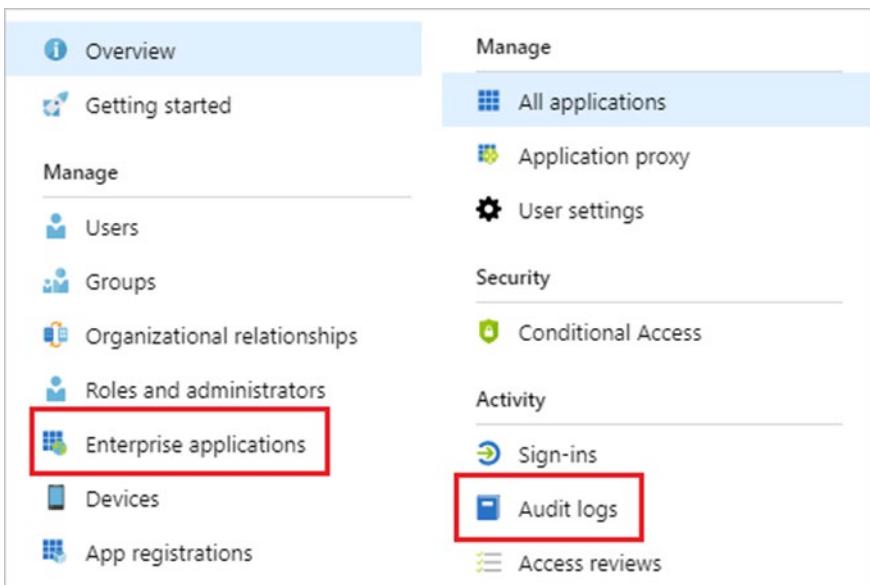
Activity	Target(s)	Modified Properties	INITIATED BY (ACTOR)	ADDITIONAL DETAILS
DATE	2/7/2019, 2:15:54 AM		TYPE	User
ACTIVITY TYPE	Set MFA registration policy		DISPLAY NAME	
CORRELATION ID	5477a398-7be5-4827-bfcb-1e40fcdee234		OBJECT ID	d7cc485d-2c1b-422c-98fd-5ce52859a4a3
CATEGORY	Policy		USER PRINCIPAL NAME	
STATUS	Failure			
STATUS REASON	Set MFA registration policy failed.			

Enterprise applications audit logs

With application-based audit reports, you can get answers to questions such as:

- What applications have been added or updated?
- What applications have been removed?
- Has a service principal for an application changed?
- Have the names of applications been changed?
- Who gave consent to an application?

If you want to review audit data related to your applications, you can find a filtered view under **Audit logs** in the **Activity** section of the **Enterprise applications** blade. This entry point has **Enterprise applications** preselected as the **Application Type**.



Implement App Registration

Introduction

In this module, you will plan your line-of-business application registration strategy, implement application registrations, and configure application permissions.

Learning objectives

In this module, you will:

- Plan your line-of-business application registration strategy.
- Implement application registrations.
- Configure application permissions.

Prerequisites

none

Plan your Line-of-Business Application Registration Strategy

Why do applications integrate with Azure AD?

Add applications to Azure AD to leverage one or more of the services it provides, including:

- Application authentication and authorization.
- User authentication and authorization.
- Single sign-on (SSO) using federation or password.
- User provisioning and synchronization.
- Role-based access control: Use the directory to define application roles to perform role-based authorization checks in an application.
- OAuth authorization services: Used by Microsoft 365 and other Microsoft applications to authorize access to APIs/resources.
- Application publishing and proxy: Publish an application from a private network to the internet.
- Directory schema extension attributes: Extend the schema of service principal and user objects to store additional data in Azure AD.

There are two representations of applications in Azure AD: **application objects¹⁸** and service principals. The next two sections explain each, as well as how they interact with one another in the Azure portal.

¹⁸ <https://docs.microsoft.com/azure/active-directory/develop/app-objects-and-service-principals>

What are application objects and where do they come from?

You can manage application objects in the Azure portal through the App Registrations experience. Application objects define and describe the application to Azure AD, enabling Azure AD to know how to issue tokens to the application based on its settings. The application object will only exist in its home directory, even if it's a multi-tenant application supporting service principals in other directories. The application object may include any of the following (as well as additional information not mentioned here):

- Name, logo, and publisher
- Redirect URIs
- Secrets (symmetric and/or asymmetric keys used to authenticate the application)
- API dependencies (OAuth)
- Published APIs/resources/scopes (OAuth)
- App roles (RBAC)
- SSO metadata and configuration
- User provisioning metadata and configuration
- Proxy metadata and configuration

You can create application objects through multiple pathways, including:

- Application registrations in the Azure portal.
- Creating a new application using Visual Studio and configuring it to use Azure AD authentication.
- When an admin adds an application from the app gallery (which will also create a service principal).
- Using the Microsoft Graph API or PowerShell to create a new application.
- Many other pathways, including various developer experiences in Azure and in API explorer experiences across developer centers.

What are service principals and where do they come from?

You can manage service principals in the Azure portal through the Enterprise Applications experience. Service principals govern an application connecting to Azure AD and can be considered the instance of the application in your directory. Any given application can have at most one application object (which is registered in a "home" directory) and one or more service principal objects representing instances of the application in every directory in which it acts.

The service principal can include:

- A reference back to an application object through the application ID property.
- Records of local user and group application-role assignments.
- Records of local user and admin permissions granted to the application.
 - For example: permission for the application to access a particular user's email.
- Records of local policies including Conditional Access policy.

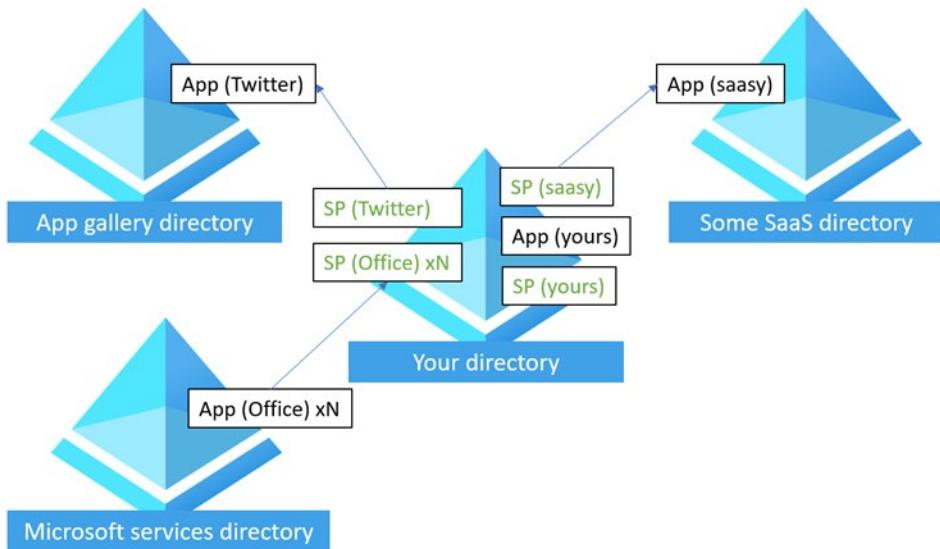
- Records of alternate local settings for an application.
 - Claims transformation rules.
 - Attribute mappings (User provisioning).
 - Directory-specific app roles (if the application supports custom roles).
 - Directory-specific name or logo.

Like application objects, service principals can be created through multiple pathways, including:

- When users sign in to a third-party application integrated with Azure AD.
 - During sign-in, users are asked to give permission to the application to access their profile and other permissions. The first person to give consent causes a service principal that represents the application to be added to the directory.
- When users sign in to Microsoft online services like Microsoft 365.
 - When you subscribe to Microsoft 365 or begin a trial, one or more service principals are created in the directory representing the various services that are used to deliver all of the functionality associated with Microsoft 365.
 - Some Microsoft 365 services, like SharePoint, create service principals on an ongoing basis to allow secure communication between components, including workflows.
- When an admin adds an application from the app gallery (this will also create an underlying app object).
- Add an application to use the Azure AD Application Proxy.
- Connect an application for SSO using SAML or password SSO.
- Programmatically via the Microsoft Graph API or PowerShell.

How are application objects and service principals related to each other?

An application has one application object in its home directory that's referenced by one or more service principals in each of the directories where it operates (including the application's home directory).



In the preceding diagram, Microsoft maintains two directories internally (shown on the left) that it uses to publish applications:

- One for Microsoft Apps (Microsoft services directory).
- One for pre-integrated third-party applications (App gallery directory).

Application publishers/vendors who integrate with Azure AD are required to have a publishing directory (shown on the right as "Some SaaS directory").

Applications that you add (represented as "App (yours)" in the diagram) include:

- Apps you developed (integrated with Azure AD).
- Apps you connected for SSO.
- Apps you published using the Azure AD Application Proxy.

Notes and exceptions to service principles

Not all service principals point back to an application object. When Azure AD was originally built, the services provided to applications were more limited, and the service principal was sufficient for establishing an application identity. The original service principal was closer in shape to the Windows Server Active Directory service account. For this reason, it's still possible to create service principals through different pathways, such as using Azure AD PowerShell, without first creating an application object. The Microsoft Graph API requires an application object before creating a service principal.

Not all of the information described above is currently exposed programmatically. The following are only available in the UI:

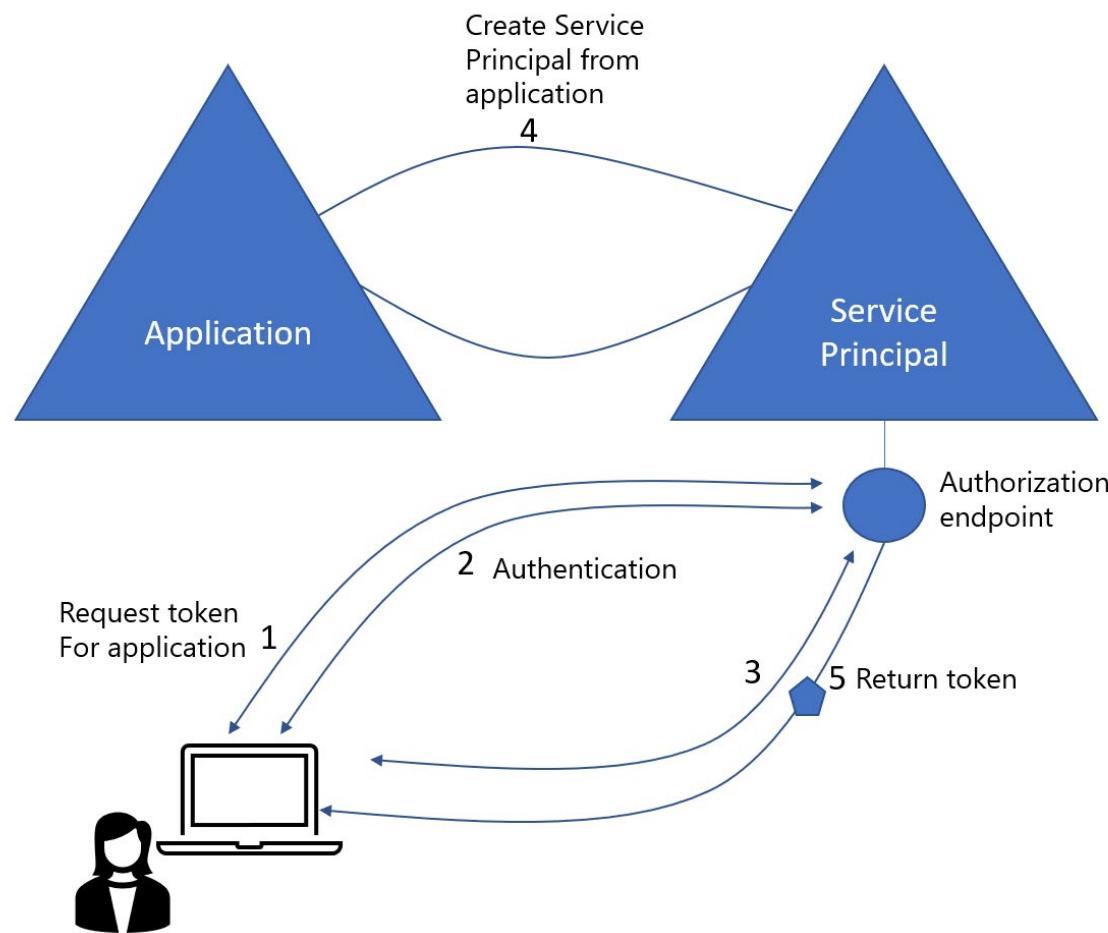
- Claims transformation rules
- Attribute mappings (User provisioning)

For more detailed information on the service principal and application objects, see the Microsoft Graph API reference documentation:

- Application

- Service Principal

Adding a new app registration



Process flow of the diagram

1. User requests to register an application – a request token is issued.
2. Authorization endpoint sends back an Authentication.
3. User consents to have the application registration.
4. Service is created from the application
5. Token returned to the user.

Who has permission to add applications to my Azure AD instance?

While there are some tasks that only Global Administrators can do (such as adding applications from the app gallery and configuring an application to use the Application Proxy), by default all users in your directory have rights to register application objects they are developing, and they have discretion over which applications they share / give access to their organizational data through consent. When the first

user in your directory signs in to an application and grants consent, that will create a service principal in your tenant; otherwise, the consent grant information will be stored on the existing service principal.

Allowing users to register and consent to applications might initially sound concerning, but keep the following in mind:

- Applications have been able to leverage Windows Server Active Directory for user authentication for many years without requiring the application to be registered or recorded in the directory. Now the organization will have improved visibility to exactly how many applications are using the directory and for what purpose.
- Delegating these responsibilities to users negates the need for an admin-driven application registration and publishing process. With Active Directory Federation Services (AD FS), an admin likely had to add an application as a relying party on behalf of their developers. Now developers can self-service.
- Users signing in to applications using their organization accounts for business purposes is a good thing. If they subsequently leave the organization, they will automatically lose access to their account in the application they were using.
- Having a record of what data was shared with which application is a good thing. Data is more transportable than ever and it's useful to have a clear record of who shared what data with which applications.
- API owners who use Azure AD for OAuth decide exactly what permissions users are able to grant to applications and which permissions require an admin to agree to. Only admins can consent to larger scopes and more significant permissions, while user consent is scoped to the users' own data and capabilities.
- When a user adds or allows an application to access their data, the event can be audited. You can view the Audit Reports within the Azure portal to determine how an application was added to the directory.

If you still want to prevent users in your directory from registering applications and from signing in to applications without administrator approval, two settings enable you to turn off those capabilities:

To prevent users from consenting to applications on their own behalf:

- In the Azure portal, go to the User settings section under Enterprise applications.
- Change **Users can consent to apps accessing company data on their behalf** to **No**.

Note - If you decide to turn off user consent, an admin will be required to consent to any new application a user needs to use.

To prevent users from registering their own applications:

- In the Azure portal, go to the User settings section under Azure Active Directory.
- Change **Users can register applications** to **No**.

Tenancy in Azure Active Directory

Azure Active Directory (Azure AD) organizes objects like users and apps into groups called *tenants*. Tenants enable an administrator to set policies on the users within the organization and the apps that the organization owns to meet their security and operational policies.

Who can sign in to your app?

When it comes to developing apps, developers can choose to configure their app to be either single-tenant or multi-tenant during app registration in the Azure portal.

- Single-tenant apps are only available in the tenant they were registered in, also known as their home tenant.
- Multi-tenant apps are available to users in both their home tenant and other tenants.

In the Azure portal, you can configure your app to be single-tenant or multi-tenant by setting the audience as follows:

Who can sign in to your app?

Audience	Single/multi-tenant	Who can sign in
Accounts in this directory only	Single tenant	All user and guest accounts in your directory can use your application or API. Use this option if your target audience is internal to your organization.
Accounts in any Azure AD directory	Multi-tenant	All users and guests with a work or school account from Microsoft can use your application or API. This includes schools and businesses that use Microsoft 365. Use this option if your target audience is business or educational customers.
Accounts in any Azure AD directory and personal Microsoft accounts (such as Skype, Xbox, Outlook.com)	Multi-tenant	All users with a work, school, or personal Microsoft account can use your application or API. It includes schools and businesses that use Microsoft 365, as well as personal accounts that are used to sign in to services like Xbox and Skype. Use this option to target the widest set of Microsoft accounts.

Best practices for multi-tenant apps

Building great multi-tenant apps can be challenging because of the number of different policies that IT administrators can set in their tenants. If you choose to build a multi-tenant app, follow these best practices:

- Test your app in a tenant that has configured Conditional Access policies.
- Follow the principle of least user access to ensure that your app only requests permissions it actually needs.
- Provide appropriate names and descriptions for any permissions you expose as part of your app. This helps users and admins know what they are agreeing to when they attempt to use your app's APIs. For more information, see the best practices section in the permissions guide.

Implement Application Registration

Each application you want the Microsoft identity platform to perform identity and access management (IAM) for must be registered. Register an app in the Azure portal so the Microsoft identity platform can provide authentication and authorization services for your application and its users. Whether it's a client application, like a web or mobile app, or a web API that backs a client app, registering it establishes a trust relationship between your application and the identity provider, the Microsoft identity platform.

Configure Application Permissions

Permissions and consent in the Microsoft identity platform endpoint

Applications that integrate with Microsoft identity platform follow an authorization model that gives users and administrators control over how data can be accessed. The implementation of the authorization model has been updated on the Microsoft identity platform endpoint, and it changes how an app must interact with the Microsoft identity platform. This unit covers the basic concepts of this authorization model, including scopes, permissions, and consent.

Scopes and permissions

The Microsoft identity platform implements the OAuth 2.0 authorization protocol, a method through which a third-party app can access web-hosted resources on behalf of a user. Any web-hosted resource that integrates with the Microsoft identity platform has a resource identifier, or **Application ID URI**. For example, Microsoft's web-hosted resources include:

- Microsoft Graph: <https://graph.microsoft.com>
- Microsoft 365 Mail API: <https://outlook.office.com>
- Azure Key Vault: <https://vault.azure.net>

The same is true for any third-party resources that have integrated with the Microsoft identity platform. Any of these resources also can define a set of permissions that can be used to divide the functionality of that resource into smaller chunks. As an example, Microsoft Graph has defined permissions for tasks such as:

- Read a user's calendar.
- Write to a user's calendar.
- Send mail as a user.

By defining these types of permissions, the resource has fine-grained control over its data and how API functionality is exposed. A third-party app can request these permissions from users and administrators who must approve the request before the app can access data or act on a user's behalf. By chunking the resource's functionality into smaller permission sets, developers can build third-party apps to request only the specific permissions that they need to perform their function. Users and administrators can know exactly what data the app has access to, and they can be more confident that it isn't behaving with malicious intent. Developers should always abide by the concept of least privilege, asking for only the permissions they need for their applications to function.

In OAuth 2.0, these types of permissions are called *scopes*. They are also often referred to as **permissions**. A permission is represented in the Microsoft identity platform as a string value. Continuing with the Microsoft Graph example, the string value for each permission is:

- Read a user's calendar by using Calendars.Read
- Write to a user's calendar by using Calendars.ReadWrite
- Send mail as a user using by Mail.Send

An app most commonly requests these permissions by specifying the scopes in requests to the Microsoft identity platform authorize endpoint. However, certain high-privilege permissions can only be granted through administrator consent and requested/granted using the administrator consent endpoint.

Permission types

Microsoft identity platform supports two types of permissions: **delegated permissions** and **application permissions**.

- **Delegated permissions** are used by apps that have a signed-in user present. For these apps, either the user or an administrator consents to the permissions that the app requests, and the app is delegated permission to act as the signed-in user when making calls to the target resource. Some delegated permissions can be consented to by non-administrative users, but some higher-privileged permissions require administrator consent. To learn which administrator roles can consent to delegated permissions, see Administrator role permissions in Azure AD.
- **Application permissions** are used by apps that run without a signed-in user present; for example, apps that run as background services or daemons. Only an administrator can consent to application permissions.

Effective permissions are those that your app will have when making requests to the target resource. It's important to understand the difference between the delegated and application permissions that your app is granted and its effective permissions when making calls to the target resource.

- For delegated permissions, the **effective permissions** of your app will be the least privileged intersection of the delegated permissions the app has been granted (via consent) and the privileges of the currently signed-in user. Your app can never have more privileges than the signed-in user. Within organizations, the privileges of the signed-in user may be determined by policy or by membership in one or more administrator roles. To learn which administrator roles can consent to delegated permissions, see Administrator role permissions in Azure AD.
- For example, assume your app has been granted the `User.ReadWrite.All` delegated permission. This permission nominally grants your app permission to read and update the profile of every user in an organization. If the signed-in user is a global administrator, your app will be able to update the profile of every user in the organization. However, if the signed-in user isn't in an administrator role, your app will be able to update only the profile of the signed-in user. It will not be able to update the profiles of other users in the organization, because the user whom it has permission to act on behalf of does not have those privileges.
- For application permissions, the **effective permissions** of your app will be the full level of privileges implied by the permission. For example, an app that has the `User.ReadWrite.All` application permission can update the profile of every user in the organization.

OpenID Connect Scopes

The Microsoft identity platform implementation of OpenID Connect has a few well-defined scopes that are also hosted on the Microsoft Graph: openid, email, profile, and offline_access. The address and phone OpenID Connect scopes are not supported.

Requesting the OIDC scopes and a token will give you a token to call the UserInfo endpoint.

OpenID

If an app performs sign-in by using OpenID Connect, it must request the openid scope. The openid scope shows on the work account consent page as the "Sign you in" permission and on the personal Microsoft account consent page as the "View your profile and connect to apps and services using your Microsoft account" permission. With this permission, an app can receive a unique identifier for the user in the form of the sub claim. It also gives the app access to the UserInfo endpoint. The openid scope can be used at the Microsoft identity platform token endpoint to acquire ID tokens, which can be used by the app for authentication.

Email

The email scope can be used with the openid scope and any others. It gives the app access to the user's primary email address in the form of the email claim. The email claim is included in a token only if an email address is associated with the user account, which isn't always the case. If it uses the email scope, your app should be prepared to handle a case in which the email claim does not exist in the token.

Profile

The profile scope can be used with the openid scope and any others. It gives the app access to a substantial amount of information about the user. The information it can access includes, but isn't limited to, the user's given name, surname, preferred username, and object ID. For a complete list of the profile claims available in the id_tokens parameter for a specific user, see the id_tokens reference.

Offline_access

The offline_access scope gives your app access to resources on behalf of the user for an extended time. On the consent page, this scope appears as the "Maintain access to data you have given it access to" permission. When a user approves the offline_access scope, your app can receive refresh tokens from the Microsoft identity platform token endpoint. Refresh tokens are long-lived. Your app can get new access tokens as older ones expire.

On the Microsoft identity platform (requests made to the v2.0 endpoint), your app must explicitly request the offline_access scope to receive refresh tokens. This means that when you redeem an authorization code in the OAuth 2.0 authorization code flow, you'll receive only an access token from the /token endpoint. The access token is valid for a short time, usually expiring in one hour. At that point, your app needs to redirect the user back to the /authorize endpoint to get a new authorization code. During this redirect, depending on the type of app, the user might need to enter their credentials again or consent again to permissions.

Note - This permission appears on all consent screens today, even for flows that don't provide a refresh token (the **implicit flow**¹⁹). This is to cover scenarios where a client can begin within the implicit flow, and then move on to the code flow where a refresh token is expected.

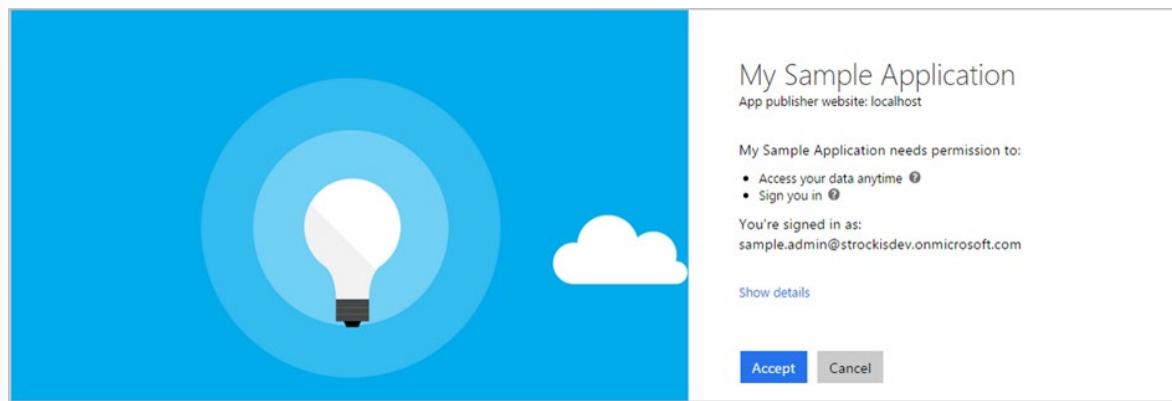
¹⁹ <https://docs.microsoft.com/azure/active-directory/develop/v2-oauth2-implicit-grant-flow>

Requesting individual user consent

In an OpenID Connect or OAuth 2.0 authorization request, an app can request the permissions it needs by using the scope query parameter. When a user signs into an app, the app sends a request for permission. The scope parameter is a space-separated list of delegated permissions that the app is requesting. Each permission is indicated by appending the permission value to the resource's identifier (the Application ID URI). In the request example, the app needs delegated permission to read the user's calendar and send mail as the user.

After the user enters their credentials, the Microsoft identity platform endpoint checks for a matching record of user consent. If the user has not consented to any of the requested permissions in the past, nor has an administrator consented to these permissions on behalf of the entire organization, the Microsoft identity platform endpoint asks the user to grant the requested permissions.

Note - At this time, the offline_access ("Maintain access to data you have given it access to") and user.read ("Sign you in and read your profile") permissions are automatically included in the initial consent to an application. These permissions are generally required for proper app functionality; offline_access gives the app access to refresh tokens, critical for native and web apps, while user.read gives access to the sub claim, allowing the client or app to correctly identify the user over time and access rudimentary user information.



When the user approves the permission request, consent is recorded, and the user doesn't have to consent again on subsequent sign-ins to the application.

Requesting consent for an entire tenant

Often, when an organization purchases a license or subscription for an application, the organization wants to proactively set up the application for use by all members of the organization. As part of this process, an administrator can grant consent for the application to act on behalf of any user in the tenant. If the admin grants consent for the entire tenant, the organization's users won't see a consent page for the application. Additionally, applications must use the admin consent endpoint to request application permissions.

Implement Application Authorization

Application roles

Application roles are used to assign permissions to users. You define app roles by using the Azure portal. When a user signs into the application, Azure AD emits a roles claim for each role that the user has been granted individually to the user and from their group membership.

There are two ways to declare app roles by using the Azure portal:

- App roles UI - Preview
- App manifest editor

Module 3 Review Questions

Module 3 Review Questions

Review Question 1

Which of the following directories is maintained by Microsoft and used to publish applications?

- SaaS directory
- Single sign-on app connected directory
- App gallery directory

Review Question 2

Which one of the following is a best practice for building multi-tenant apps?

- Follow the principle of least user access to ensure that your app only requests permissions it actually needs.
- Test your app in each tenant to ensure functionality.
- Use names and descriptions that are only meaningful to your team.

Review Question 3

Which two ways do you declare app roles by using the Azure portal?

- Certificates and secrets.
- Use the App manifest editor and API permissions.
- Use the App roles and App manifest editor.

Review Question 4

What service and connector work together to securely pass a user sign-on token from Azure AD to a web application running in an organization's on-premises datacenter?

- The Azure AD Application Proxy service and Application Proxy connector
- An Application Proxy connector and the Azure Firewall service
- The Azure AD Application Proxy service and Application Gateway

Review Question 5

Which user provision modes are supported for applications in the Azure AD gallery?

- Administrator approved and automatic.
- You should only use Manual Provisioning to ensure security.
- Manual and automatic

Review Question 6

Which of the following groups of information can be found in the Azure Active Directory Usage and insights report?

- The top used application in your organization -and- Who gave consent to an application -and- The top sign-in errors for each application
- The top used application in your organization and The applications with the most failed sign-ins and The service that logged the occurrence
- The top used applications in your organization and The application with the most failed sign-ins and The top sign-in errors for each application

Review Question 7

What is Microsoft's Cloud Access Security Broker solution?

- Microsoft Cloud App Security
- Microsoft Cloud Computing Services
- Microsoft Security Center

Review Question 8

By default, who has the ability to create application registrations or consent to applications in Azure Active Directory?

- All Azure AD Users
- All Azure AD and Guest users
- Only users assigned the Global Administrator role

Review Question 9

Which statement best describes the Cloud Application Administrator role?

- Users in this role have the same permissions as the Application Administrator role, excluding the ability to manage application proxy. Users assigned to this role are not added as owners when creating new application registrations or enterprise applications.
- Users in this role have the same permissions as the Application Management role, excluding the ability to manage application proxy. Users assigned to this role are not added as owners when creating new application registrations or enterprise applications.
- Users in this role have the same permissions as the Site Administrator role, including the ability to manage application proxy.

Module 3 Hands-on Exercises

Lab 20: Implement access management for apps

To download the most recent version of this lab, please visit the SC-300 [GitHub repository²⁰](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

Your organization requires that only specific users or groups have access to enterprise applications. You must assign a user to a specific application.

Objectives

After you complete this lab, you will be able to:

- Add an app to your Azure AD tenant
- Assign users to an app

Lab setup

- Estimated time: 5 minutes

Lab 21: Implement access management for apps

To download the most recent version of this lab, please visit the SC-300 [GitHub repository²¹](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

You need to create a new custom role for app management. This new role should be limited to only the specific permissions required to perform credential management.

Objectives

After you complete this lab, you will be able to:

- Create a new custom role to grant access to manage app registrations

²⁰ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

²¹ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

Lab setup

- Estimated time: 5 minutes

Lab 22: Register an application

To download the most recent version of this lab, please visit the SC-300 [GitHub repository²²](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

Registering your application establishes a trust relationship between your app and the Microsoft identity platform. The trust is unidirectional: Your app trusts the Microsoft identity platform—not the other way around.

Objectives

After you complete this lab, you will be able to:

- Register an application
- Add a redirect URI
- Add credentials
- Add a certificate
- Register the web API
- Add a scope
- Add a scope requiring admin consent
- Verify the exposed scopes
- Using the exposed scopes

Lab setup

- Estimated time: 20 minutes

Lab 23: Grant tenant-wide admin consent to an application

To download the most recent version of this lab, please visit the SC-300 [GitHub repository²³](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version

²² <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

²³ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

For applications your organization has developed or for those that are registered directly in your Azure AD tenant, you can grant tenant-wide admin consent from App registrations in the Azure portal.

Objectives

After you complete this lab, you will be able to:

- Grant admin consent in App registrations
- Grant admin consent in Enterprise apps

Lab setup

- Estimated time: 10 minutes

Lab 24: Add app roles to your app and receive them in the token

To download the most recent version of this lab, please visit the SC-300 [GitHub repository²⁴](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

Role-based access control (RBAC) is a popular mechanism to enforce authorization in applications. When using RBAC, an administrator grants permissions to roles, and not to individual users or groups. The administrator can then assign roles to different users and groups to control who has access to what content and functionality. You plan to implement RBAC roles and need to verify you understand how to perform the procedure.

Objectives

After you complete this lab, you will be able to:

- Declare app roles using the App roles UI
- Assign users and groups to roles

Lab setup

- Estimated time: 10 minutes

²⁴ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

Module 3 Summary

Summary for Module 2

During this module you saw how companies can use their Azure AD and on-premises AD systems to run their business and protect their users and resources. We saw how you can manage how your users authenticate and then use conditional access to ensure they have secure access only to the resources they need. Finally, we looked at how identity protection helps to protect your users.

Plan and Design the Integration of Enterprise Apps for Seamless Single Sign-on

During this lesson you explored how to:

- Discover apps by using MCAS or ADFS app report.
- Design and implement access management for apps.
- Design and implement app management roles.
- Configure pre-integrated (gallery) SaaS apps.

Implement and Monitor the Integration of Enterprise Apps for SSO

During this lesson you explored how to:

- Implement token customizations
- Implement and configure consent settings
- Integrate on-premises apps by using Azure AD application proxy
- Integrate custom SaaS apps for SSO
- Implement application user provisioning
- Monitor and audit access/Sign-On to Azure Active Directory integrated enterprise applications

Implement App Registration

During this lesson you explored how to:

- Plan your line-of-business application registration strategy.
- Implement application registrations.
- Configure application permissions.

Supplemental Resources

Use these resources to discover more:

- **ClaimsXRay in Azure AD with Directory Extension²⁵**

²⁵ <https://techcommunity.microsoft.com/t5/core-infrastructure-and-security/claimsxray-in-azuread-with-directory-extension/ba-p/1505737>

- **Configure authentication session management²⁶**

²⁶ <https://docs.microsoft.com/azure/active-directory/conditional-access/howto-conditional-access-session-lifetime>

Answers

Review Question 1

Which of the following directories is maintained by Microsoft and used to publish applications?

- SaaS directory
- Single sign-on app connected directory
- App gallery directory

Explanation

The two directories that Microsoft maintains are the App gallery directory and the Microsoft services directory.

Review Question 2

Which one of the following is a best practice for building multi-tenant apps?

- Follow the principle of least user access to ensure that your app only requests permissions it actually needs.
- Test your app in each tenant to ensure functionality.
- Use names and descriptions that are only meaningful to your team.

Explanation

Provide appropriate names and descriptions for any permissions you expose as part of your app. This helps users and admins know what they are agreeing to when they attempt to use your app's APIs.

Review Question 3

Which two ways do you declare app roles by using the Azure portal?

- Certificates and secrets.
- Use the App manifest editor and API permissions.
- Use the App roles and App manifest editor.

Explanation

Application roles are used to assign permissions to users. You define app roles by using the Azure portal. When a user signs into the application, Azure AD emits a roles claim for each role that the user has been granted individually to the user and from their group membership. The App roles and Manifest editor are found in the Manage section of the app.

Review Question 4

What service and connector work together to securely pass a user sign-on token from Azure AD to a web application running in an organization's on-premises datacenter?

- The Azure AD Application Proxy service and Application Proxy connector
- An Application Proxy connector and the Azure Firewall service
- The Azure AD Application Proxy service and Application Gateway

Explanation

The Azure AD Application Proxy service and the Application Proxy connector work together to securely pass the user sign-on token from Azure AD to the web application.

Review Question 5

Which user provision modes are supported for applications in the Azure AD gallery?

- Administrator approved and automatic.
- You should only use Manual Provisioning to ensure security.
- Manual and automatic

Explanation

Manual provisioning means there is no automatic Azure AD provisioning connector for the app yet. User accounts must be created manually, for example by adding users directly into the app's administrative portal or uploading a spreadsheet with user account detail. Consult the documentation provided by the app or contact the app developer to determine what mechanisms are available. Automatic means that an Azure AD provisioning connector has been developed for this application.

Review Question 6

Which of the following groups of information can be found in the Azure Active Directory Usage and insights report?

- The top used application in your organization -and- Who gave consent to an application -and- The top sign-in errors for each application
- The top used application in your organization and The applications with the most failed sign-ins and The service that logged the occurrence
- The top used applications in your organization and The application with the most failed sign-ins and The top sign-in errors for each application

Explanation

With the usage and insights report, you can get an application-centric view of your sign-in data to find information about these three topics.

Review Question 7

What is Microsoft's Cloud Access Security Broker solution?

- Microsoft Cloud App Security
- Microsoft Cloud Computing Services
- Microsoft Security Center

Explanation

Microsoft Cloud App Security is a Cloud Access Security Broker (CASB) that supports various deployment modes including log collection, API connectors, and reverse proxy.

Review Question 8

By default, who has the ability to create application registrations or consent to applications in Azure Active Directory?

- All Azure AD Users
- All Azure AD and Guest users
- Only users assigned the Global Administrator role

Explanation

In Azure AD all users can register application registrations and manage all aspects of applications they create. Everyone also has the ability to consent to apps accessing company data on their behalf.

Review Question 9

Which statement best describes the Cloud Application Administrator role?

- Users in this role have the same permissions as the Application Administrator role, excluding the ability to manage application proxy. Users assigned to this role are not added as owners when creating new application registrations or enterprise applications.
- Users in this role have the same permissions as the Application Management role, excluding the ability to manage application proxy. Users assigned to this role are not added as owners when creating new application registrations or enterprise applications.
- Users in this role have the same permissions as the Site Administrator role, including the ability to manage application proxy.

Explanation

Users in this role have the same permissions as the Application Administrator role, excluding the ability to manage application proxy. Users assigned to this role are not added as owners when creating new application registrations or enterprise applications.

Module 4 Plan and Implement an Identity Governance Strategy

Learning Objectives

Learning Objectives

After completing this module, you'll be able to:

- Plan and implement entitlement management
 - define catalogs
 - define access packages
 - plan, implement and manage entitlements
 - implement and manage terms of use
 - manage the lifecycle of external users in Azure AD Identity Governance settings
- Plan, implement, and manage access reviews
 - plan for access reviews□ create access reviews for groups and apps
 - monitor access review findings
 - manage licenses for access reviews
 - automate access review management tasks
 - configure recurring access reviews
- Plan and implement privileged access
 - define a privileged access strategy for administrative users (resources, roles, approvals, thresholds)
 - configure Privileged Identity Management for Azure AD roles
 - configure Privileged Identity Management for Azure resources

- assign roles
- manage PIM requests
- analyze PIM audit history and reports
- create and manage break-glass accounts
- Monitor and maintain Azure Active Directory
 - analyze and investigate sign-in logs to troubleshoot access issues
 - review and monitor Azure AD audit logs
 - enable and integrate Azure AD diagnostic logs with Log Analytics / Azure Sentinel
 - export sign-in and audit logs to a third-party SIEM
 - review Azure AD activity by using Log Analytics / Azure Sentinel, excluding KQL use
 - analyze Azure Active Directory workbooks / reporting
 - configure notifications

Plan and Implement Entitlement Management

Introduction

When new users or external users join your site, quickly assigning the access to Azure solutions is a must. Explore how to entitle users to access your site and resources. In this module, you will learn how to provide the appropriate access to your users, create reviews for that access, and more.

Learning objectives

By the end of this module, you will be able to:

- Define catalogs.
- Define access packages.
- Plan, implement and manage entitlements.
- Implement and manage terms of use.
- Manage the lifecycle of external users in Azure AD Identity Governance settings.

Prerequisites

None

Define Access Packages

Why use entitlement management?

Enterprise organizations often face challenges when managing employee access to resources such as:

- Users may not know what access they should have, and even if they do, they may have difficulty locating the right individuals to approve their access
- Once users find and receive access to a resource, they may hold on to access longer than is required for business purposes

These problems are compounded for users who need access from another organization, such as external users who are from supply chain organizations or other business partners. For example, Azure AD entitlement management can help organizations ensure that everyone has access to the correct directories and that all user access is managed consistently.

This video provides an overview of entitlement management and its value:

	Watch this video to learn more about Azure Active Directory entitlement management
-----------------------------------------------------------------------------------	------------------------------------------------------------------------------------



<https://www.microsoft.com/videoplayer/embed/RE4MFib>

What can I do with entitlement management?

Capabilities of entitlement management include:

Action	Outcome
Delegate to non-administrators the ability to create access packages.	These access packages contain resources that users can request, and the delegated access package managers can define policies with rules for which users can request, who must approve their access, and when access expires.
Select connected organizations whose users can request access.	When a user who is not yet in your directory requests access and is approved, they are automatically invited into your directory and assigned access. When their access expires, if they have no other access package assignments, their B2B account in your directory can be automatically removed.

Summary of terminology

Before exploring entitlement management and its documentation in depth, you should know the terms below. Feel free to reference back to this list at any time during this course.

Term	Description
access package	A bundle of resources that a team or project needs and is governed with policies. An access package is always contained in a catalog. You would create a new access package for a scenario in which users need to request access.
access request	A request to access the resources in an access package. A request typically goes through an approval workflow. If approved, the requesting user receives an access package assignment.
assignment	An assignment of an access package to a user ensures the user has all the resource roles of that access package. Access package assignments typically have a time limit before they expire.
catalog	A container of related resources and access packages. Catalogs are used for delegation so non-administrators can create their own access packages. Catalog owners can add resources they own to a catalog.
catalog creator	A collection of users who are authorized to create new catalogs. When a non-administrator user who is authorized to be a catalog creator creates a new catalog, they automatically become the owner of that catalog.
connected organization	An external Azure AD directory or domain that you have a relationship with. The users from a connected organization can be specified in a policy as being allowed to request access.
policy	A set of rules that defines the access lifecycle, such as how users get access, who can approve, and how long users have access through an assignment. A policy is linked to an access package. For example, an access package could have two policies: one for employees to request access and a second for external users to request access.
resource	An asset, such as an Office group, a security group, an application, or a SharePoint Online site, with a role that a user can be granted permissions to.
resource directory	A directory that has one or more resources to share.
resource role	A collection of permissions associated with and defined by a resource. A group has two roles: member and owner. SharePoint sites typically have three roles but may have additional custom roles. Applications can have custom roles.

What are access packages and what resources can I manage with them?

Entitlement management introduces to Azure AD the concept of an **access package**. An access package is a bundle of all the resources with the access a user needs to work on a project or perform their task. Access packages are used to govern access for your internal employees and users outside your organization. You can manage user access to the following resources with entitlement management:

- Membership of Azure AD security groups.
- Membership of Microsoft 365 Groups and Teams.
- Assignment to Azure AD enterprise applications, including SaaS applications and custom-integrated applications that support federation/single sign-on and/or provisioning.
- Membership of SharePoint Online sites.

You can also control access to other resources that rely upon Azure AD security groups or Microsoft 365 Groups. For example, you can provide:

- Licenses for Microsoft 365 by using an Azure AD security group in an access package and configuring group-based licensing for that group.
- Access to manage Azure resources by using an Azure AD security group in an access package and creating an Azure role assignment for that group.
- Access to manage Azure AD roles by using groups assignable to Azure AD roles in an access package and assigning an Azure AD role to that group.

How do I control who gets access?

With an **access package**, an administrator or delegated access package manager lists the resources (groups, apps, and sites) and the roles the users need for those resources.

Access packages also include one or more *policies*. A policy defines the rules or guardrails for assignment to access package. Each policy can be used to ensure that only the appropriate users are able to request access, that there are approvers for their request, and that their access to those resources is time-limited and will expire if not renewed.



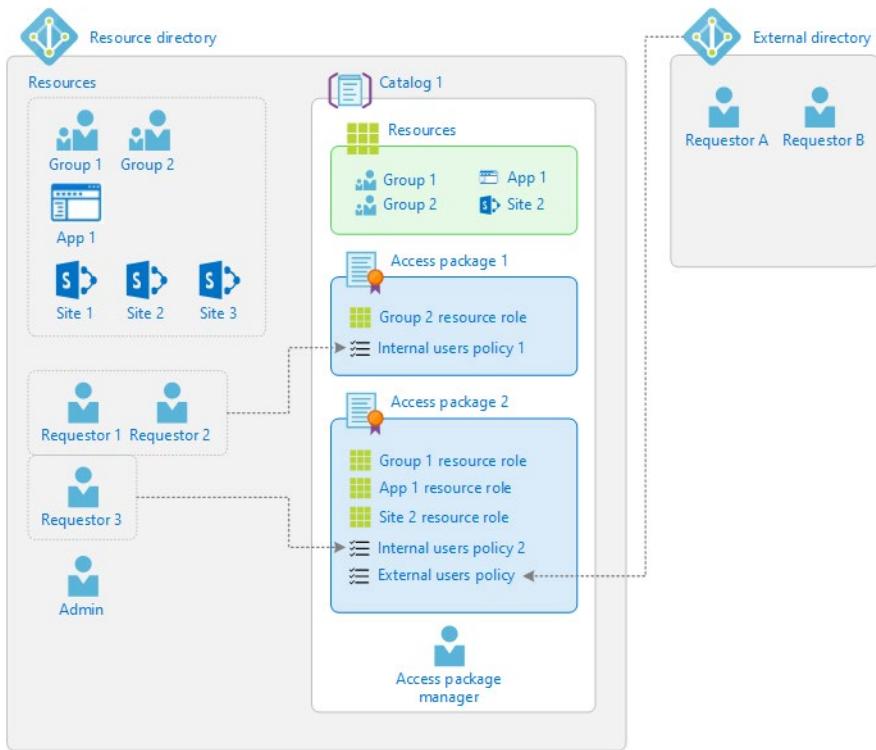
Within each policy, an administrator or access package manager defines the already existing users who are eligible to request access, the process to approve or deny access, and the duration of a user's access.

When should I use access packages?

Access packages do not replace other mechanisms for access assignment. They are most appropriate in situations such as when:

- Employees need time-limited access for a particular task. For example, you might use group-based licensing and a dynamic group to ensure all employees have an Exchange Online mailbox, and then use access packages for situations in which employees need additional access, such as to read departmental resources from another department.
- Access requires the approval of an employee's manager or other designated individuals.
- Departments wish to manage their own access policies for their resources without IT involvement.
- Two or more organizations are collaborating on a project, and as a result, multiple users from one organization will need to be brought in via Azure AD B2B to access another organization's resources.

The following diagram shows an example of the elements in entitlement management:



In **Access package 1**, there is only one single group as a resource. Access is defined with a policy that enables a set of users in the directory to request access. **Access package 2** includes a group, an application, and a SharePoint Online site as resources. Access is defined with two different policies. The first policy enables a set of users in the directory to request access. The second policy enables users in an external directory to request access.

Configure Entitlement Management

There are several ways that you can configure entitlement management for your organization. However, if you're just getting started, it's helpful to understand the common scenarios for administrators, catalog owners, access package managers, approvers, and requestors.

- Delegate
 - Administrator: Delegate management of resources.
 - Catalog creator: Delegate management of resources.
 - Catalog owner: Delegate management of resources.
 - Catalog owner: Delegate management of access packages.
- Govern access for users in your organization
 - Access package manager: Allow employees in your organization to request access to resources.
 - Requestor: Request access to resources.
 - Approver: Approve requests to resources.
 - Requestor: View the resources you already have access to.
- Govern access for users outside your organization
 - Administrator: Collaborate with an external partner organization.
 - Access package manager: Collaborate with an external partner organization.
 - Requestor: Request access to resources as an external user.
 - Approver: Approve requests to resources.
 - Requestor: View the resources you already have access to.
- Day-to-day management
 - Access package manager: Update the resources for a project.
 - Access package manager: Update the duration for a project.
 - Access package manager: Update how access is approved for a project.
 - Access package manager: Update the people for a project.
 - Access package manager: Directly assign specific users to an access package.
- Assignments and reports
 - Administrator: View who has assignments to an access package.
 - Administrator: View resources assigned to users.

Programmatic administration

You can also manage access packages, catalogs, policies, requests, and assignments using Microsoft Graph. A user in an appropriate role with an application that has the delegated `EntitlementManagement.ReadWrite.All` permission can call the **entitlement management API¹**.

¹ <https://docs.microsoft.com/graph/tutorial-access-package-api?view=graph-rest-beta>

Plan, Implement, and Manage Access Reviews

Introduction

Once identity is deployed, proper governance using access reviews is necessary for a secure solution. Explore how to plan for and implement access reviews.

In this module, you will learn all about access reviews, including why they're important to the security of your organization, how to prepare for and perform them, and how to configure them to occur on a recurring basis.

Learning objectives

By the end of this module you should be able to:

- Plan for access reviews.
- Create access reviews for groups and apps.
- Monitor access review findings.
- Manage licenses for access reviews.
- Automate access review management tasks.
- Configure recurring access reviews.

Prerequisites

None

Plan for Access Reviews

What is an access review?

An **Access Review** as the name implies, is a planned review of the access needs, rights, and history of user access. Access Reviews help users ensure that the right people have the right access to the right resources.

They mitigate access risk by protecting, monitoring, and auditing access to critical assets—while ensuring employee and business partner productivity. Finally, the access review is performed in Azure AD Identity Governance. An Azure AD premium P2 license is required.

Consider your organizational needs to determine the strategy for deploying access reviews in your environment.

Engage the right stakeholders

When technology projects fail, they typically do so due to mismatched expectations on impact, outcomes, and responsibilities. To avoid these pitfalls, ensure that you're engaging the right stakeholders and that project roles are clear. For access reviews, you will likely include representatives from the following teams within your organization:

- **IT administration** manages your IT infrastructure and administers your cloud investments and Software as a Service (SaaS) apps.

- **Development teams** build and maintain applications for your organization
- **Business units** manage projects and own applications.
- **Corporate governance** ensures that the organization is following internal policy and complying with regulations.

Note - For reviews requiring manual evaluations, be sure to plan for adequate reviewers and review cycles that meet your policy and compliance needs. If review cycles are too frequent, or there are too few reviewers, quality may be lost and too many or too few people may have access.

What is Azure AD Identity Governance?

Azure Active Directory (Azure AD) Identity Governance enables you to balance your organization's need for security and employee productivity with the right processes and visibility. It provides you with capabilities to ensure that the right people have the right access to the right resources. These and related Azure AD and Enterprise Mobility + Security features allows you to mitigate access risk by protecting, monitoring, and auditing access to critical assets—while ensuring employee and business partner productivity.

Identity Governance gives organizations the ability to complete tasks across employees, business partners and vendors, and across services and applications both on-premises and in clouds. Specifically, it is intended to help organizations address these four key questions:

- Which users should have access to which resources?
- What are those users doing with that access?
- Are there effective organizational controls for managing access?
- Can auditors verify that the controls are working?

Plan a pilot

We encourage customers to initially pilot access reviews with a small group and target non-critical resources. Piloting can help you adjust processes and communications as needed and increase users' and reviewers' ability to meet security and compliance requirements.

In your pilot, we recommend that you:

- Start with reviews where the results are not automatically applied, and you can control the implications.
- Ensure that all users have valid email addresses listed in Azure AD and that they receive email communication to take the appropriate action.
- Document any access removed as a part of the pilot in case you need to quickly restore it.
- Monitor audit logs to ensure all events are properly audited.

What resource types can be reviewed?

Once you integrate your organization's resources with Azure AD (such as users, applications, and groups), they can be managed and reviewed.

Typical targets for review include:

- User access to applications integrated with Azure AD for single sign-on (such as SaaS, line-of-business).

- Group membership (synchronized to Azure AD, or created in Azure AD or Microsoft 365, including Microsoft Teams).
- Access Package that groups resources (groups, apps, and sites) into a single package to manage access.
- Azure AD roles and Azure Resource roles as defined in Privileged Identity Management (PIM).

Who will create and manage access reviews?

The administrative role required to create, manage, or read an Access Review depends on the type of resource being reviewed.

Resource type	Create and manage access reviews (Creators)	Read Access Review results
Group or application	Global Administrator or User Administrator	Creators and Security Administrator
Privileged roles in Azure AD	Global Administrator or Privileged Role Administrator	Creators or Security Reader or Security Administrator
Privileged roles in Azure (resources)	Global Administrator or User Administrator Resource Owner	Creators
Access package	Global Administrator or Creator of Access Package	Global Administrator only

Who will review the access to the resource?

The creator of the access review decides at the time of creation who will perform the review. This setting can't be changed once the review is started. Reviewers are represented by three personas:

- Resource Owners, who are the business owners of the resource.
- A set of individually selected delegates, as selected by the access reviews administrator.
- End users who will each self-attest to their need for continued access.

When creating an Access Review, administrators can choose one or more reviewers. All reviewers can start and carry out a review, choosing to grant users continued access to a resource or removing them.

Components of an access review

Before implementing your access reviews, you should plan the types of reviews relevant to your organization. To do so, you will need to make business decisions about what you want to review and the actions to take based on those reviews.

To create an access review policy, you must have the following information.

- What resource(s) must be reviewed?
- Whose access is being reviewed?
- How often should the review occur?
- Who will perform the review?
- How will they be notified to review?
- What are the timelines to be enforced for review?

- What automatic actions should be enforced based on the review?
- What happens if the reviewer doesn't respond in time?
- What manual actions will be taken as a result based on the review?
- What communications should be sent based on actions taken?

Example Access Review plan

Component	Value
Resources to review	Access to Microsoft Dynamics
Review frequency	Monthly
Who performs review	Dynamics business group program managers
Notification	Email 24 hours prior to review to alias Dynamics-Pms and Include encouraging custom message to reviewers to secure their buy-in
Timeline	48 hours from notification
Automatic actions	Remove access from any account that has no interactive sign-in within 90 days by removing the user from the security group dynamics-access. and Perform actions if not reviewed within timeline.
Manual actions	Reviewers may perform removals approval prior to automated action if desired.
Communications	Send internal (member) users who are removed an email explaining they are removed and how to regain access.

Plan access reviews for access packages

Access packages can vastly simplify your governance and Access Review strategy. An Access Package is a bundle of all the resources with the access a user needs to work on a project or perform their task. For example, you may want to create an Access Package that includes all the applications that developers in your organization need, or all applications to which external users should have access. An administrator or delegated Access Package manager then groups the resources (groups or apps) and the roles the users need for those resources.

When creating an Access Package, you can create one or more access policies that set conditions for which users can request an Access Package, what the approval process looks like, and how often a person would have to re-request access. Access reviews are configured while creating or editing an Access Package policy.

Plan access reviews for groups

Besides Access Packages, reviewing group membership is the most effective way of governing access. We recommend that access to resources is assigned via security groups or Microsoft 365 groups, and that users are added to those groups to gain access.

A single group can be granted access to all appropriate resources. You can assign the group access to individual resources, or to an Access Package that groups applications and other resources. With this method, you can review access to the group rather than an individual's access to each application.

Group membership can be reviewed by:

- Administrators
- Group owners
- Selected users, delegated review capability when the review is created
- Members of the group, attesting for themselves

Group ownership

We recommend that group owners review membership, as they're best situated to know who needs access. Ownership of groups differs with the type of group.

- Groups that are created in Microsoft 365 and Azure AD have one or more well-defined owners. In most cases, these owners make perfect reviewers for their own groups as they know who should have access. For example, Microsoft Teams uses Microsoft 365 Groups as the underlying authorization model to grant users access to resources that are in SharePoint, Exchange, OneNote, or other Microsoft 365 services. The creator of the team automatically becomes an owner and should be responsible for attesting to the membership of that group.
- Groups created manually in the Azure AD portal or via scripting through Microsoft Graph may not necessarily have owners defined. We recommend that you define them either through the Azure AD Portal in the group's "Owners" section or via Graph.
- Groups that are synchronized from on-premises Active Directory cannot have an owner in Azure AD. When creating an Access Review for them, you should select individuals who are best suited to decide on membership in them.

Note - We recommend defining business policies that define how groups are created to ensure clear group ownership and accountability for regular review of membership.

Review membership of exclusion groups in CA policies

There are times when Conditional Access (CA) policies designed to keep your network secure shouldn't apply to all users. For example, a CA policy that only allows users to sign in while on the corporate network may not apply to the sales team, which travels extensively. In that case, the sales team members would be put into a group and that group would be excluded from the CA policy.

Review external users' group memberships

To minimize manual work and associated potential errors, consider using Dynamic Groups to assign group membership based on a user's attributes. You may want to create one or more Dynamic Groups for external users. The internal sponsor can act as a reviewer for membership in the group.

Review access to on-premises groups

Access reviews can't change the group membership of groups that you synchronize from on-premises with Azure AD Connect. This is because the source of authority is on-premises. You can still use access reviews to schedule and maintain regular reviews of on-premises groups. Reviewers will then take action in the on-premises group. This strategy keeps access reviews as the tool for all reviews. You can use the results from an Access Review on on-premises groups and process them further. The data is available in a CSV file or from Microsoft Graph.

Plan access reviews for applications

When you review access to an application, you're reviewing the access for employees and external identities to the information and data within the application. Choose to review an application when you need to know who has access to a specific application, instead of an Access Package or a group.

We recommend you plan reviews for applications in the following scenarios:

- Users are granted direct access to the application (outside of a group or Access Package).
- The application exposes critical or sensitive information.
- The application has specific compliance requirements to which you must attest.
- You suspect inappropriate access.

Reviewers for an application

Access reviews can be for the members of a group or for users who were assigned to an application. Applications in Azure AD don't necessarily have an owner, which is why the option for selecting the application owner as a reviewer isn't possible. You can further scope a review to review only guest users assigned to the application, rather than reviewing all access.

Plan review of Azure AD and Azure resource roles

Privileged Identity Management (PIM) simplifies how enterprises manage privileged access to resources in Azure AD. This keeps the list of privileged roles, both in Azure AD and Azure resources, much smaller and increases the overall security of the directory.

Access reviews allow reviewers to attest whether users still need to be in a role. Just like access reviews for Access Packages, reviews for Azure AD roles and Azure resource are integrated into the PIM admin user experience. We recommend you review the following role assignments regularly:

- Global Administrator
- User Administrator
- Privileged Authentication Administrator
- Conditional Access Administrator
- Security Administrator
- All Microsoft 365 and Dynamics Service Administration roles

Deploy access reviews

After you've prepared a strategy and a plan to review access for resources integrated with Azure AD, deploy and manage reviews by using the resources below.

Review access packages

To reduce the risk of stale access, administrators can enable periodic reviews of users who have active assignments to an access package. You can create access reviews, perform access reviews for others that are assigned to an Access Package, or perform a self-review of assigned Access Package(s).

Review groups and apps

Employees' and guests' access needs to groups and applications likely change over time. To reduce the risk associated with stale access assignments, administrators can create access reviews for group members or application access.

You can create access reviews for group members or application access, perform access reviews for members of a group or users with access to an application, allow members to review their own access to a group or an application, view access reviews, and take action for on-premises groups with PowerShell.

Review Azure AD roles

To reduce the risk associated with stale role assignments, you should regularly review access of privileged Azure AD roles.

Review Azure resource roles

To reduce the risk associated with stale role assignments, you should regularly review access of privileged Azure resource roles.

Use the access reviews API

The access reviews methods in the Microsoft Graph API are available for both application and user contexts. When running scripts in the application context, the account used to run the API (the service principle) must be granted the `AccessReview.Read.All` permission to query access reviews information.

Popular access reviews tasks to automate using the Graph API for access reviews are:

- Create and start an Access Review.
- Manually end an Access Review before its scheduled end.
- List all running Access Reviews and their status.
- See the history of a review series and the decisions and actions taken in each review.
- Collect decisions from an Access Review.
- Collect decisions from completed reviews where the reviewer took a different decision than what the system recommended.

Note - When creating new Graph API queries for automation, we recommend using the Graph Explorer. You can build and explore your Graph queries before putting them into scripts and code. This can help you quickly iterate your query so that you get exactly the results you're looking for, without changing the code of your script.

Monitor access reviews

Access reviews activities are recorded and available from the Azure AD audit logs. You can filter the audit data on the category, activity type, and date range. Here is a sample query:

Category	Policy
Activity type	Create access review
	Update access review

Category	Policy
	Access Review ended
	Delete access review
	Approve decision
	Deny decision
	Reset decision
	Apply decision
Date range	Seven days

For more advanced queries and analysis of access reviews, and to track changes and completion of reviews, we recommend you export your Azure AD Audit Logs to Azure Log Analytics or Azure Event Hub. When logs are stored in Azure Log Analytics, you can use the powerful analytics language and build your own dashboards.

Plan communications

Communication is critical to the success of any new business process. Proactively communicate to users how and when their experience will change and how to gain support if they experience issues.

Communicate changes in accountability: Access Reviews support shifting responsibility of reviewing and acting on continued access to business owners. Decoupling access decisions from IT drives more accurate access decisions. This is a cultural change in resource owners' accountability and responsibility. Proactively communicate this change and ensure resource owners are trained and able to use the insights to make good decisions.

Clearly, IT will want to stay in control for all infrastructure-related access decisions and privileged role assignments.

Customize email communication: When you schedule a review, you nominate users who will perform this review. These reviewers then receive an email notification of new reviews assigned to them, as well as reminders before a review assigned to them expires.

Administrators can choose to send this notification either halfway before the review expires or a day before it expires.

The email sent to reviewers can be customized to include a custom short message that encourages them to act on the review. We recommend you use the additional text to:

- Include a personal message to reviewers, so they understand it is sent by your Compliance or IT department.
- Include a hyperlink or reference to internal information on what the expectations of the review are and additional reference or training material.
- Include a link to instructions on how to perform a self-review of access.

Upon selecting Start review, reviewers will be directed to the MyAccess portal for group and application Access Reviews. The portal gives them an overview of all users who have access to the resource they're reviewing and system recommendations based on last sign-in and access information.

Create Access Review for Groups and Apps

Access to groups and applications for employees and guests changes over time. To reduce the risk associated with stale access assignments, administrators can use Azure Active Directory (Azure AD) to

create access reviews for group members or application access. If you need to routinely review access, you can also create recurring access reviews.

	Watch this video to learn more about how to deploy and create access reviews.
-----------------------------------------------------------------------------------	-------------------------------------------------------------------------------



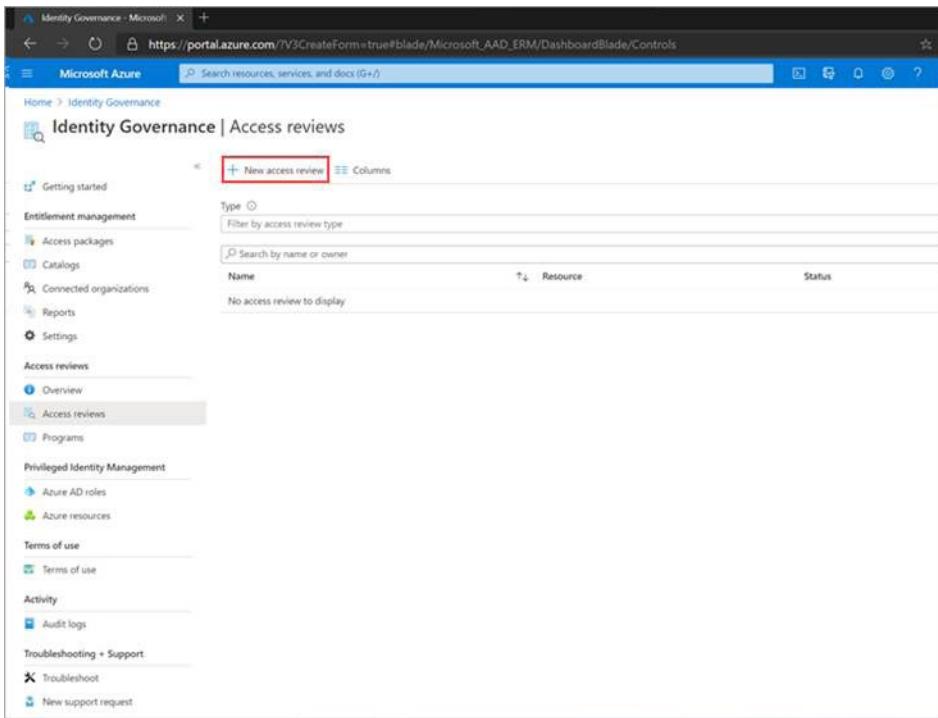
<https://www.microsoft.com/videoplayer/embed/RE4MAkF>

Prerequisites

- Azure AD Premium P2
- Global Administrator or User Administrator

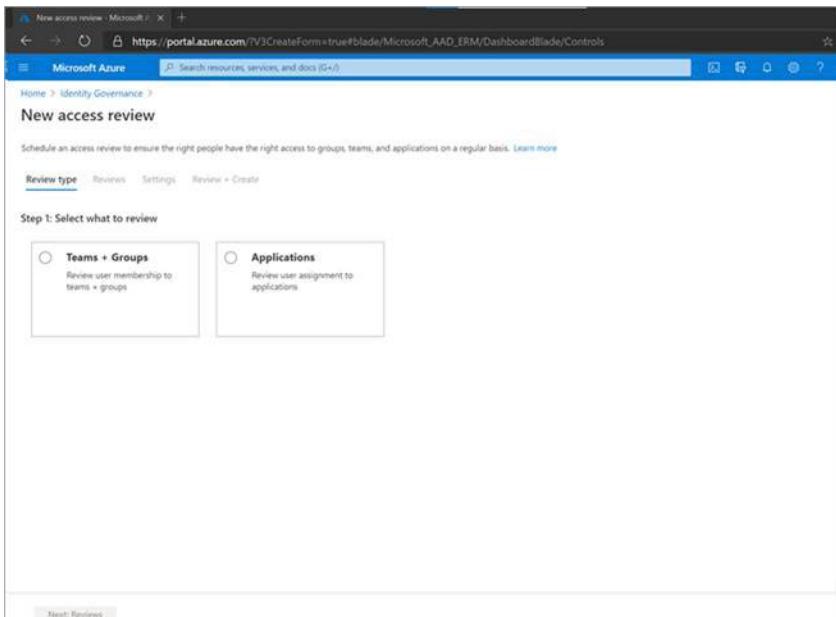
Create one or more access reviews

1. Sign in to the Azure portal and open the Identity Governance page.
2. In the left menu, click **Access reviews**.
3. Click **New access review** to create a new access review.



The screenshot shows the Microsoft Azure Identity Governance interface. On the left, there's a navigation sidebar with sections like 'Getting started', 'Entitlement management', 'Access packages', 'Catalogs', 'Connected organizations', 'Reports', 'Settings', 'Access reviews' (which is currently selected), 'Programs', 'Privileged Identity Management', 'Azure AD roles', 'Azure resources', 'Terms of use', 'Activity', 'Audit logs', 'Troubleshooting + Support', and 'New support request'. The main area is titled 'Identity Governance | Access reviews' and contains a table with columns 'Name', 'Resource', and 'Status'. A red box highlights the '+ New access review' button at the top left of the main content area.

4. In **Step 1: Select what to review** select the resource you would like to review.



The screenshot shows the 'New access review' setup page. At the top, there are tabs for 'Review type', 'Reviews', 'Settings', and 'Review + Create'. Below that, it says 'Step 1: Select what to review'. There are two options: 'Teams + Groups' (selected) and 'Applications'. Each option has a brief description. At the bottom, there's a 'Next: Review' button.

5. If you selected **Teams + Groups** in Step 1, you have two options in Step 2:

- **All Microsoft 365 groups with guest users.** Select this option if you would like to create recurring reviews on all your guest users across all your Microsoft Teams and M365 groups in your organization. You can choose to exclude certain groups by clicking on "Select group(s) to exclude."
- **Select teams + groups.** Select this option if you would like to specify a finite set of teams and/or groups to review. After clicking on this option, you will see a list of groups to the right to pick from.

The screenshot shows the Microsoft Azure portal with the URL https://portal.azure.com/#CreateForm=true#blade/Microsoft_AAD_ERM/DashboardBlade/Controls. The page title is "New access review". The top navigation bar includes "Microsoft Azure", a search bar, and various icons. Below the title, it says "Home > Identity Governance > New access review". A sub-header states: "Schedule an access review to ensure the right people have the right access to groups, teams, and applications on a regular basis. Learn more". Below this are four tabs: "Review type" (which is selected), "Reviews", "Settings", and "Review + Create".
Step 1: Select what to review
Two options are shown:

- Teams + Groups**
Review user membership to teams + groups
- Applications**
Review user assignment to applications

Step 2: Select which Teams + Groups
Two radio button options are available:

- All Microsoft 365 groups with guest users
- Select teams + groups

Below these options is a link: "Select group(s) to exclude".
Step 3: Select review scope
Two radio button options are available:

- Guest users only
- Everyone

Next: Reviews

The screenshot shows the Microsoft Azure portal with the same URL and title as the previous screenshot. The "Review type" tab is selected. The main content area is identical to the first screenshot.
Step 1: Select what to review
The "Teams + Groups" option is selected.
Step 2: Select which Teams + Groups
The "Select teams + groups" option is selected.
Group *
A link: "Select group(s)"

Select group (Modal Window)
The modal window has a search bar at the top. Below it is a list of groups:

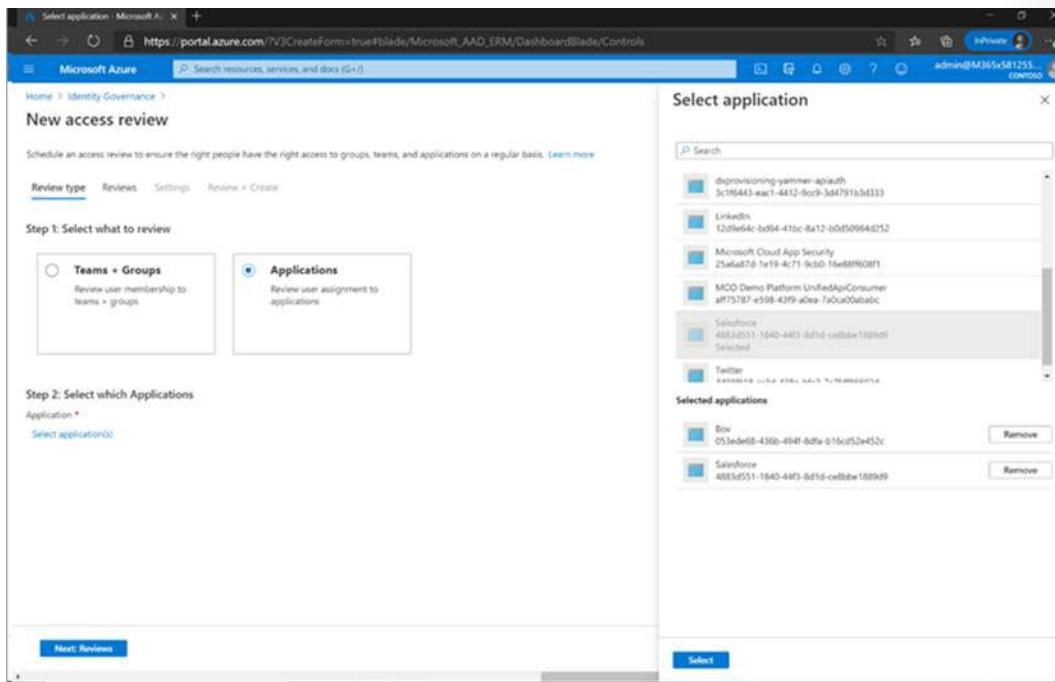
- All Company (allcompany@M365x581255.onmicrosoft.com)
- All Employees (Employees@M365x581255.onmicrosoft.com)
- Ask HR (askhr@M365x581255.onmicrosoft.com)
- CEO Connection (ceoconnection@M365x581255.onmicrosoft.com)
- Contoso (Contoso@M365x581255.onmicrosoft.com)
- Contoso Life (contosolife@M365x581255.onmicrosoft.com)

Selected groups:
No groups selected

Select

- If you selected **Applications** in Step 1, you can then select one or more applications in Step 2.

Note - Selecting multiple groups and/or applications will result in multiple access reviews created. For example, if you select five groups to review, that will result in five separate access reviews



- Next, in Step 3 you can select a scope for the review. Your options are:

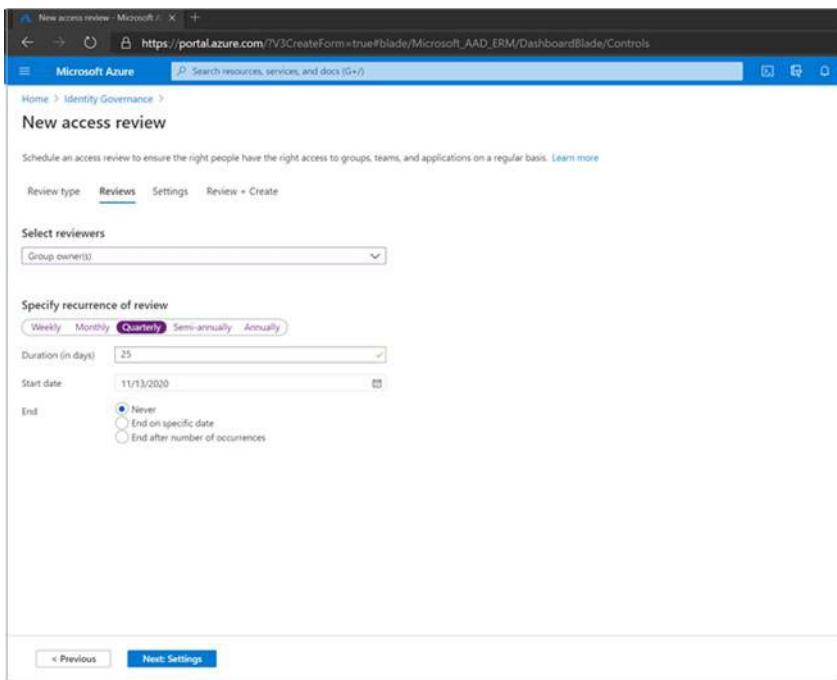
- Guest users only.** Selecting this option limits the access review to just the Azure AD B2B guest users in your directory.
- Everyone.** Selecting this option scopes the access review to all user objects associated with the resource.

Note - If you selected All Microsoft 365 groups with guest users in Step 2, then your only option is to review Guest users in Step 3.

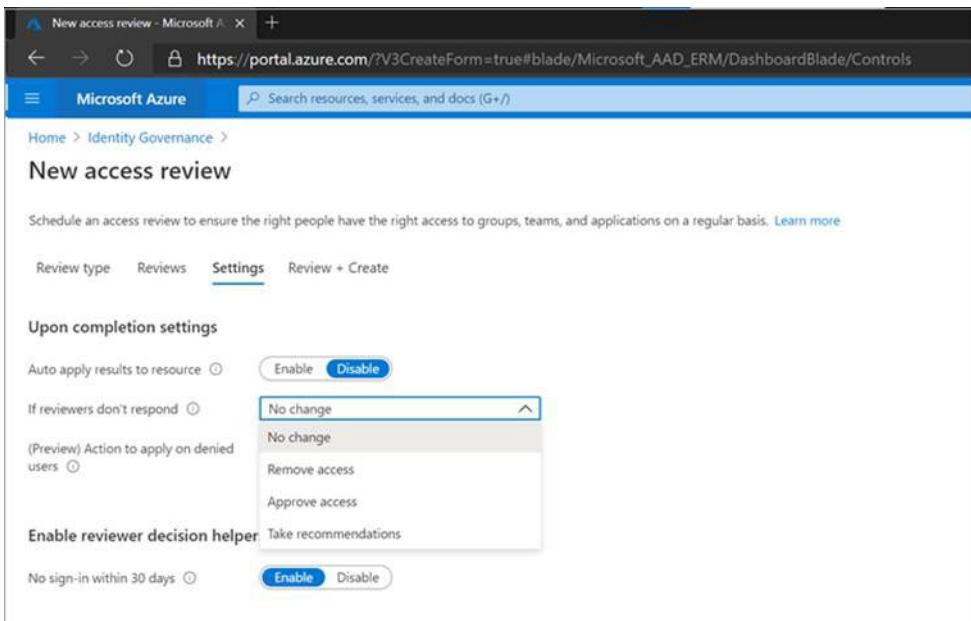
- Click on Next: Reviews
- In the **Select reviewers** section, select one or more people to perform the access reviews. You can choose from:
 - Group owner(s)** (Only available when performing a review on a team or group)
 - Selected user(s) or groups(s)**
 - Users review own access**
 - (Preview) Managers of users.** If you choose either **Managers of users** or **Group owners**, you also have the option to specify a fallback reviewer. Fallback reviewers are asked to do a review when the user has no manager specified in the directory or the group does not have an owner.

The screenshot shows the Microsoft Azure portal interface for creating a new access review. The top navigation bar includes the Microsoft Azure logo, a search bar, and a link to the portal address. Below the navigation is a breadcrumb trail: Home > Identity Governance > New access review. The main content area has tabs for 'Review type', 'Reviews' (which is underlined in blue), 'Settings', and 'Review + Create'. A prominent dropdown menu is open under the heading 'Select reviewers', containing four options: 'Group owner(s)', 'Selected user(s) or group(s)', 'Users review own access', and 'Managers of users'. The URL in the browser's address bar is https://portal.azure.com/?V3CreateForm=true#blade/Microsoft_AAD_ERM/DashboardBlade/Controls.

10. In the **Specify recurrence of review** section, you can specify a frequency such as **Weekly, Monthly, Quarterly, Semi-annually, Annually**. You then specify a **Duration**, which defines how long a review will be open for input from reviewers. For example, the maximum duration that you can set for a monthly review is 27 days to avoid overlapping reviews. You might want to shorten the duration to ensure that your reviewers input is applied earlier. Next, you can select a **Start date** and **End date**.



11. Click the **Next: Settings** button at the bottom of the page.
12. In the **Upon completion settings**, you can specify what happens after the review completes.



If you want to automatically remove access for denied users, set **Auto apply results to resource** to **Enable**. If you want to manually apply the results when the review completes, set the switch to **Disable**. Use the **If reviewers don't respond** list to specify what happens for users that are not reviewed by the reviewer within the review period. This setting does not impact users who have been reviewed by the reviewers manually. If the final reviewer's decision is Deny, then the user's access will be removed.

- **No change** - Leave user's access unchanged

- **Remove access** - Remove user's access
- **Approve access** - Approve user's access
- **Take recommendations** - Take the system's recommendation on denying or approving the user's continued access

The screenshot shows the 'New access review' settings page in the Microsoft Azure portal. The URL is https://portal.azure.com/?V3CreateForm=true#blade/Microsoft_AAD_ERM/DashboardBlade/Controls. The page title is 'New access review'. Below it, a sub-header says 'Schedule an access review to ensure the right people have the right access to groups, teams, and applications on a regular basis. [Learn more](#)'. There are tabs for 'Review type', 'Reviews', 'Settings' (which is selected), and 'Review + Create'. Under 'Upon completion settings', there are three main sections: 'Auto apply results to resource' (radio button set to 'Disable'), 'If reviewers don't respond' (dropdown menu showing 'No change' as selected, with other options like 'No change', 'Remove access', and 'Approve access' listed), and '(Preview) Action to apply on denied users' (radio button set to 'No change'). At the bottom, there is a section for 'Enable reviewer decision helper' with a dropdown menu showing 'Take recommendations' as selected, and 'No sign-in within 30 days' with 'Enable' and 'Disable' buttons.

Use the Action to apply on denied **guest** users to specify what happens to guest users if they are denied.

- **Remove user's membership from the resource** will remove denied user's access to the group or application being reviewed, they will still be able to sign-in to the tenant.
- **Block user from signing in for 30 days, then remove user from the tenant** will block the denied users from signing in to the tenant, regardless if they have access to other resources. If there was a mistake or if an admin decides to re-enable one's access, they can do so within 30 days after the user has been disabled. If there is no action taken on the disabled users, they will be deleted from the tenant.

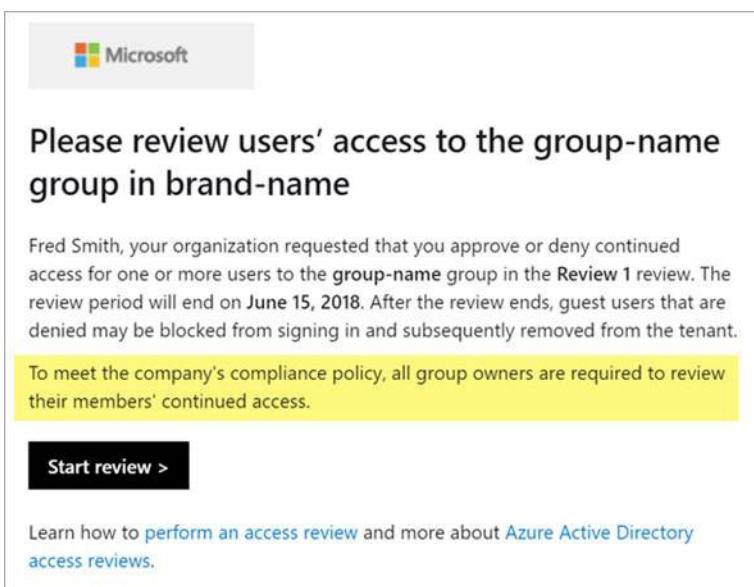
Note - Action to apply on denied guest users is not configurable on reviews scoped to more than guest users. It is also not configurable for reviews of all M365 groups with guest users. When not configurable, the default option of removing user's membership from the resource is used on denied users.

13. In **Enable review decision helpers** choose whether you would like your reviewer to receive recommendations during the review process.

The screenshot shows a configuration dialog for 'Enable review decision helpers'. It contains a single section with the heading 'Enable review decision helpers'. Below it is a 'No sign-in within 30 days' checkbox followed by 'Enable' and 'Disable' buttons. The 'Enable' button is highlighted with a blue background.

14. In the **Advanced settings** section, you can choose the following

- Set **Justification required** to **Enable** to require the reviewer to supply a reason for approval.
- Set **email notifications** to **Enable** to have Azure AD send email notifications to reviewers when an access review starts, and to administrators when a review completes.
- Set **Reminders** to **Enable** to have Azure AD send reminders of access reviews in progress to reviewers who have not completed their review. These reminders will be half-way through the duration of the review.
- The content of the email sent to reviewers is autogenerated based on the review details, such as review name, resource name, due date, etc. If you need a way to communicate additional information, such as additional instructions or contact information, you can specify these details in the **Additional content for reviewer email** section. The information that you enter is included in the invitation and reminder emails sent to assigned reviewers. The section highlighted in the image below shows where this information is displayed.



15. Click on **Next: Review + Create** to move to the next page.

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

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

The screenshot shows the 'New access review' creation interface. At the top, there's a breadcrumb navigation: Home > Identity Governance > New access review. Below it, a subtitle says 'Schedule an access review to ensure the right people have the right access to groups, teams, and applications on a regular basis. Learn more'. There are tabs for 'Review type', 'Reviews', 'Settings', and 'Review + Create', with 'Review + Create' being the active tab. A section titled 'Name new access review' contains fields for 'Review name' (set to 'Review of Finance Team membership') and 'Description'. Below this is a summary section titled 'Confirm access review + create' which lists the selected resources (1 group selected), review scope (Guest users only), reviewers (Group owner(s)), frequency (Quarterly), and end date (No end). It also includes settings for auto-apply results (Disabled) and handling non-responding reviewers (No change). At the bottom of this section are buttons for '< Previous' and 'Create', with 'Create' being highlighted with a red border.

Start the access review

Once you have specified the settings for an access review, click **Start**. The access review will appear in your list with an indicator of its status.

The screenshot shows the 'Identity Governance - Access reviews' list page. On the left is a sidebar with links: Getting started, Entitlement management (Preview), Access packages, Catalogs, User assignments reports, Settings, Access reviews (which is the current section), Overview, Access reviews (highlighted in blue), Programs, and Audit logs. The main area has a header with 'New access review' and 'Columns' buttons. It shows a table with two rows:

CONTROL NAME	RESOURCE	STATUS
Marketing group	Group Marketing	Active
Engineering group	Group Engineering	Complete

By default, Azure AD sends an email to reviewers shortly after the review starts. If you choose not to have Azure AD 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 yourself to groups or applications.

If you have assigned guests as reviewers and they have not accepted the invite, they will not receive an email from access reviews because they must first accept the invitation prior to reviewing.

Access review status table

Status	Definition
NotStarted	Review was created, user discovery is waiting to start.
Initializing	User discovery is in progress to identify all users who are part of the review.
Starting	Review is starting. If email notifications are enabled, emails are being sent to reviewers.
InProgress	Review has started. If email notifications are enabled, emails have been sent to reviewers. Reviewers can submit decisions until the due date.
Completing	Review is being completed, and emails are being sent to the review owner.
Auto-Reviewing	Review is in a system reviewing stage. The system is recording decisions for users who were not reviewed based on recommendations or preconfigured decisions.
Auto-Reviewed	Decisions have been recorded by the system for all users who were not reviewed. Review is ready to proceed to Applying if Auto-Apply is enabled.
Applying	There will be no change in access for users who were approved.
Applied	Denied users, if any, have been removed from the resource or directory.
Failed	Review could not progress. This error could be related to the deletion of the tenant, a change in licenses, or other internal tenant changes.

Create reviews via APIs

You can also create access reviews using APIs. What you do to manage access reviews of groups and application users in the Azure portal can also be done using Microsoft Graph APIs.

Monitor Findings

Azure AD simplifies how enterprises manage access to groups and applications in Azure AD and other Microsoft Online Services with a feature called Azure AD access reviews.

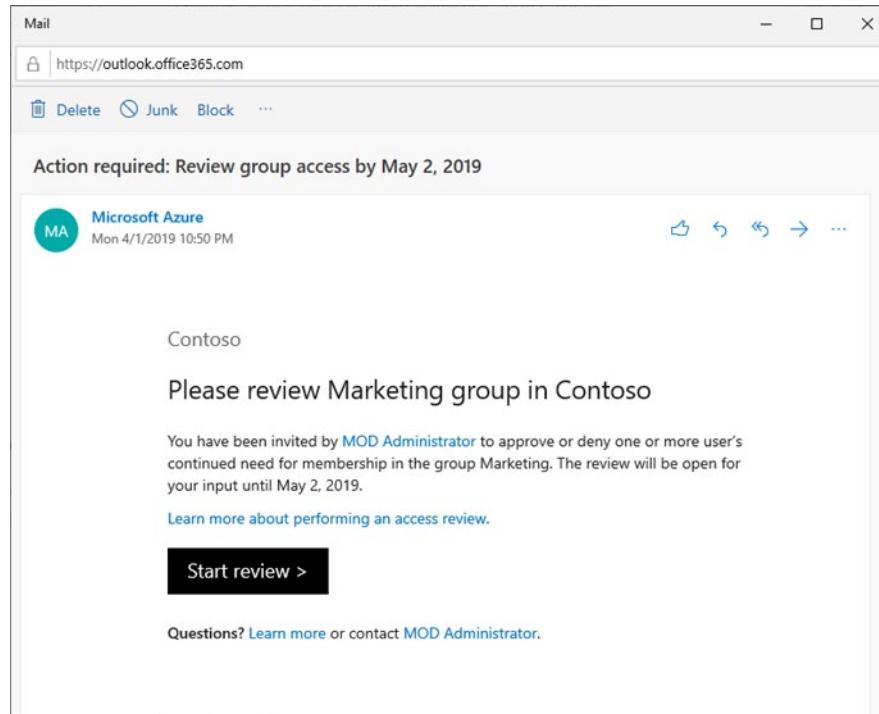
Perform access review using My Apps

You can start the Access Review process from the notification email or by going directly to the site.

1. Email:

Important - There could be delays in receiving email. In some cases it could take up to 24 hours. Add azure-noreply@microsoft.com to your safe recipients list to make sure that you are receiving all emails.

a. Look for an email from Microsoft asking you to review access. Here is an example email to review the access for a group.



b. Click the **Start review** link to open the access review.

2. **If you don't have the email**, you can find your pending access reviews by following these steps:

- Sign in to the My Apps portal at <https://myapps.microsoft.com>².
- In the upper-right corner of the page, click the user next to your name and default organization. If more than one organization is listed, select the organization that requested an access review.
- Click the **Access reviews** tile to see a list of pending access reviews.

Important - If the Access reviews tile isn't visible, there are no access reviews to perform for that organization and no action is needed at this time.

d. Click the **Begin review** link for the access review you want to perform.

Once you have opened the access review, you see the names of users who need to have their access reviewed.

There are two ways that you can approve or deny access:

- You can approve or deny access for one or more users 'manually' by choosing the appropriate action for each user request.
- You can accept the system recommendations.

² <https://myapps.microsoft.com/>

Approve or deny access for one or more users

1. Review the list of users and decide whether to approve or deny their continued access.
 - To approve or deny access for a single user, click the row to open a window to specify the action to take.
 - To approve or deny access for multiple users, add check marks next to the users and then click the **Review X user(s)** button to open a window to specify the action to take.
2. Click **Approve or Deny**.

Note - If you are unsure, you can click "Don't know" and the user gets to keep their access and your choice is recorded in the audit logs.
3. The administrator of the access review may require that you supply a reason in the **Reason** box for your decision. Even when a reason is not required. You can still provide a reason for your decision and the information that you include will be available to other reviewers.
4. Once you have specified the action to take, click **Save**.

Note - You can change your response at any time before the access review has ends. If you want to change your response, select the row and update the response. For example, you can approve a previously denied user or deny a previously approved user.

 - If a user is denied access, they aren't removed immediately. They are removed when the review period has ended or when an administrator stops the review if **Auto apply**³ is enabled.
 - If there are multiple reviewers, the last submitted response is recorded. Consider an example where an administrator designates two reviewers – Alice and Bob. Alice opens the access review first and approves a user's access request. Before the review period ends, Bob opens the access review and denies access on the same request previously approved by Alice. The last decision denying the access is the response that gets recorded.

Approve or deny access based on recommendations

To make access reviews easier and faster for you, we also provide recommendations that you can accept with a single click. The recommendations are generated based on the user's sign-in activity.

1. In the blue bar at the bottom of the page, click **Accept recommendations**.

You see a summary of the recommended actions.
2. Click **Ok** to accept the recommendations.

Manage Licenses for Access Reviews

How many licenses must you have?

Your directory needs at least as many Azure AD Premium P2 licenses as the number of employees who will be performing the following tasks:

- Member users who are assigned as reviewers
- Member users who perform a self-review
- Member users as group owners who perform an access review

³ <https://docs.microsoft.com/azure/active-directory/governance/complete-access-review>

- Member users as application owners who perform an access review

For guest users, licensing needs will depend on the licensing model you're using. However, the below guest users' activities are considered Azure AD Premium P2 usage:

- Guest users who are assigned as reviewers
- Guest users who perform a self-review
- Guest users as group owners who perform an access review
- Guest users as application owners who perform an access review

Azure AD Premium P2 licenses are not required for users with the Global Administrator or User Administrator roles who set up access reviews, configure settings, or apply the decisions from the reviews.

Example license scenarios

Scenario	Calculation	Number of licenses
An administrator creates an access review of Group A with 75 users and 1 group owner, and assigns the group owner as the reviewer.	1 license for the group owner as reviewer.	1
An administrator creates an access review of Group B with 500 users and 3 group owners, and assigns the 3 group owners as reviewers.	3 licenses for each group owner as reviewers.	3
An administrator creates an access review of Group B with 500 users. Makes it a self-review.	500 licenses for each user as self-reviewers.	500
An administrator creates an access review of Group C with 50 member users and 25 guest users. Makes it a self-review.	50 licenses for each user as self-reviewers.	50
An administrator creates an access review of Group D with 6 member users and 108 guest users. Makes it a self-review.	6 licenses for each user as self-reviewers. Guest users are billed on a monthly active user (MAU) basis. No additional licenses are required.	6

Automate Management Tasks of Access Reviews

You can choose to have access removal automated by setting the **Auto apply results to resource option** to **Enable**. Once the review is completed and has ended, users who were not approved by the reviewer will automatically be removed from the resource—or kept with continued access. This could mean removing their group membership, their application assignment, or revoking their right to elevate to a privileged role.

Take recommendations

The recommendations are displayed to reviewers as part of the reviewer experience and indicate a person's last sign-in to the tenant or last access to an application. This information helps reviewers make the right access decision. Selecting "Take recommendations" will follow Access Review's recommendations. At the end of an Access Review, the system will apply these recommendations automatically for users who reviewers have not responded for.

Recommendations are based on the criteria in the access review. For example, if you configure the review to remove access with no interactive sign-in for 90 days, it will recommend that all users who fit that criteria be removed. Microsoft is continually working on enhancing recommendations.

Review guest user access

Use Access Reviews to review and clean up collaboration partners' identities from external organizations. Configuration of a per-partner review may satisfy compliance requirements.

External identities can be granted access to company resources through one of the following actions:

- Added to a group.
- Invited to Teams.
- Assigned to an enterprise application or access package.
- Assigned a privileged role in Azure AD or in an Azure subscription.

This **sample script**⁴ shows where external identities invited into the tenant are used. You can see external users' group membership, role assignments, and application assignments in Azure AD. The script won't show any assignments outside of Azure AD, such as direct rights assignment to SharePoint resources, without the use of groups.

When creating an Access Review for groups or applications, you can choose to let the reviewer focus on **Everyone with access**, or **Guest users only**. By selecting Guest users only, reviewers are given a focused list of external identities from Azure AD B2B that have access to the resource.

Configure Recurring Access Reviews

Access reviews can be set to occur on a recurring basis. Name your Access Review, select a start date, frequency, duration, end date, and you're ready to go. Reviewers will be notified at the start of each review. Reviewers can approve or deny access with a friendly interface and with the help of smart recommendations.

Why are recurring access review important? Because of lifecycle management. Everything that starts, needs to have an end date. Between the start and end, we need to ensure permissions are what we need them to be. Not too much, not too little. And we regularly ask an owner if everything is still what they want it to be. With recurrence we make sure this checking will be done regularly.

⁴ <https://github.com/microsoft/access-reviews-samples/tree/master/ExternalIdentityUse>

Plan and Implement Privileged Access

Introduction

Increasing your Azure solution security by ensuring that administrative roles are protected and managed is a must. Explore how to use Privileged Identity Management (PIM) to protect your data and resources. In this module, you will learn how to create an access strategy, configure and assign PIM roles and resources accurately, and manage emergency access accounts.

Learning objectives

By the end of this module, you will be able to:

- Define a privileged access strategy for administrative users (resources, roles, approvals, thresholds).
- Configure PIM for Azure AD roles.
- Configure PIM for Azure resources.
- Assign roles.
- Manage PIM requests.
- Analyze PIM audit history and reports.
- Create and manage emergency access accounts.

What is Azure Active Directory Privileged Identity Management?

PIM is a service in Azure Active Directory (Azure AD) that enables you to manage, control, and monitor access to important resources in your organization. Such resources include those in Azure AD, Azure, and other Microsoft Online Services, such as Microsoft 365 or Microsoft Intune.

Note - Note that an Azure AD Premium 2 license is needed to use this capability.

Reasons to use

Organizations want to minimize the number of people who have access to secure information or resources, because that reduces the chance of a malicious actor getting that access or an authorized user inadvertently impacting a sensitive resource. However, users still need to carry out privileged operations in Azure AD, Azure, Microsoft 365, or SaaS apps. Organizations can give users just-in-time privileged access to Azure resources and Azure AD, while managing the need for oversight of what those users are doing with their administrator privileges.

What does it do?

PIM provides time-based and approval-based role activation to mitigate the risks of excessive, unnecessary, or misused access permissions on resources that you care about. Key features of PIM include:

- Provide just-in-time privileged access to Azure AD and Azure resources.
- Assign time-bound access to resources using start and end dates.
- Require approval to activate privileged roles.

- Enforce multifactor authentication to activate any role.
- Use justification to understand why users activate.
- Get notifications when privileged roles are activated.
- Conduct access reviews to ensure users still need roles.
- Download audit history for internal or external audit.

What can I do with it?

Once you set up PIM, you will see tasks, manage, and activity options in the left navigation menu. As an administrator, you will choose between options such as managing Azure AD roles, managing Azure resource roles, or privileged access groups. When you choose what you want to manage, you will see the appropriate set of options.

Prerequisites

None

Define a Strategy for Administrative Users

Before you deploy PIM in your organization, follow the instructions and understand the concepts in this section to help you create a plan tailored to your organization's privileged identity requirements.

Note - PIM requires a Premium P2 license.

Identify your stakeholders

The following section helps you identify all the stakeholders who are involved in the project and need to sign off, review, or stay informed. It includes separate tables for deploying PIM for Azure AD roles and PIM for Azure roles. Add stakeholders to the following table as appropriate for your organization.

- **SO** = Sign off on this project
- **R** = Review this project and provide input
- **I** = Informed of this project

Stakeholders: Privileged Identity Management for Azure AD roles

Name	Role	Action
Name and email	Identity architect or Azure Global Administrator - A representative from the identity management team in charge of defining how to align this change with the core identity management infrastructure in your organization.	SO/R/I

Name	Role	Action
Name and email	Service owner / Line manager - A representative from the IT owners of a service or a group of services. They're key in making decisions and helping to roll out PIM for their team.	SO/R/I
Name and email	Security owner - A representative from the security team who can sign off that the plan meets the security requirements of your organization.	SO/R
Name and email	IT support manager / Help-desk - A representative from the IT support organization who can provide feedback on the portability of this change from a helpdesk perspective.	R/I
Name and email for pilot users	Privileged role users - The group of users for which privileged identity management is implemented. They'll need to know how to activate their roles once PIM is implemented.	I

Stakeholders: Privileged Identity Management for Azure roles

Name	Role	Action
Name and email	Subscription/Resource owner - A representative from the IT owners of each subscription or resource that you want to deploy PIM for.	SO/R/I
Name and email	Security owner - A representative from the security team that can sign off that the plan meets the security requirements of your organization.	SO/R
Name and email	IT support manager / Help-desk - A representative from the IT support organization who can provide feedback on the portability of this change from a helpdesk perspective.	R/I

Name	Role	Action
Name and email for pilot users	Azure role users - The group of users for which privileged identity management is implemented. They'll need to know how to activate their roles once PIM is implemented.	I

Start using Privileged Identity Management

As part of the planning process, prepare PIM by following our "Start using Privileged Identity Management" article. PIM gives you access to some features that are designed to help with your deployment.

If your goal is to deploy PIM for Azure resources, follow our "Discover Azure resources to manage in Privileged Identity Management" article. Only owners of subscriptions and management groups can bring these resources under management by PIM. After it is under management, the PIM functionality is available for owners at all levels, including management group, subscription, resource group, and resource. If you are a Global Administrator trying to deploy PIM for your Azure resources, you can elevate access to manage all Azure subscriptions to give yourself access to all Azure resources in the directory for discovery. However, we advise that you get approval from each of your subscription owners before managing their resources with PIM.

Enforce principle of least privilege

It's important to make sure that you've enforced the principle of least privilege in your organization for both your Azure AD and your Azure roles.

Plan least privilege delegation

For Azure AD roles, it is common for organizations to assign the Global Administrator role to a number of administrators when most administrators only need one or two specific and less-powerful administrator roles. With a large number of Global Administrators or other high-privilege roles, it's hard to track your privileged role assignments closely enough.

Follow these steps to implement the principle of least privilege for your Azure AD roles.

1. Understand the granularity of the roles by reading and understanding the available Azure AD administrator roles. You and your team should also reference administrator roles by identity task in Azure AD, which explains the least privileged role for specific tasks.
2. List who has privileged roles in your organization. You can use the PIM Discovery and insights (preview) to reduce your exposure.

The screenshot shows the Microsoft Azure Privileged Identity Management (PIM) interface. The left sidebar includes links for Quick start, Overview, Tasks (My roles, Pending requests, Approve requests, Review access), Manage (Roles, Assignments, Alerts, Access reviews, Discovery and insights (Preview) - highlighted with a red box), Activity (Resource audit, My audit), and Settings. The main content area is titled 'Discovery and insights' and describes how it finds privileged role assignments across Azure AD and provides recommendations for securing them using PIM features. It lists 'Key Concepts' such as what PIM is, eligible role assignments, and using access reviews. Below this is a section titled 'Discovered assignments in Contoso' showing three categories: permanent global administrator assignments (644), accounts assigned to highly privileged roles (51), and service principals with privileged role assignments (8). Each category has a brief description and a 'Review' or 'Eliminate standing access' button.

3. For all Global Administrators in your organization, find out why they need the role. Then remove them from the Global Administrator role and assign built-in roles or custom roles with lower privilege inside Azure AD. FYI, Microsoft currently only has about 10 administrators with the Global Administrator role.
4. For all other Azure AD roles, review the list of assignments, identify administrators who no longer need the role, and remove them from their assignments.

To automate the last two steps, you can use access reviews in PIM. Following the steps in "Start an access review for Azure AD roles in Privileged Identity Management," you can set up an access review for every Azure AD role that has one or more members.

Create an access review

Access reviews allow reviewers to attest to whether users still need to be in a role. [Learn more about access reviews here.](#)

* Review name: Review all global administrators in the organization ✓

Description:

* Start date: 2019-01-17 ✓

Frequency: One time ✓

Duration (in days): 1

End: Never End by Occurrences

* Number of times: 0

* End date: 2019-02-17 ✓

Users

Scope: Everyone

* Review role membership: Global Administrator >

Reviewers

Reviewers: Members (self) ✓

Upon completion settings

Advanced settings

Start

Set the reviewers to **Members (self)**. All users in the role will receive an email asking them to confirm that they need the access. Also, turn on **Require reason on approval** in the advanced settings so that users must state why they need the role. Based on this information, you can remove users from unnecessary roles or delegate them to more granular administrator roles.

Access reviews rely on emails to notify people to review their access to the roles. If you have privileged accounts that don't have emails linked, be sure to populate the secondary email field on those accounts.

Plan Azure resource role delegation

For Azure subscriptions and resources, you can set up a similar Access review process to review the roles in each subscription or resource. The goal of this process is to minimize Owner and User Access Adminis-

trator assignments attached to each subscription or resource and to remove unnecessary assignments. However, organizations often delegate such tasks to the owner of each subscription or resource because they have a better understanding of the specific roles (especially custom roles).

If you're in the Global Administrator role trying to deploy PIM for Azure roles in your organization, you can **elevate access to manage all Azure subscriptions**⁵ to get access to each subscription. You can then find each subscription owner and work with them to remove unnecessary assignments and minimize owner role assignment.

Users with the Owner role for an Azure subscription can also use **access reviews for Azure resources**⁶ to audit and remove unnecessary role assignments similar to the process described earlier for Azure AD roles.

Decide which role assignments should be protected by Privileged Identity Management

After cleaning up privileged role assignments in your organization, you'll need to decide which roles to protect with PIM.

If a role is protected by PIM, eligible users assigned to it must elevate to use the privileges granted by the role. The elevation process might also include obtaining approval, using multifactor authentication, and providing the reason they're activating. PIM can also track elevations through notifications and the PIM and Azure AD audit event logs.

Choosing which roles to protect with PIM can be difficult and will be different for each organization. This section provides our best practices for Azure AD and Azure roles.

Azure AD roles

It's important to prioritize protecting Azure AD roles that have the most permissions. Based on usage patterns among all PIM customers, the top 10 Azure AD roles managed by PIM are:

- Global Administrator
- Security Administrator
- User Administrator
- Exchange Administrator
- SharePoint Administrator
- Intune Administrator
- Security Reader
- Service Administrator
- Billing Administrator
- Skype for Business Administrator

Tip - Microsoft recommends you manage all your Global Administrators and Security Administrators using PIM as a first step, because they are the users who can do the most harm when compromised.

⁵ <https://docs.microsoft.com/azure/role-based-access-control/elevate-access-global-admin?toc=/azure/active-directory/privileged-identity-management/toc.json>

⁶ <https://docs.microsoft.com/azure/active-directory/privileged-identity-management/pim-resource-roles-start-access-review>

It's important to consider the most sensitive data and permissions for your organization. As an example, some organizations may want to protect their Power BI Administrator role or their Teams Administrator role using PIM, since they can access data and change core workflows.

If there are any roles with guest users assigned, they're vulnerable to attack.

Tip - Microsoft recommends that you manage all roles with guest users using PIM to reduce risk associated with compromised guest user accounts.

Reader roles like the Directory Reader, Message Center Reader, and Security Reader are sometimes regarded as less important than other roles, because they don't have write permission. However, we have some customers who also protect those roles because attackers with access to those accounts might be able to read sensitive data, including personal data. Take this risk into consideration when deciding whether you want reader roles in your organization to be managed using PIM.

Azure roles

When deciding which role assignments should be managed using PIM for Azure resources, you must first identify the subscriptions/resources that are most vital for your organization. Examples of such subscriptions/resources are:

- Resources that host the most sensitive data.
- Resources that core customer-facing applications depend on.

If you're a Global Administrator having trouble deciding which subscriptions and resources are most important, you should contact subscription owners in your organization to gather a list of resources managed by each subscription. Then, work with the subscription owners to group the resources based on severity level in the case they're compromised (low, medium, high). Prioritize managing resources with PIM based on this severity level.

Tip - Microsoft recommends you work with subscription/resource owners of critical services to set up PIM workflow for all roles inside sensitive subscriptions/resources.

PIM for Azure resources supports time-bound service accounts. You should treat service accounts exactly the same as you would treat a regular user account.

For subscriptions/resources that are not as critical, you won't need to set up PIM for all roles. However, you should still protect the Owner and User Access Administrator roles with PIM.

Tip - Microsoft recommends that you manage Owner roles and User Access Administrator roles of all subscriptions/resources using PIM.

Decide whether to use a group to assign roles

Whether to assign a role to a group instead of to individual users is a strategic decision. When planning, consider assigning a role to a group to manage role assignments when:

- Many users are assigned to a role.
- You want to delegate assigning the role.

Many users are assigned to a role

Manually keeping track of who is assigned to a role and managing their assignments based on when they need it can take time. To assign a group to a role, first create a role-assignable group and then assign the group as eligible for a role. This action subjects everyone in the group to the same activation process as

individual users who are eligible to elevate into the role. Group members activate their assignments to the group individually using the PIM activation request and approval process. The group isn't activated—just the user's group membership.

You want to delegate assigning the role

A group owner can manage membership for a group. For Azure AD role-assignable groups, only the Privileged Role Administrator, the Global Administrator, and the group owners can manage group membership. By adding new members to the group, the member gets access to the roles to which the group is assigned whether the assignment is eligible or active. Use group owners to delegate the management of group membership for an assigned role to reduce the breadth of privilege required.

Tip - Microsoft recommends that you bring Azure AD role-assignable groups under management by PIM. After a role-assignable group is brought under management by PIM, it's called a privileged access group. Use PIM to require group owners to activate their Owner role assignment before they can manage group membership.

Decide which role assignments should be permanent or eligible

Once you have decided the list of roles to be managed by PIM, you must decide which users should get the eligible role versus the permanently active role. **Permanently active roles are the normal roles assigned through Azure AD and Azure resources, while eligible roles can only be assigned in PIM.**

Microsoft recommends you have zero permanently active assignments for both Azure AD roles and Azure roles other than the recommended **two break-glass emergency access accounts⁷**, which should have the permanent Global Administrator role.

Even though we recommend zero standing administrators, it is sometimes difficult for organizations to achieve this right away. Things to consider when making this decision include:

- Frequency of elevation – If the user only needs the privileged assignment once, they shouldn't have the permanent assignment. On the other hand, if the user needs the role for their day-to-day job and using PIM would greatly reduce their productivity, they can be considered for the permanent role.
- Cases specific to your organization – If the person being given the eligible role is from a distant team or a high-ranking executive to the point that communicating and enforcing the elevation process is difficult, they can be considered for the permanent role.

Tip - Microsoft recommends you to set up recurring access reviews for users with permanent role assignments (should you have any).

Draft your Privileged Identity Management settings

Before you implement your PIM solution, it is good practice to draft your PIM settings for every privileged role your organization uses. This section has some examples of PIM settings for particular roles; they are for reference only and might be different for your organization. Each of these settings is explained in detail with Microsoft's recommendations after the tables.

⁷ <https://docs.microsoft.com/azure/active-directory/roles/security-emergency-access>

Privileged Identity Management settings for Azure AD roles

Role	Global Administrator	Exchange Administrator	Helpdesk Administrator
Require MFA	Yes	Yes	No
Notification	Yes	Yes	No
Incident ticket	Yes	No	Yes
Require approval	Yes	No	No
Approver	Other Global Administrators	None	None
Activation Duration	1 hour	2 hour	8 hour
Permanent admin	Emergency access accounts	None	None

Privileged Identity Management settings for Azure roles

Role	Owner of critical subscriptions	User Access Administrator of less critical subscriptions	Virtual Machine Contributor
Require MFA	Yes	Yes	No
Notification	Yes	Yes	Yes
Require approval	Yes	No	No
Approver	Other owners of the subscription	None	None
Activation Duration	1 hour	1 hour	3 hour
Active admin	None	None	None
Active expiration	n/a	n/a	n/a

The following table describes each of the settings.

Setting	Description
Role	Name of the role you are defining the settings for.
Require MFA	Whether the eligible user needs to perform MFA before activating the role. Microsoft recommends you enforce MFA for all administrator roles, especially if the roles have guest users.
Notification	If set to true, Global Administrator, Privileged Role Administrator, and Security Administrator in the organization will receive an email notification when an eligible user activates the role. Note: Some organizations don't have an email address tied to their administrator accounts. To get these email notifications, set an alternative email address so administrators will receive these emails.

Setting	Description
Incident ticket	Whether the eligible user needs to record an incident ticket number when activating their role. This setting helps an organization identify each activation with an internal incident number to mitigate unwanted activations. Microsoft recommends taking advantage of incident ticket numbers to tie PIM into your internal system. This method can be useful for approvers who need context for the activation.
Require approval	Whether the eligible user needs to get approval to activate the role. Microsoft recommends that you set up approval for roles with the most permission. Based on usage patterns of all PIM customers, Global Administrator, User Administrator, Exchange Administrator, Security Administrator, and Password Administrator are the most common roles with approval set up.
Approver	If approval is required to activate the eligible role, list the people who should approve the request. By default, PIM sets the approver to be all users who are privileged role administrators whether they are permanent or eligible. Note: If a user is both eligible for an Azure AD role and an approver of the role, they will not be able to approve themselves. Microsoft recommends that you choose approvers to be users who are most knowledgeable about the role and its frequent users rather than a Global Administrator.
Activation duration	The length of time a user will be activated in the role before it will expire.
Permanent admin	List of users who will be a permanent administrator for the role (never have to activate). Microsoft recommends you have zero standing administrator for all roles except for Global Administrators.
Active admin	For Azure resources, active administrator is the list of users who will never have to activate to use the role. This list is not referred to as permanent administrator like in Azure AD roles because you can set an expiration time for when the user will lose this role.
Active expiration	Active role assignments for Azure roles expire after the configured duration. You can choose from 15 days, 1 month, 3 months, 6 months, 1 year or permanently active.
Eligible expiration	Eligible role assignments for Azure roles expire after this duration. You can choose from 15 days, 1 month, 3 months, 6 months, 1 year or permanently eligible.

Configure PIM for Azure Resources

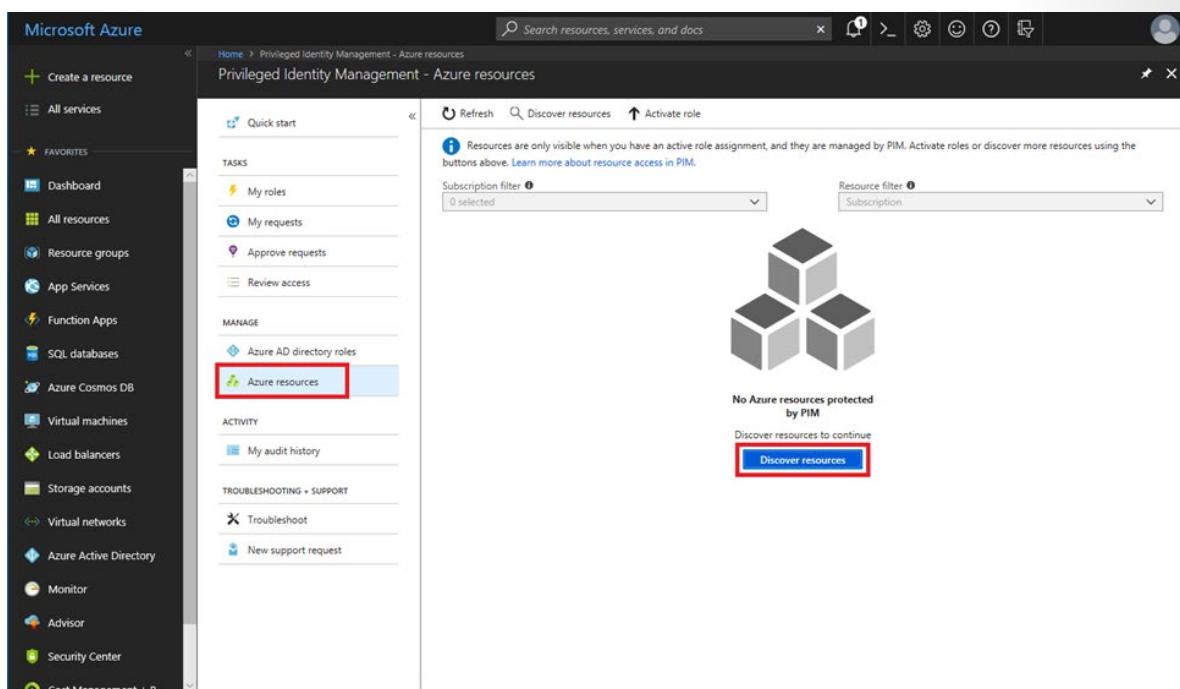
Using Azure AD PIM, you can improve the protection of your Azure resources. This is helpful to:

- Organizations that already use PIM to protect Azure AD roles.
- Management group and subscription owners who are trying to secure production resources.

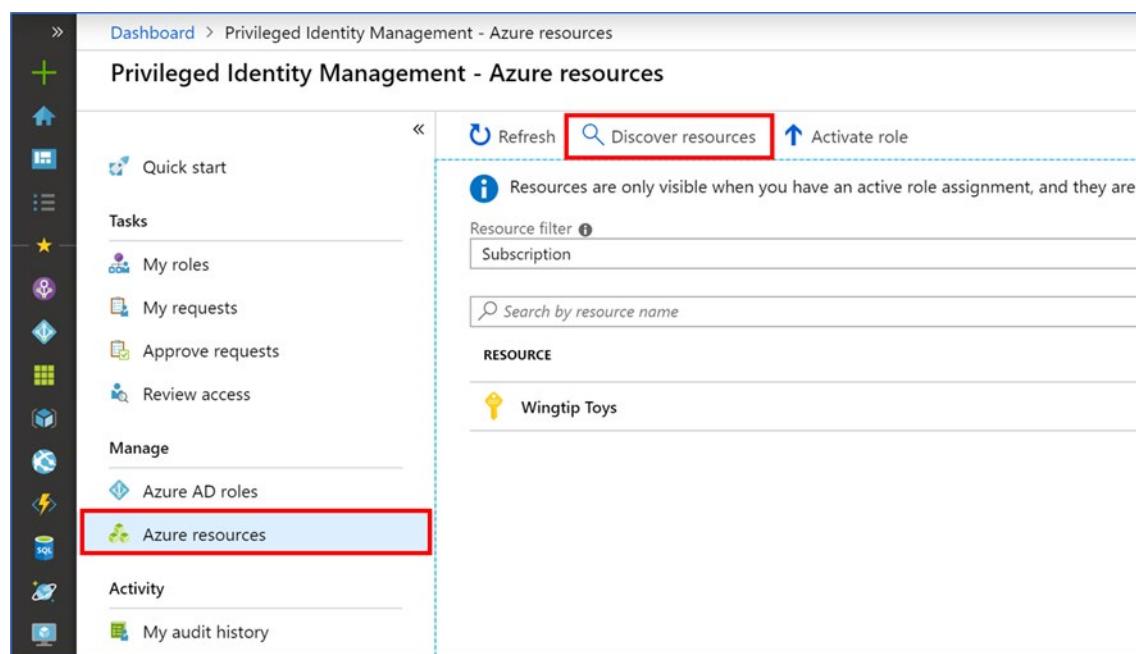
When you first set up PIM for Azure resources, you need to discover and select the resources to protect with PIM. There's no limit to the number of resources that you can manage with PIM. However, we recommend starting with your most critical production resources.

Discover resources

1. Sign in to the Azure portal.
2. Open **Azure AD Privileged Identity Management**.
3. Select **Azure resources**. If this is your first time using PIM for Azure resources, you'll see a **Discover resources** page.

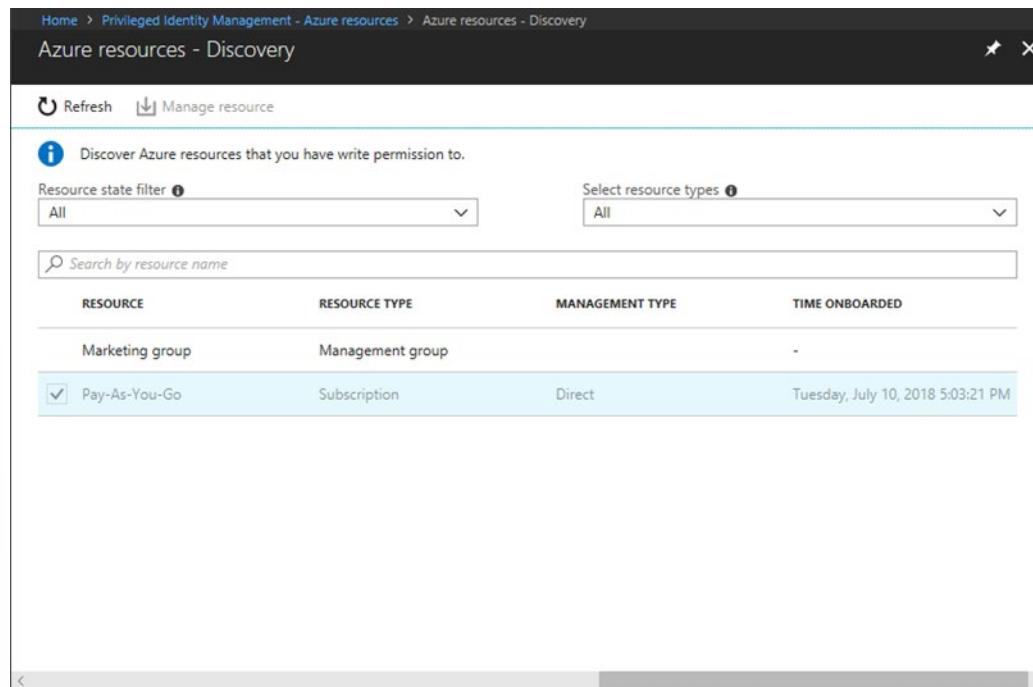


4. If another administrator in your organization is already managing Azure resources in PIM, you'll see a list of the resources that are currently being managed.



The screenshot shows the 'Privileged Identity Management - Azure resources' page. On the left, there's a sidebar with various icons and links: 'Quick start', 'Tasks' (with 'My roles', 'My requests', 'Approve requests', 'Review access'), 'Manage' (with 'Azure AD roles' and 'Azure resources' highlighted with a red box), and 'Activity' ('My audit history'). At the top right, there are buttons for 'Refresh', 'Discover resources' (which is highlighted with a red box), and 'Activate role'. Below these are sections for 'Resource filter' (set to 'Subscription') and 'Search by resource name' (with 'Wingtip Toys' listed). A note says 'Resources are only visible when you have an active role assignment, and they are not discoverable if you do not have the appropriate permissions.'

5. Select **Discover resources** to launch the discovery experience.



The screenshot shows the 'Azure resources - Discovery' page. It has a header with 'Home > Privileged Identity Management - Azure resources > Azure resources - Discovery'. Below the header are 'Refresh' and 'Manage resource' buttons. There are filters for 'Resource state filter' (set to 'All') and 'Select resource types' (set to 'All'). A search bar is labeled 'Search by resource name'. A table lists resources: 'Marketing group' (Management group) and 'Pay-As-You-Go' (Subscription, Direct management, onboarded on 'Tuesday, July 10, 2018 5:03:21 PM').

6. On the **Discovery** page, use **Resource state filter** and **Select resource type** to filter the management groups or subscriptions you have write permission to. It's probably easiest to start with **All** initially. You can search for and select management group or subscription resources to manage in PIM. When you manage a management group or a subscription in PIM, you can also manage its child resources.

Note - When you add a new child Azure resource to a PIM-managed management group, you can bring the child resource under management by searching for it in PIM.

7. Select any unmanaged resources that you want to manage.

8. Select **Manage resource** to start managing the selected resources.

Note - Once a management group or subscription is managed, it can't be unmanaged. This prevents another resource administrator from removing Privileged Identity Management settings.

The screenshot shows the 'Azure resources - Discovery' page. At the top, there's a 'Manage resource' button with a red box around it. Below it, there's a note: 'Discover Azure resources that you have write permission to.' There are filters for 'Resource state filter' (set to 'All') and 'Select resource types' (set to 'All'). A search bar says 'Search by resource name'. The main table has columns: RESOURCE, RESOURCE TYPE, MANAGEMENT TYPE, and TIME ONBOARDED. It lists two items: 'Marketing group' (Management group) and 'Pay-As-You-Go' (Subscription, Direct, onboared on Tuesday, July 10, 2018, 5:03:21 PM).

9. If you see a message to confirm the onboarding of the selected resource for management, select **Yes**.

The screenshot shows a confirmation dialog titled 'Onboarding selected resource for management'. It says 'PIM will manage all child objects for the selected resource(s), please confirm to continue.' At the bottom are 'Yes' and 'No' buttons.

Analyze PIM Audit History using Reports

With PIM, you can view activity, activations, and audit history for Azure privileged access group members and owners within your Azure AD organization.

If your organization has outsourced management functions to a service provider who uses **Azure delegated resource management**⁸, role assignments authorized by that service provider won't be shown here.

Follow these steps to view the audit history for privileged access groups.

View resource audit history

Resource audit gives you a view of all activity associated with your privileged access groups.

1. Open **Azure AD Privileged Identity Management**.
2. Select **Privileged access groups (Preview)**.
3. Select the privileged access group you want to view audit history for.
4. Under **Activity**, select **Resource audit**.

⁸ <https://docs.microsoft.com/azure/lighthouse/concepts/azure-delegated-resource-management>

5. Filter the history using a predefined date or custom range.

The screenshot shows the Microsoft Azure portal interface for 'App administrators | Resource audit'. The top navigation bar includes the Microsoft Azure logo, a search bar, and various icons. The main title is 'App administrators | Resource audit' under 'Privileged Identity Management | Privileged access groups'. On the left, there's a sidebar with 'Tasks' (My roles, Pending requests, Approve requests), 'Manage' (Roles, Assignments, Settings), and 'Activity' (Resource audit, My audit). The 'Resource audit' item is highlighted with a red box. The main content area has a filter section with 'Time span' (Last day), 'Audit type' (All), 'Original requestor' (All), 'Subject type' (All), and an 'Apply' button. Below this is a search bar ('Search by member name'). A table lists audit entries:

Time	Requestor	Action	Group	Primary target	Subject	Subject type
7/17/2020, 12:04:57 PM	Dritan Kodra	Add eligible member to role in PIM c	App administrators	Owner	acuser1@contoso.com	User
7/17/2020, 12:04:56 PM	Dritan Kodra	Add eligible member to role in PIM r	App administrators	Owner	acuser1@contoso.com	User
7/17/2020, 11:42:01 AM	Dritan Kodra	Add eligible member to role in PIM r	App administrators	Member	dritan@contoso.com	User
7/17/2020, 11:42:01 AM	Dritan Kodra	Add eligible member to role in PIM r	App administrators	Member	dritan@contoso.com	User
7/17/2020, 10:43:34 AM	Dritan Kodra	Add eligible member to role in PIM c	App administrators	Owner	aadmigration_0	User
7/17/2020, 10:43:33 AM	Dritan Kodra	Add eligible member to role in PIM r	App administrators	Owner	aadmigration_0	User

A modal window titled 'Audit details' is open, showing specific audit details:

Date	7/17/2020, 11:42:01 AM	Reason	-
Requestor	Dritan Kodra	Ticket system	-
Action	Add eligible member to role in PIM re...	Ticket number	-
Resource	App administrators	Scope	App administrators
Primary target	Member	Correlation ID	da0bfe3a-d1b9-43e9-8258-caeca587...
Subject	dritan@contoso.com		
Type	User		
Status	Succeeded		

View my audit

My audit enables you to view your personal role activity for a privileged access group.

1. Open **Azure AD Privileged Identity Management**.
2. Select **Privileged access groups (Preview)**.
3. Select the privileged access group you want to view audit history for.
4. Under **Activity**, select **My audit**.
5. Filter the history using a predefined date or custom range.

Action	Resource Name	Primary Target	Subject	Subject Type	Status
Add eligible member to role in PIM completed.	Wingtip Toys	Automation Operator	Shaun	Member	✓
Add eligible member to role in PIM requested.	Wingtip Toys	Automation Operator	Shaun	Member	✓
Remove member from role in PIM completed.	Wingtip Toys	Automation Operator	Shaun	Member	✓
Remove eligible member from role in PIM completed.	Wingtip Toys	Automation Operator	Tom	Member	✓
Add eligible member to role in PIM completed.	Wingtip Toys	Automation Operator	Tom	Member	✓
Add eligible member to role in PIM requested.	Wingtip Toys	Automation Operator	Tom	Member	✓
Add eligible member to role in PIM requested.	Wingtip Toys	Billing Reader	Shaun	Member	✓
Add member to role completed (PIM activation).	Wingtip Toys	Owner	Shaun	Member	✓
Add member to role requested (PIM activation).	Wingtip Toys	Owner	Shaun	Member	✓
Remove member from role (PIM activation failed).	Wingtip Toys	Owner	Shaun	Member	✓
Add eligible member to role in PIM completed.	Wingtip Toys	EventGrid EventSubscription Co...	Shaun	Member	✓
Add eligible member to role in PIM requested.	Wingtip Toys	EventGrid EventSubscription Co...	Shaun	Member	✓
Add eligible member to role in PIM completed.	Wingtip Toys	EventGrid EventSubscription Co...	-	Member	✓
Add eligible member to role in PIM requested.	Wingtip Toys	EventGrid EventSubscription Co...	-	Member	✓
Update role setting in PIM.	Wingtip Toys	EventGrid EventSubscription Co...	-	-	✓

Create and Manage Emergency Access Accounts

It is important that you prevent being accidentally locked out of your Azure Active Directory (Azure AD) organization because you can't sign in or activate another user's account as an administrator. You can mitigate the impact of accidental lack of administrative access by creating two or more **emergency access accounts** in your organization.

Emergency access accounts are highly privileged, and they are not assigned to specific individuals. Emergency access accounts are limited to emergency or **break glass** scenarios where normal administrative accounts can't be used. We recommend that you maintain a goal of restricting emergency account use to only the times when it is absolutely necessary.

This article provides guidelines for managing emergency access accounts in Azure AD.

Why use an emergency access account

An organization might need to use an emergency access account in the following situations:

- The user accounts are federated, and federation is currently unavailable because of a cell-network break or an identity-provider outage. For example, if the identity provider host in your environment has gone down, users might be unable to sign in when Azure AD redirects to their identity provider.
- The administrators are registered through Azure AD Multi-Factor Authentication, and all their individual devices are unavailable or the service is unavailable. Users might be unable to complete Multi-Factor Authentication to activate a role. For example, a cell network outage is preventing them from answering phone calls or receiving text messages, the only two authentication mechanisms that they registered for their device.
- The person with the most recent Global Administrator access has left the organization. Azure AD prevents the last Global Administrator account from being deleted, but it does not prevent the account from being deleted or disabled on-premises. Either situation might make the organization unable to recover the account.
- Unforeseen circumstances such as a natural disaster emergency, during which a mobile phone or other networks might be unavailable.

Create emergency access accounts

Create two or more emergency access accounts. These accounts should be cloud-only accounts that use the .onmicrosoft.com domain and that are not federated or synchronized from an on-premises environment.

When configuring these accounts, the following requirements must be met:

- The emergency access accounts should not be associated with any individual user in the organization. Make sure that your accounts are not connected with any employee-supplied mobile phones, hardware tokens that travel with individual employees, or other employee-specific credentials. This precaution covers instances where an individual employee is unreachable when the credential is needed. It is important to ensure that any registered devices are kept in a known, secure location that has multiple means of communicating with Azure AD.
- The authentication mechanism used for an emergency access account should be distinct from that used by your other administrative accounts, including other emergency access accounts. For example, if your normal administrator sign-in is via on-premises MFA, then Azure AD MFA would be a different mechanism. However if Azure AD MFA is your primary part of authentication for your administrative accounts, then consider a different approach for these, such as using Conditional Access with a third-party MFA provider via Custom controls.
- The device or credential must not expire or be in scope of automated cleanup due to lack of use.
- You should make the Global Administrator role assignment permanent for your emergency access accounts.

Exclude at least one account from phone-based multi-factor authentication

To reduce the risk of an attack resulting from a compromised password, Azure AD recommends that you require multi-factor authentication for all individual users. This group includes administrators and all others (for example, financial officers) whose compromised account would have a significant impact.

However, at least one of your emergency access accounts should not have the same multi-factor authentication mechanism as your other non-emergency accounts. This includes third-party multi-factor authentication solutions. If you have a Conditional Access policy to require multi-factor authentication for every administrator for Azure AD and other connected software as a service (SaaS) apps, you should exclude emergency access accounts from this requirement, and configure a different mechanism instead. Additionally, you should make sure the accounts do not have a per-user multi-factor authentication policy.

Exclude at least one account from Conditional Access policies

During an emergency, you do not want a policy to potentially block your access to fix an issue. At least one emergency access account should be excluded from all Conditional Access policies.

Federation guidance

An additional option for organizations that use AD Domain Services and ADFS or similar identity provider to federate to Azure AD, is to configure an emergency access account whose MFA claim could be supplied by that identity provider. For example, the emergency access account could be backed by a certificate and key pair such as one stored on a smartcard. When that user is authenticated to AD, ADFS can

supply a claim to Azure AD indicating that the user has met MFA requirements. Even with this approach, organizations must still have cloud-based emergency access accounts in case federation cannot be established.

Monitor sign-in and audit logs

Organizations should monitor sign-in and audit log activity from the emergency accounts and trigger notifications to other administrators. When you monitor the activity on break glass accounts, you can verify these accounts are only used for testing or actual emergencies. You can use Azure Log Analytics to monitor the sign-in logs and trigger email and SMS alerts to your admins whenever break glass accounts sign in.

Validate accounts regularly

When you train staff members to use emergency access accounts and validate the emergency access accounts, at minimum do the following steps at regular intervals:

- Ensure that security-monitoring staff are aware that the account-check activity is ongoing.
- Ensure that the emergency break glass process to use these accounts is documented and current.
- Ensure that administrators and security officers who might need to perform these steps during an emergency are trained on the process.
- Update the account credentials, in particular any passwords, for your emergency access accounts, and then validate that the emergency access accounts can sign-in and perform administrative tasks.
- Ensure that users have not registered Multi-Factor Authentication or self-service password reset (SSPR) to any individual user's device or personal details.
- If the accounts are registered for Multi-Factor Authentication to a device, for use during sign-in or role activation, ensure that the device is accessible to all administrators who might need to use it during an emergency. Also verify that the device can communicate through at least two network paths that do not share a common failure mode. For example, the device can communicate to the internet through both a facility's wireless network and a cell provider network.

These steps should be performed at regular intervals and for key changes:

- At least every 90 days
- When there has been a recent change in IT staff, such as a job change, a departure, or a new hire
- When the Azure AD subscriptions in the organization have changed

Monitor and Maintain Azure Active Directory

Introduction

Azure Active Directory (Azure AD) audit and diagnostic logs provide a rich view into how users are accessing your Azure solution. Learn to monitor, troubleshoot, and analyze sign-in data.

In this module, you will:

- Learn about sign-in, audit, and diagnostic logs.
- Learn about managing sign-in through a third-party security information and event management (SIEM) tool.
- Learn about reporting.

Learning objectives

By the end of this module you should be able to:

- Analyze and investigate sign-in logs to troubleshoot access issues.
- Review and monitor Azure AD audit logs.
- Enable and integrate Azure AD diagnostic logs with Log Analytics / Azure Sentinel.
- Export sign-in and audit logs to a third-party SIEM tool.
- Review Azure AD activity by using Log Analytics / Azure Sentinel, excluding KQL use.
- Analyze Azure AD workbooks/reporting.
- Configure notifications.

Prerequisites

None

Analyze Sign-in Logs and Troubleshoot Access Issues

The reporting architecture in Azure AD consists of the following components:

- **Activity**
 - **Sign-ins** - Information about the usage of managed applications and user sign-in activities.
 - **Audit logs** - Audit logs provide system activity information about users and group management, managed applications, and directory activities.
 - **Provisioning logs** - Provisioning logs enable customers to monitor activity by the provisioning service, such as creating a group in ServiceNow or a user imported from Workday.
- **Security**
 - **Risky sign-ins** - A risky sign-in is an indicator for a sign-in attempt by someone who isn't the legitimate owner of a user account.
 - **Users flagged for risk** - A risky user is an indicator for a user account that might have been compromised.

Who can access the data?

- Users in the Security Administrator, Security Reader, Global Reader, and Report Reader roles
- Global Administrators
- Any user (non-admins) can access their own sign-ins

What Azure AD license do you need to access sign-in activity?

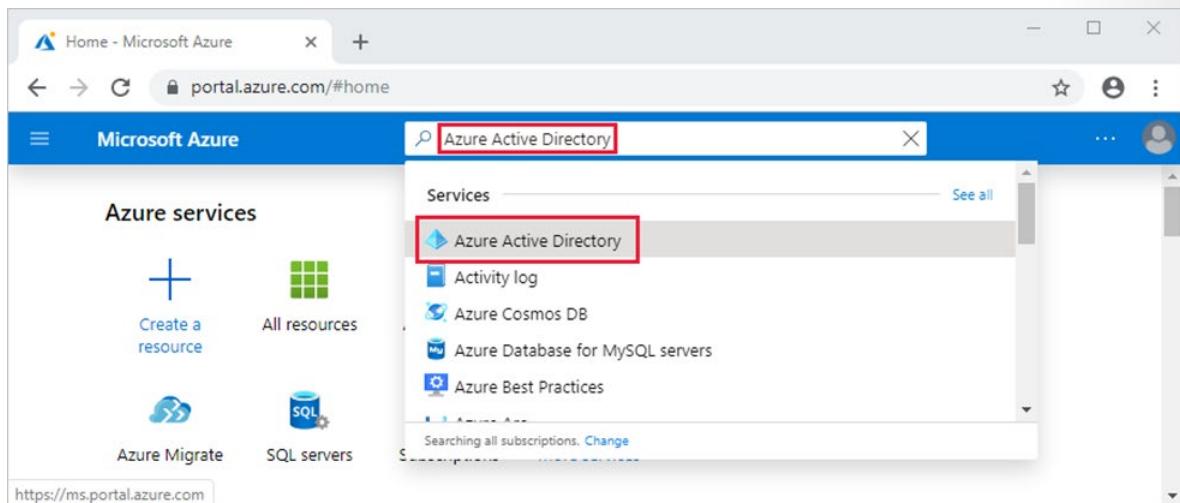
The sign-in activity report is available in all editions of Azure AD and can also be accessed through the Microsoft Graph API.

Sign-ins report

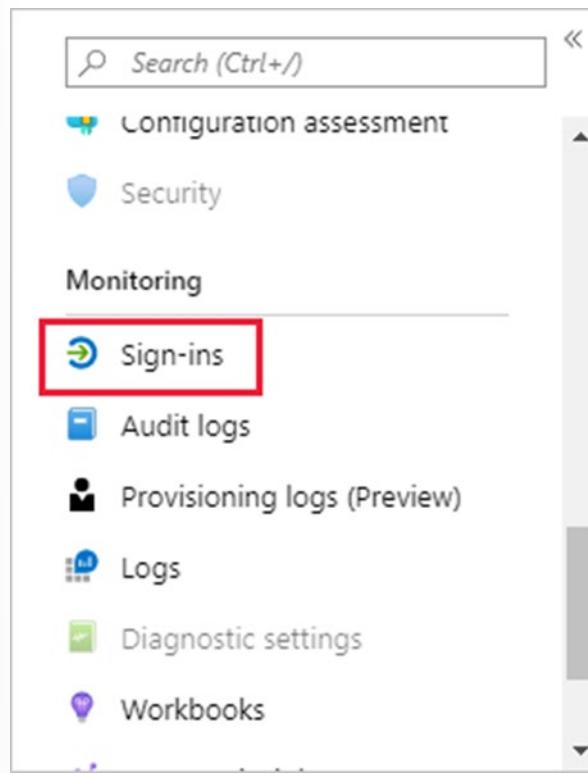
The user sign-ins report provides answers to the following questions:

- What is the sign-in pattern of a user?
- How many users have signed in over a week?
- What's the status of these sign-ins?

On the Azure portal menu, select **Azure Active Directory**, or search for and select **Azure Active Directory** from any page.



Under **Monitoring**, select **Sign-ins** to open the Sign-ins report.

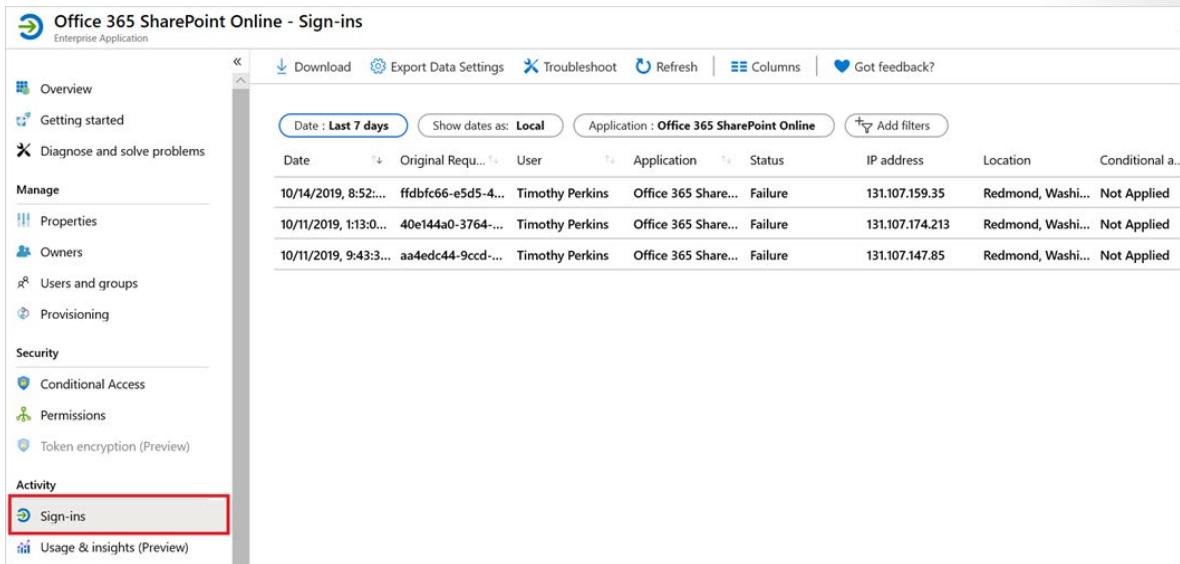


It may take up to two hours for some sign-in records to show up in the portal.

Important - The sign-ins report only displays the interactive sign-ins—those in which a user manually signs in using their username and password. Non-interactive sign-ins, such as service-to-service authentication, are not displayed in the sign-ins report.

A sign-ins log has a default list view that shows the:

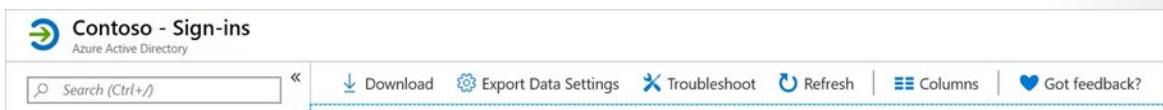
- Sign-in date
- Related user
- Application the user has signed in to
- Sign-in status
- Status of the risk detection
- Status of the multifactor authentication (MFA) requirement



The screenshot shows the 'Sign-ins' section of the Office 365 SharePoint Online interface. On the left, there's a navigation pane with links like Overview, Getting started, Diagnose and solve problems, Manage (Properties, Owners, Users and groups, Provisioning), Security (Conditional Access, Permissions, Token encryption (Preview)), and Activity (Sign-ins, Usage & insights (Preview)). The 'Sign-ins' link is highlighted with a red box. The main area displays a table of sign-in logs:

Date	Original Request ID	User	Application	Status	IP address	Location	Conditional a...
10/14/2019, 8:52:...	ffdbfc66-e5d5-4...	Timothy Perkins	Office 365 Share...	Failure	131.107.159.35	Redmond, Washi...	Not Applied
10/11/2019, 1:13:0...	40e144a0-3764-...	Timothy Perkins	Office 365 Share...	Failure	131.107.174.213	Redmond, Washi...	Not Applied
10/11/2019, 9:43:3...	aa4edc44-9cc...	Timothy Perkins	Office 365 Share...	Failure	131.107.147.85	Redmond, Washi...	Not Applied

You can customize the list view by clicking Columns in the toolbar.



The screenshot shows the 'Sign-ins' section of the Contoso - Sign-ins interface. The navigation pane is similar to the previous one, with the 'Sign-ins' link also highlighted with a red box. The main area displays a table of sign-in logs, which is identical to the one shown in the first screenshot.

The Columns dialog gives you access to the selectable attributes. In a sign-in report, you can't have fields that have more than one value for a given sign-in request as column. For example, this is true for authentication details, conditional access data, and network location.

Columns

<input type="checkbox"/> Date
<input checked="" type="checkbox"/> Request ID
<input checked="" type="checkbox"/> User
<input type="checkbox"/> Username
<input checked="" type="checkbox"/> Application
<input checked="" type="checkbox"/> Status
<input checked="" type="checkbox"/> IP address
<input checked="" type="checkbox"/> Location
<input type="checkbox"/> Resource
<input type="checkbox"/> Resource ID
<input type="checkbox"/> Client app
<input type="checkbox"/> Operating system
<input type="checkbox"/> Device browser
<input type="checkbox"/> Correlation ID
<input checked="" type="checkbox"/> Conditional access
<input type="checkbox"/> Alternate sign-in name
<input type="checkbox"/> Token issuer name
<input type="checkbox"/> Token issuer type

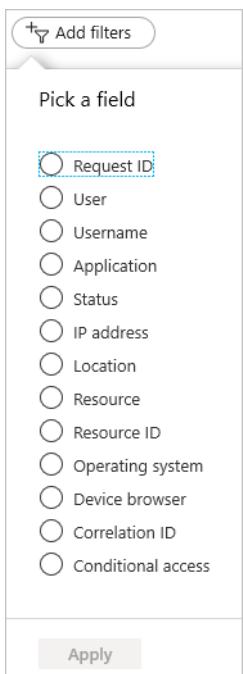
Select an item in the list view to get more detailed information.

Date	Original Req...	User	Application	Status	IP address	Location	Conditional a...
10/15/2019, 4:00...	3b849c9a-1671-4705-8f56-77fd8fa10000	Timothy Perkins	Azure Portal	Success	167.220.2.8	Redmond, Wash...	Not Applied
10/15/2019, 3:55...	ebcaecd4-c7cf-4...	Timothy Perkins	Kusto Web Expl...	Success	131.107.147.47	Redmond, Wash...	Not Applied
10/15/2019, 3:55...	b2b6a4fa-a726...	Timothy Perkins	Kusto Web Expl...	Success	131.107.147.47	Redmond, Wash...	Not Applied

Customers can now troubleshoot Conditional Access policies through all sign-in reports. By clicking on the Conditional Access tab for a sign-in record, customers can review the Conditional Access status and dive into the details of the policies that applied to the sign-in and the result for each policy. For more information, see the **FAQ about CA information in all sign-ins⁹**.

Filter sign-in activities

First, narrow down the reported data to a level that works for you. Second, filter sign-in data using date field as default filter. Azure AD provides you with a broad range of additional filters you can set:



Request ID - The ID of the request you care about.

User - The name or the user principal name (UPN) of the user you care about.

Application - The name of the target application.

Status - The sign-in status you care about:

- Success
- Failure
- Interrupted

IP address - The IP address of the device used to connect to your tenant.

Location - The location the connection was initiated from:

- City
- State/Province
- Country/Region

Resource - The name of the service used for the sign-in.

⁹ <https://docs.microsoft.com/azure/active-directory/reports-monitoring/reports-faq>

Resource ID - The ID of the service used for the sign-in.

Client app - The type of the client app used to connect to your tenant:

0 selected ^

- Authenticated SMTP
- Auto Discover
- Browser
- Exchange ActiveSync
- Exchange Online Powershell
- Exchange Web Services
- IMAP4
- MAPI Over HTTP
- Mobile Apps and Desktop clients
- Offline Address Book
- Other clients
- Outlook Anywhere (RPC over HTTP)
- Outlook Service
- POP3
- Reporting Web Services

Name	Modern authentication	Description
Authenticated SMTP		Used by POP and IMAP clients to send email messages.
Autodiscover		Used by Outlook and EAS clients to find and connect to mailboxes in Exchange Online.
Exchange ActiveSync		Shows all sign-in attempts where the EAS protocol has been attempted.
Browser	yes	Shows all sign-in attempts from users using web browsers.
Exchange ActiveSync		Shows all sign-in attempts from users with client apps using Exchange ActiveSync to connect to Exchange Online.
Exchange Online PowerShell		Used to connect to Exchange Online with remote PowerShell. If you block basic authentication for Exchange Online PowerShell, you need to use the Exchange Online PowerShell module to connect.
Exchange Web Services		A programming interface that's used by Outlook, Outlook for Mac, and third-party apps.

Name	Modern authentication	Description
IMAP4		A legacy mail client using IMAP to retrieve email.
MAPI over HTTP		Used by Outlook 2010 and later.
Mobile apps and desktop clients	yes	Shows all sign-in attempts from users using mobile apps and desktop clients.
Offline Address Book		A copy of address list collections that are downloaded and used by Outlook.
Outlook Anywhere (RPC over HTTP)		Used by Outlook 2016 and earlier.
Outlook Service		Used by the Mail and Calendar app for Windows 10.
POP3		A legacy mail client using POP3 to retrieve email.
Reporting Web Services		Used to retrieve report data in Exchange Online.
Other clients		Shows all sign-in attempts from users where the client app is not included or unknown.

Operating system - The operating system running on the device used to sign on to your tenant.

Device browser - If the connection was initiated from a browser, this field enables you to filter by browser name.

Correlation ID - The correlation ID of the activity.

Conditional access - The status of the applied conditional access rules.

- **Not applied:** No policy applied to the user and application during sign-in.
- **Success:** One or more conditional access policies applied to the user and application (but not necessarily the other conditions) during sign-in.
- **Failure:** The sign-in satisfied the user and application condition of at least one Conditional Access policy, and grant controls are either not satisfied or set to block access.

Download sign-in activities

Click the **Download** option to create a CSV or JSON file of the most recent 250,000 records. Start with **Download Sign-ins** if you want to work with the data outside the Azure portal.

Download Sign-ins

⚠️ You can download up to 250,000 records. If you want to download more, use reporting APIs. [Click here to learn more.](#)

i Your download will be based on the filter selections you have made.

Format
 CSV JSON

File Name
SignIns_2020-02-24_2020-02-26

Important - The number of records you can download is constrained by the **Azure AD report retention policies¹⁰**.

Sign-ins data shortcuts

Azure AD and the Azure portal both provide you with additional entry points to sign-in data:

- The Identity security protection overview
- Users
- Groups
- Enterprise applications

Users sign-in data in Identity security protection

The user sign-in graph in the **Identity security protection** overview page shows weekly aggregations of sign-ins. The default for the period is 30 days.



Click on a day in the sign-in graph, you get an overview of the sign-in activities for this day.

¹⁰ <https://docs.microsoft.com/azure/active-directory/reports-monitoring/reference-reports-data-retention>

Each row in the sign-in activities list shows:

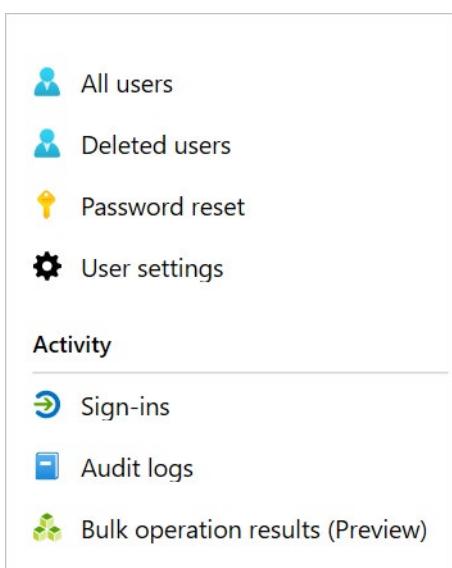
- Who has signed in?
- What application was the target of the sign-in?
- What is the status of the sign-in?
- What is the MFA status of the sign-in?

By clicking an item, you get more details about the sign-in operation:

- User ID
- User
- Username
- Application ID
- Application
- Client
- Location
- IP address
- Date
- MFA Required
- Sign-in status

Note - IP addresses are issued in such a way that there is no definitive connection between an IP address and where the computer with that address is physically located. Mapping IP addresses is complicated by the fact that mobile providers and VPNs issue IP addresses from central pools that are often very far from where the client device is actually used. Currently in Azure AD reports, converting an IP address to a physical location is a best effort based on traces, registry data, reverse look-ups, and other information.

On the **Users** page, you get a complete overview of all user sign-ins by clicking **Sign-ins** in the **Activity** section.

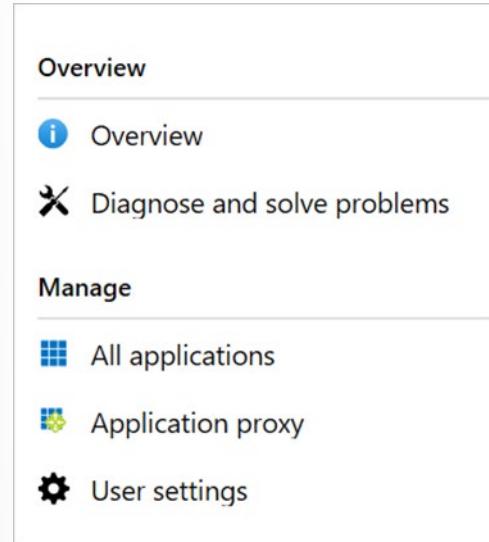


Usage of managed applications

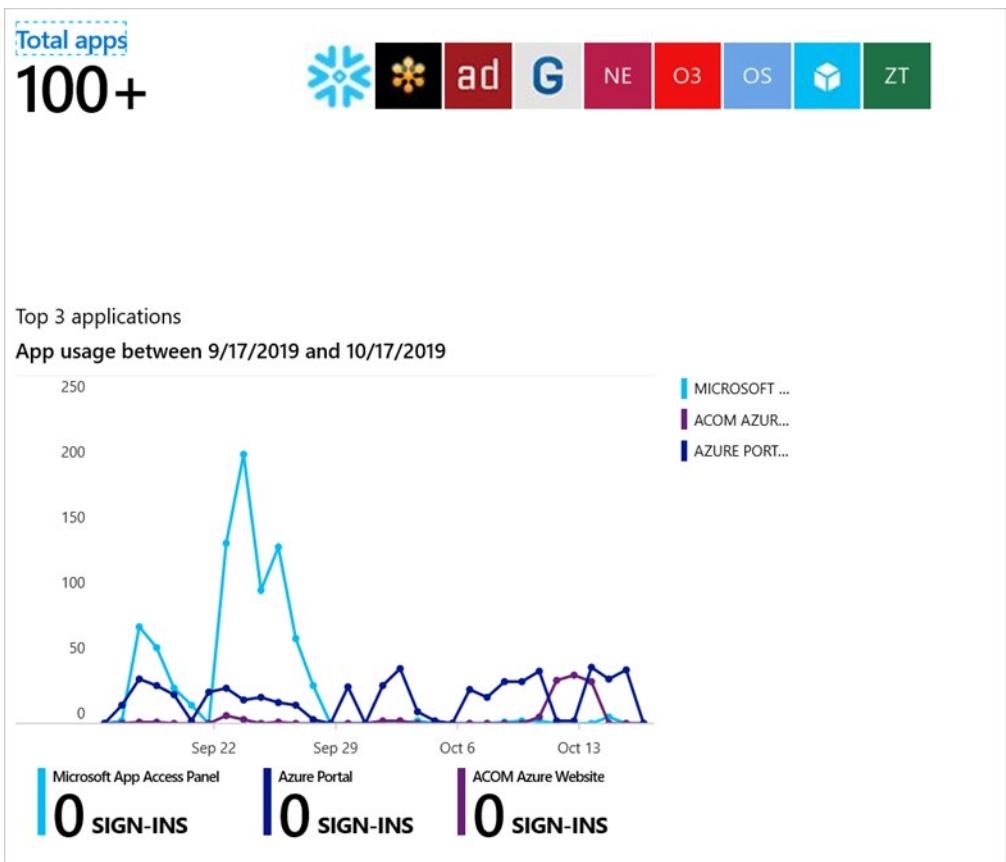
With an application-centric view of your sign-in data, you can answer questions such as:

- Who is using my applications?
- What are the top three applications in my organization?
- How is my newest application doing?

The entry point to this data is the top three applications in your organization. The data is contained within the last 30 days report in the **Overview** section under **Enterprise applications**.



The app-usage graphs weekly aggregations of sign-ins for your top three applications in a given time period. The default time period is 30 days.



If you want to, you can set the focus on a specific application.

Salesforce
Enterprise Application - PREVIEW

Delete

Search (Ctrl+ /)

GENERAL

- Overview
- Properties
- Owners
- User experience

MONITOR

- Audit
- Usage
- Provisioning

MANAGE

App usage between 7/12/2016 and 8/12/2016

SALESFORCE
15 sign-ins

The screenshot shows the Azure portal's Enterprise Application blade for the Salesforce application. The left sidebar has sections for General (Overview, Properties, Owners, User experience), Monitor (Audit, Usage, Provisioning), and Manage. The main area displays the app's usage graph from July 12, 2016, to August 12, 2016. The graph shows a sharp decline in sign-ins, starting at 12 and dropping to 0 by July 24. Below the graph, a summary states "15 sign-ins".

When you click on a day in the app usage graph, you get a detailed list of the sign-in activities.

The **Sign-ins** option gives you a complete overview of all sign-in events to your applications.

Review and Monitor Azure AD Audit-logs

With Azure AD reports, you can get the information you need to determine how your environment is doing.

Audit logs

The Azure AD audit logs provide records of system activities for compliance. To access the audit report, select **Audit logs** in the **Monitoring** section of **Azure Active Directory**.

An audit log has a default list view that shows the:

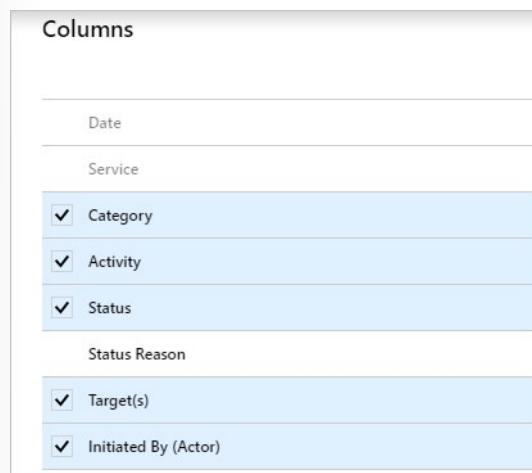
- Date and time of the occurrence
- Service that logged the occurrence
- Category and name of the activity (what)
- Status of the activity (success or failure)
- Target
- Initiator/actor (who) of an activity

DATE	SERVICE	CATEGORY	ACTIVITY	STATUS	TARGET(S)	INITIATED BY (ACTOR)
2/7/2019, 2:16:33 AM	Core Directory	Policy	Delete policy	Success	MFA Registration	admin@aad171ccscpt.net
2/7/2019, 2:16:33 AM	Core Directory	Policy	Delete policy	Success	02/07/2019 10:15 AM	admin@aad171ccscpt.net
2/7/2019, 2:15:56 AM	Identity Protection	Policy	Set MFA registration policy	Failure	Test_Test_aad171	admin@aad171ccscpt.net
2/7/2019, 2:15:56 AM	Core Directory	Policy	Add policy	Failure	MFA Registration	Azure AD Identity Protection

You can customize the list view by clicking **Columns** in the toolbar.



This enables you to display additional fields or remove fields that are already displayed.



Select an item in the list view to get more detailed information.

Activity	Target(s)	Modified Properties	INITIATED BY (ACTOR)	ADDITIONAL DETAILS
ACTIVITY				
DATE	2/7/2019, 2:15:54 AM		TYPE	User
ACTIVITY TYPE	Set MFA registration policy		DISPLAY NAME	
CORRELATION ID	5477a398-7be5-4827-bfcb-1e40fcdee234		OBJECT ID	d7cc485d-2c1b-422c-98fd-5ce52859a4a3
CATEGORY	Policy		USER PRINCIPAL NAME	
STATUS	Failure			
STATUS REASON	Set MFA registration policy failed.			

Filtering audit logs

You can filter the audit data on the following fields:

- Service
- Category
- Activity
- Status
- Target
- Initiated by (Actor)
- Date range

The screenshot shows a user interface for filtering audit logs. At the top, there are several buttons for setting filters: "Date : Last 1 month", "Show dates as: UTC", "Service : All", "Category : All", "Activity : All", and a "Add filters" button. To the right of these buttons is a modal window titled "Pick a field". Inside the modal, there is a list of three filter options: "Target" (which is selected, indicated by a blue dashed border around the input), "Initiated By (Actor)", and "Status". Below the list is a "Apply" button. The main area of the interface is mostly empty, suggesting no results are currently displayed.

The **Service** filter allows you to select from a drop-down list of the following services:

- All
- AAD Management UX
- Access Reviews
- Account Provisioning
- Application Proxy
- Authentication Methods
- B2C
- Conditional Access
- Core Directory

- Entitlement Management
- Hybrid Authentication
- Identity Protection
- Invited Users
- MIM Service
- MyApps
- PIM
- Self-service Group Management
- Self-service Password Management
- Terms of Use

The **Category** filter enables you to select one of the following filters:

- All
- AdministrativeUnit
- ApplicationManagement
- Authentication
- Authorization
- Contact
- Device
- DeviceConfiguration
- DirectoryManagement
- EntitlementManagement
- GroupManagement
- KerberosDomain
- KeyManagement
- Label
- Other
- PermissionGrantPolicy
- Policy
- ResourceManagement
- RoleManagement
- UserManagement

The **Activity** filter is based on the category and activity resource type selection you make. You can select a specific activity you want to see or choose all.

You can get the list of all Audit Activities using the Graph API: <https://graph.windows.net/<tenantdomain>/activities/auditActivityTypesV2?api-version=beta>

The **Status** filter allows you to filter based on the status of an audit operation. The status can be one of the following:

- All
- Success
- Failure

The **Target** filter allows you to search for a particular target by the starting of the name or user principal name (UPN). The target name and UPN are case-sensitive.

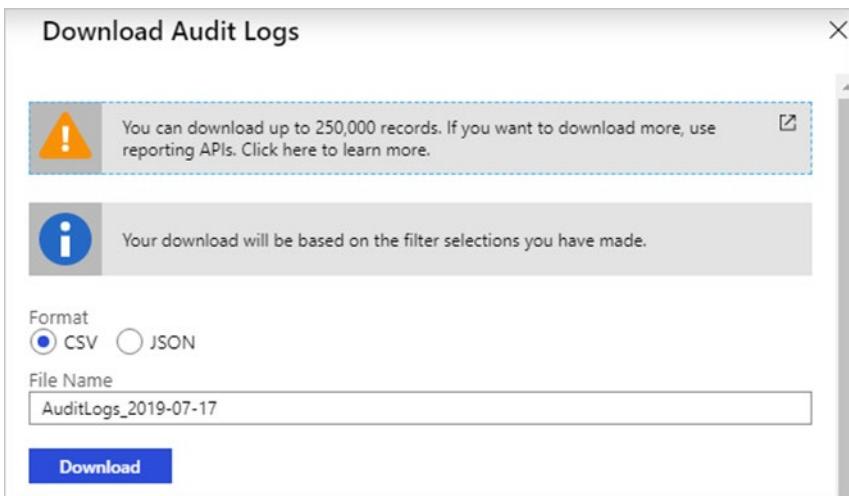
The **Initiated by** filter enables you to define what an actor's name or a universal principal name (UPN) starts with. The name and UPN are case-sensitive.

The **Date range** filter enables to you to define a timeframe for the returned data. Possible values are:

- 7 days
- 24 hours
- Custom

When you select a custom timeframe, you can configure a start time and an end time.

You can also choose to download the filtered data, up to 250,000 records, by selecting the **Download** button. You can download the logs in either CSV or JSON format. The number of records you can download is constrained by the Azure AD report retention policies.



Audit logs shortcuts

In addition to **Azure AD**, the Azure portal provides you with two additional entry points to audit data:

- Users and groups
- Enterprise applications

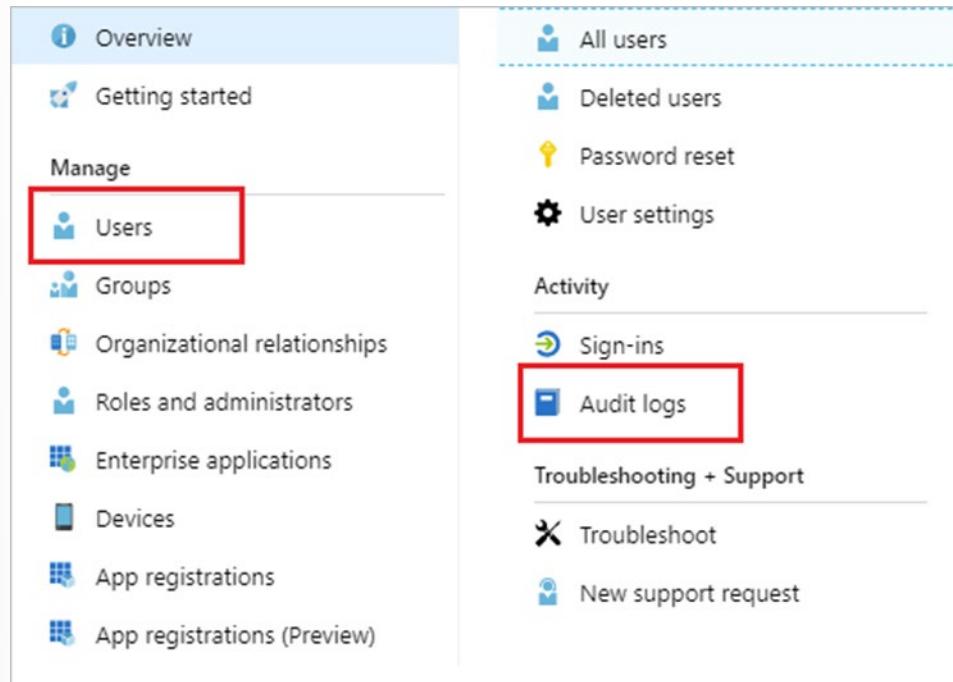
Users and groups audit logs

With user and group-based audit reports, you can get answers to questions such as:

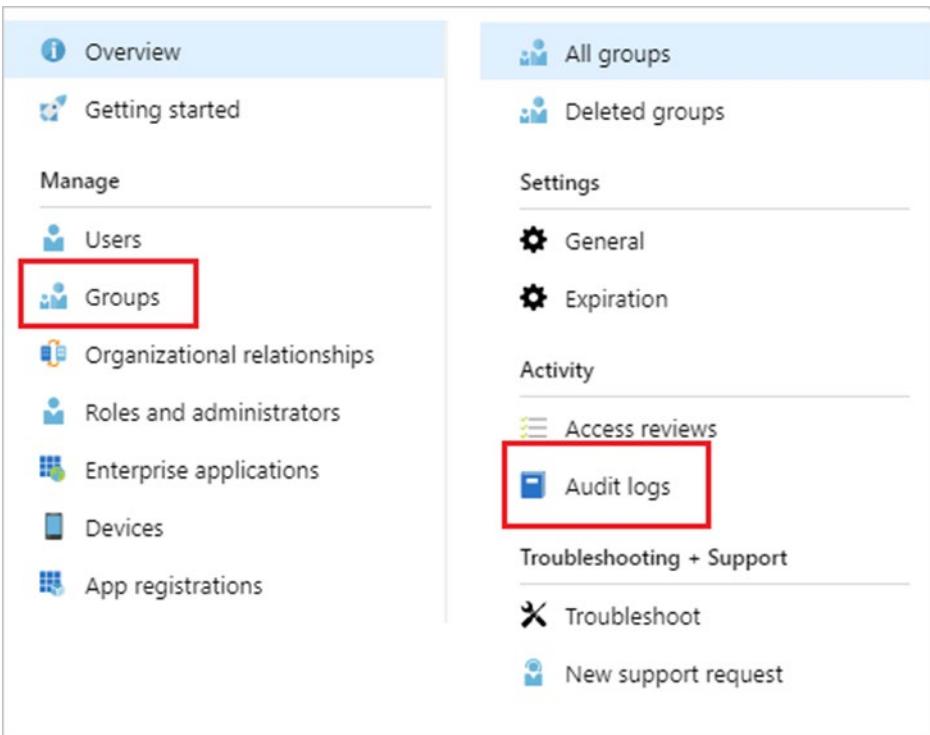
- What types of updates have been applied to users?
- How many users were changed?

- How many passwords were changed?
- What has an administrator done in a directory?
- What are the groups that have been added?
- Are there groups with membership changes?
- Have the owners of a group been changed?
- What licenses have been assigned to a group or a user?

If you want to review only auditing data that is related to users, you can find a filtered view under **Audit logs** in the **Monitoring** section of the **Users** tab. This entry point has **UserManagement** as preselected category.



If you want to review only auditing data that is related to groups, you can find a filtered view under **Audit logs** in the **Monitoring** section of the **Groups** tab. This entry point has **GroupManagement** as preselected category.



Enterprise applications audit logs

With application-based audit reports, you can get answers to questions such as:

- What applications have been added or updated?
- What applications have been removed?
- Has a service principal for an application changed?
- Have the names of applications been changed?
- Who gave consent to an application?

If you want to review audit data related to your applications, you can find a filtered view under **Audit logs** in the **Activity** section of the **Enterprise applications** blade. This entry point has **Enterprise applications** preselected as the **Application Type**.

The screenshot shows the Microsoft 365 Admin Center navigation menu. On the left, under 'Manage', there are links for Overview, Getting started, Users, Groups, Organizational relationships, Roles and administrators, Enterprise applications (which is highlighted with a red box), Devices, and App registrations. On the right, under 'Manage', there are links for All applications (which is highlighted with a blue bar at the top), Application proxy, User settings, Conditional Access, Sign-ins, Audit logs (which is highlighted with a red box), and Access reviews.

Microsoft 365 activity logs

You can view Microsoft 365 activity logs from the Microsoft 365 admin center. Even though Microsoft 365 activity and Azure AD activity logs share a lot of the directory resources, only the Microsoft 365 admin center provides a full view of the Microsoft 365 activity logs. You can also access the Microsoft 365 activity logs programmatically by using the Office 365 Management APIs.

Export Logs to Sentinel and Third-Party Security Tools

What is Azure Sentinel

The screenshot shows the Azure Sentinel log search interface. At the top, there is a breadcrumb trail: Home > Azure Sentinel. Below it, there are filter options: Subscription == Visual Studio Enterprise, Resource group == all, and Location == all. The main area displays a message: 'Showing 0 to 0 of 0 records.' At the bottom, there are sorting options: Name ↑↓ and Resource group ↑↓.

Microsoft Azure Sentinel is a scalable, cloud-native, security information event management (SIEM) and security orchestration automated response (SOAR) solution. Azure Sentinel is your birds-eye view across the enterprise alleviating the stress of increasingly sophisticated attacks, increasing volumes of alerts, and long resolution time frames.

- Collect data at cloud scale across all users, devices, applications, and infrastructure, both on-premises and in multiple clouds

- Detect previously undetected threats, and minimize false positives using Microsoft's analytics and unparalleled threat intelligence
- Investigate threats with artificial intelligence, and hunt for suspicious activities at scale, tapping into years of cyber security work at Microsoft
- Respond to incidents rapidly with built-in orchestration and automation of common tasks

Note - we don't cover it here, but there is a lab on configuring Azure Sentinel.

What do you need to use Azure Sentinel?

- Any Azure AD license (Free/O365/P1/P2) is sufficient to ingest sign-in logs into Azure Sentinel. Additional per-gigabyte charges may apply for Azure Monitor (Log Analytics) and Azure Sentinel.
- Your user must be assigned the Azure Sentinel Contributor role on the workspace.
- Your user must be assigned the Global Administrator or Security Administrator roles on the tenant you want to stream the logs from.
- Your user must have read and write permissions to the Azure AD diagnostic settings to be able to see the connection status.

Using third-party SIEM and SOAR tools

Since the introduction of Azure Monitor, significant strides have been made to consolidate Azure services onto a single logging pipeline. Most of the top Azure services, including Azure Resource Manager and Azure Security Center, have onboarded to Azure Monitor and are producing relevant security logs.

The integration process has also been simplified with key capabilities like security information and event management (SIEM) tools, such as routing data to a single event hub and enabling multiple diagnostic settings per resource. Work in flight will ease setup and management of log routing across large Azure environments.

Azure has also partnered with the top SIEM partners to build connectors that get the data from Azure Monitor into those tools. These connectors consume data routed to Azure Event Hubs by Azure Monitor—a simple, scalable, and manageable approach for delivering log data to an external application, and Microsoft's recommended approach for integrating Azure with SIEM tools going forward.

We've continued to support customers who are using the Azure Log Integration tool (AzLog) to integrate with these same SIEMs. AzLog was initially released to help customers navigate the complex process of consolidating, translating, and forwarding logs from a variety of Azure services to a SIEM tool. At the time, Azure Monitor didn't exist, and there was very little standardization in terms of how Azure services exposed log data to customers. Some dumped data into a storage account, others exposed an API, etc.

Integration recommendations

The table below indicates what you should do based on the SIEM tool(s) you are using and your current integration status. Only SIEM tools that were officially supported by AzLog are included below.

SIEM Tool	Currently using log integrator	Currently investigating SIEM integration options
Splunk	Begin migrating to the Azure Monitor Add-On for Splunk.	Use the Azure Monitor Add-On for Splunk.

SIEM Tool	Currently using log integrator	Currently investigating SIEM integration options
IBM QRadar	Begin migrating to the Microsoft Azure DSM and Microsoft Azure Event Hub Protocol, available from the IBM support website.	Use the Microsoft Azure DSM and Microsoft Azure Event Hub Protocol, available from the IBM support website. You can learn more about the integration with Azure.
ArcSight	The Azure log integration tool offered collecting Azure logs into JSON files for the purpose of integrating with ArcSight using existing JSON connectors from ArcSight, with JSON to CEF mapping available only for Azure Activity Logs and not for the other types of Azure logs.	

Integration roadmap

Today, Azure Monitor's SIEM integration capabilities can't do everything the Azure Log Integration tool could do. Below is our roadmap for addressing known gaps between what you could accomplish with Azure Log Integration and what you can accomplish with Azure Monitor.

Azure Active Directory logs – Azure Active Directory logs are the only log type directly integrated with AzLog that aren't yet available in Azure Monitor.

Integrate Azure VM logs – AzLog provided the option to integrate your Azure VM guest operating system logs (e.g., Windows Security Events) with select SIEMs. Azure Monitor has agents available for Linux and Windows that are capable of routing OS logs to an event hub, but end-to-end integration with SIEMs is nontrivial.

End-to-end setup – AzLog has a script that automates the end-to-end setup of log sources. While Azure Monitor offers the ability to script out creation of diagnostic settings, we're partnering with the Azure Policy team to deliver seamless enablement via Resource Manager policies that ensure log data is being routed from all sources.

Integration with other SIEM tools – AzLog provided a generic capability to push standardized Azure logs in JSON format to disk. While other SIEM tools weren't officially supported by AzLog, this offered a way to easily get log data into tools such as LogRhythm. Our recommendation for customers using AzLog for these tools is to work with the producer of that tool to provide an Azure Monitor Event Hubs integration.

The security of your Azure environment is always top priority on the Azure team, both in terms of how we engineer the Azure platform and in terms of the capabilities we provide for you for securing your own assets on that platform. Moving SIEM integration to Azure Monitor is a step towards enabling you to manageably secure your applications on Azure at scale.

Analyze Azure AD Workbooks and Reporting

With the usage and insights report, you can get an application-centric view of your sign-in data. You can find answers to the following questions:

- What are the most used applications in my organization?

- What applications have the most failed sign-ins?
- What are the top sign-in errors for each application?

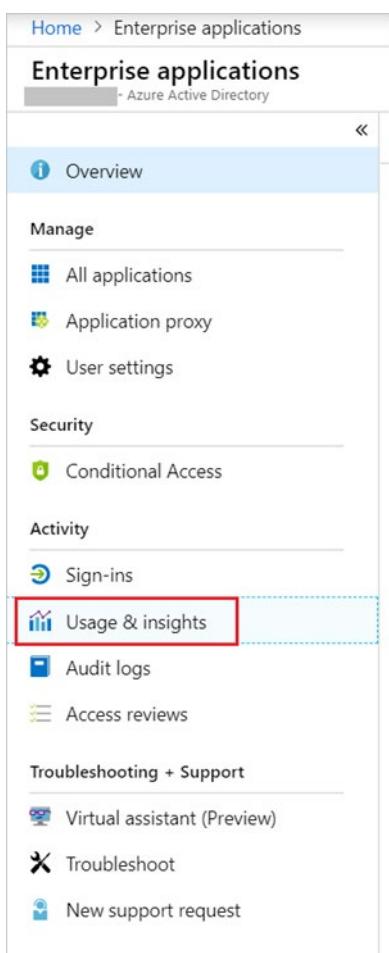
Prerequisites

To access the data from the usage and insights report, you need:

- An Azure AD tenant.
- An Azure AD premium (P1/P2) license to view the sign-in data.
- A user in the Global Administrator, Security Administrator, Security Reader or Report Reader roles. In addition, any user (non-admins) can access their own sign-ins.

Access the usage and insights report

1. Navigate to the Azure portal.
2. Select the right directory, then select **Azure Active Directory** and choose **Enterprise applications**.
3. From the **Activity** section, select **Usage & insights** to open the report.



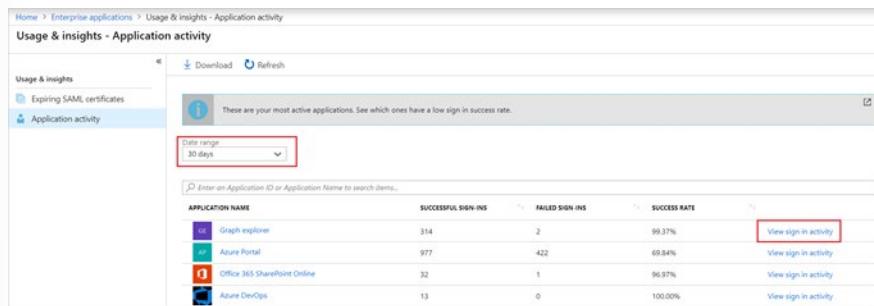
Use the report

The usage and insights report shows the list of applications with one or more sign-in attempts, and allows you to sort by the number of successful sign-ins, failed sign-ins, and the success rate.

Clicking **load more** at the bottom of the list allows you to view additional applications on the page. You can select the date range to view all applications that have been used within the range.

You can also set the focus on a specific application. Select **view sign-in activity** to see the sign-in activity over time for the application, as well as the top errors.

When you select a day in the application usage graph, you get a detailed list of the sign-in activities for the application.



The screenshot shows the 'Usage & insights - Application activity' report in the Azure portal. The left sidebar has 'Usage & insights' selected. The main area displays a message: 'These are your most active applications. See which ones have a low sign in success rate.' Below this is a search bar with 'Date range: 30 days'. A table lists four applications with their sign-in statistics and a 'View sign-in activity' link:

APPLICATION NAME	SUCCESSFUL SIGN-INS	FAILED SIGN-INS	SUCCESS RATE	
Graph explorer	314	2	99.37%	View sign-in activity
Azure Portal	977	422	69.84%	View sign-in activity
Office 365 SharePoint Online	32	1	96.97%	View sign-in activity
Azure DevOps	13	0	100.00%	View sign-in activity

Configure Notifications

The health of an Azure Active Directory Domain Services (Azure AD DS) managed domain is monitored by the Azure platform. The health status page in the Azure portal shows any alerts for the managed domain. To ensure issues are responded to in a timely manner, email notifications can be configured to report on health alerts as soon as they're detected in the Azure AD DS managed domain.

Email notification overview

To alert you of issues with a managed domain, you can configure email notifications. These email notifications specify the managed domain that the alert is present on, and they provide the time of detection and a link to the health page in the Azure portal. You can then follow the provided troubleshooting advice to resolve the issues.

The following example email notification indicates a critical warning or alert was generated on the managed domain:



You have alerts on your managed domain

We detected critical or warning alerts on your Azure Active Directory Domain Services managed domain, on August 21, 2018 14:48 UTC. These issues may negatively affect your service—please resolve them as soon as possible.

To see your alerts and check the health of your managed domain, visit the Health page on the [Azure portal](#), or click the button below.

[View and resolve these alerts >](#)

Why am I receiving this email?

Your email is set up to receive Azure Active Directory Domain Services notifications about your managed domain. You may edit your [notification settings](#) on the Azure portal any time.

Why are there no alerts on my Health page?

Managed domains are checked for alerts every hour. If an alert is resolved, then it disappears from the Health page on the Azure portal. If there are no alerts visible, it could be that someone else resolved your alert or it had been automatically resolved.



[Privacy Statement](#)

Microsoft Corporation, One Microsoft Way, Redmond, WA 98052



Warning - Always make sure that the email comes from a verified Microsoft sender before you click the links in the message. The email notifications always come from the azure-noreply@microsoft.com address.

Why would I receive email notifications?

Azure AD DS sends email notifications for important updates about the managed domain. These notifications are only for urgent issues that impact the service and should be addressed immediately. Each email notification is triggered by an alert on the managed domain. The alerts also appear in the Azure portal and can be viewed on the [Azure AD DS health page¹¹](#).

Azure AD DS doesn't send emails for advertisement, updates, or sales purposes.

When will I receive email notifications?

A notification is sent immediately when a **new alert¹²** is found on a managed domain. If the alert isn't resolved, additional email notifications are sent as a reminder every four days.

¹¹ <https://docs.microsoft.com/azure/active-directory-domain-services/check-health>

¹² <https://docs.microsoft.com/azure/active-directory-domain-services/troubleshoot-alerts>

Who should receive the email notifications?

The list of email recipients for Azure AD DS should be composed of people who are able to administer and make changes to the managed domain. This email list should be thought of as your “first responders” to any alerts and issues.

You can add up to five additional recipients for email notifications. If you want more than five recipients, create a distribution list and add that to the notification list instead.

You can also choose to have all **Global Administrators** of the Azure AD directory and every member of the **AAD DC Administrators** group receive email notifications. Azure AD DS only sends notification to up to 100 email addresses, including the list of Global Administrators and AAD DC Administrators.

Configure email notifications

To review the existing email notification recipients or add additional recipients, complete the following steps:

1. In the Azure portal, search for and select **Azure AD Domain Services**.
2. Select your managed domain, such as aaddscontoso.com.
3. On the left-hand side of the Azure AD DS resource window, select **Notification settings**. The existing recipients for email notifications are shown.
4. To add an email recipient, enter the email address in the additional recipients table.
5. When done, select **Save** on the top-hand navigation.

Warning - When you change the notification settings, the notification settings for the entire managed domain—not just for you—are updated.

Frequently asked questions

I received an email notification for an alert, but when I logged on to the Azure portal there was no alert. What happened?

If an alert is resolved, the alert is cleared from the Azure portal. The most likely reasons are someone else who receives email notifications resolved the alert on the managed domain or the alert was autoresolved by Azure platform.

Why can't I edit the notification settings?

If you're unable to access the notification settings page in the Azure portal, you don't have the permissions to edit the managed domain. Contact a Global Administrator to either get permissions to edit Azure AD DS resources or be removed from the recipient list.

I don't seem to be receiving email notifications even though I provided my email address. Why?

Check your spam or junk folder in your email for the notification and make sure to allow the sender azure-noreply@microsoft.com.

Module 4 Review Questions

Module 4 Review Questions

Review Question 1

What is the purpose of the audit logs?

- Azure AD audit logs provide a comparison of budgeted Azure usage compared to actual.
- Azure AD audit logs provide records of system activities for compliance reporting.
- Azure AD audit logs allow customer to monitor activity when provisioning new services within Azure.

Review Question 2

Can Azure export logging data to third-party SIEM tools?

- Yes, Azure supports exporting log data to several common third-party SIEM tools.
- No, Azure only supports the export to Azure Sentinel.
- Insert the second incorrect answer text in this cell.

Review Question 3

John wants to configure email notifications to be sent from Azure AD Domain Services when issues are detected. In Azure, where this would be configured?

- Azure Microsoft Portal > Azure Active Directory > Monitoring > Notifications > Add email recipient.
- Azure Microsoft Portal > Azure AD Domain Services > Notification settings > Add email recipient.
- Azure Microsoft Portal > Notification Hubs > Azure Active Directory > Add email recipient.

Review Question 4

How can Discovery and insights for privileged identity management help an organization?

- Discovery and insights can find privileged role assignments across Azure AD, and then provide recommendations on how to secure them using Azure AD governance features like Privileged Identity Management (PIM).
- Discovery and insights can find when guest's access resources across Azure AD.
- Discovery and insights can find security group assignments across Azure AD, and then provide recommendations on how to secure them using Azure AD governance features like Privileged Identity Management (PIM).

Review Question 5

Whether to assign a role to a group instead of to individual users is a strategic decision. When planning, consider assigning a role to a group to manage role assignments when the desired outcome is to delegate assigning the role and what else?

- You want to use conditional access policies.
- Many Azure resources need to be managed.
- Many users are assigned to a role.

Review Question 6

Which roles can only be assigned using Privileged Identity Management?

- Permanently active roles.
- Eligible roles.
- Transient roles.

Review Question 7

Who should be engaged when planning a technology project?

- Engage the right stakeholders.
- Start planning with a small team to avoid extra work for others.
- Keep your team small to avoid project creep.

Review Question 8

What is one reason to regularly review Azure role assignments?

- To ensure naming conventions are properly applied.
- To reduce the risk associated with stale role assignments.
- To eliminate extra distribution groups that are no longer used.

Review Question 9

What is an access package?

- An access package is a group of users with the access they need to work on a project or perform a task.
- An access package is a bundle of all the resources with the access a user needs to work on a project or perform their task.
- An access package is used to create a transitive trust between B2B organizations.

Review Question 10

What do catalogs contain?

- Device registrations
- Resources and access packages
- User lists

Review Question 11

How long are deleted users retained by Azure AD by default?

- 14 days
- 30 days
- 60 days

Review Question 12

When should access packages be used?

- To allow one organization access when collaborating on a project.
- An employee requires permanent permissions to perform their job role.
- For access that requires the approval of an employee's manager or other designated individuals.

Module 4 Hands-on Exercises

Lab 25: Create and manage a catalog of resources in Azure AD entitlement management

To download the most recent version of this lab, please visit the SC-300 [GitHub repository¹³](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

A catalog is a container of resources and access packages. You create a catalog when you want to group related resources and access packages. Whoever creates the catalog becomes the first catalog owner. A catalog owner can add additional catalog owners. You must create and configure a catalog in your organization.

Objectives

After you complete this lab, you will be able to:

- Create a catalog
- Add resources to a catalog
- Add additional catalog owners
- Edit a catalog
- Delete a catalog

Lab setup

- Estimated time: 15 minutes

Lab 26: Add terms of use and acceptance reporting

To download the most recent version of this lab, please visit the SC-300 [GitHub repository¹⁴](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

¹³ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

¹⁴ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

Lab scenario

Azure AD terms of use policies provide a simple method that organizations can use to present information to end users. This presentation ensures users see relevant disclaimers for legal or compliance requirements. This article describes how to get started with terms of use (ToU) policies. You must create and enforce a ToU policy for your organization.

Objectives

After you complete this lab, you will be able to:

- Add terms of use
- View report of who has accepted and declined
- What terms of use looks like for users
- How users can review their terms of use
- Edit terms of use details
- Update an existing terms of use document

Lab setup

- Estimated time: 20 minutes

Lab 27: Manage the lifecycle of external users in Azure AD Identity Governance settings

To download the most recent version of this lab, please visit the SC-300 [GitHub repository¹⁵](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

You can select what happens when an external user, who was invited to your directory through an access package request being approved, no longer has any access package assignments. This can happen if the user relinquishes all their access package assignments, or their last access package assignment expires. By default, when an external user no longer has any access package assignments, they are blocked from signing in to your directory. After 30 days, their guest user account is removed from your directory.

Objectives

After you complete this lab, you will be able to:

- Manage the lifecycle of external users in Azure AD Identity Governance settings

¹⁵ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

Lab setup

- Estimated time: 5 minutes

Lab 28: Configure Privileged Identity Management for Azure AD roles

To download the most recent version of this lab, please visit the SC-300 [GitHub repository¹⁶](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

A Privileged role administrator can customize Privileged Identity Management (PIM) in their Azure Active Directory (Azure AD) organization, including changing the experience for a user who is activating an eligible role assignment. You must become familiar with configuring PIM.

Objectives

After you complete this lab, you will be able to:

- Configure Azure AD role settings
- Require approval to activate

Lab setup

- Estimated time: 15 minutes

Lab 29: Assign Azure AD roles in Privileged Identity Management

To download the most recent version of this lab, please visit the SC-300 [GitHub repository¹⁷](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

With Azure Active Directory (Azure AD), a Global administrator can make permanent Azure AD admin role assignments. These role assignments can be created using the Azure portal or using PowerShell commands. The Azure AD Privileged Identity Management (PIM) service also allows Privileged role adminis-

¹⁶ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

¹⁷ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

trators to make permanent admin role assignments. Additionally, Privileged role administrators can make users eligible for Azure AD admin roles. An eligible administrator can activate the role when they need it, and then their permissions expire once they're done.

Objectives

After you complete this lab, you will be able to:

- Assign a role
- Activate your Azure AD roles
- Assign a role with restricted scope
- Update or remove an existing role assignment

Lab setup

- Estimated time: 15 minutes

Lab 30: Assign Azure resource roles in Privileged Identity Management

To download the most recent version of this lab, please visit the SC-300 [GitHub repository¹⁸](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

Azure Active Directory (Azure AD) Privileged Identity Management (PIM) can manage the built-in Azure resource roles, as well as custom roles, including (but not limited to):

- Owner
- User Access Administrator
- Contributor
- Security Admin
- Security Manager

You need to make a user eligible for an Azure resource role.

Objectives

After you complete this lab, you will be able to:

- Assign Azure resource roles
- Update or remove an existing resource role assignment

¹⁸ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

Lab setup

- Estimated time: 10 minutes

Lab 31: Connect data from Azure Active Directory (Azure AD) to Azure Sentinel

To download the most recent version of this lab, please visit the SC-300 [GitHub repository¹⁹](#).

Note: Depending on the lab hosting solution used for your class delivery, the lab instructions may display directly in the virtual lab environment, in addition to being available in the GitHub repository. In that case, follow guidance from your instructor as to where you will access the lab steps. Even if the GitHub version is not used for your class, you may still find it useful to be able to reference the lab instructions on GitHub after class is over.

Lab scenario

Your company expects to begin using a Security information and event management (SIEM) solution. You know you have access to Azure Sentinel and need to become familiar with connecting it to your Azure AD.

Objectives

After you complete this lab, you will be able to:

- Create and add an Azure Sentinel workspace
- Connect to Azure Active Directory

Lab setup

- Estimated time: 10 minutes

¹⁹ <https://github.com/MicrosoftLearning/SC-300-Identity-and-Access-Administrator>

Module 4 Summary

Summary for Module 4

Once you deploy and identity management systems you need to manage and maintain it with a proper governance tools and processes. During this module you explored some of the tools available in Azure to help with this. You can use entitlement management to grant access to new users, while using access reviews to track who is accessing your resources, and what they are doing with them. Finally, using privileged access lets you ensure that admins only have access to privileged resources for limited amount of time, to ensure security. These tools and others allow you to monitor and maintain your Azure AD solution and the resources it accesses.

Plan and Implement Entitlement Management

During this lesson you explored how to:

- Define catalogs.
- Define access packages.
- Plan, implement and manage entitlements.
- Implement and manage terms of use.
- Manage the lifecycle of external users in Azure AD Identity Governance settings.

Plan, Implement, and Manage Access Reviews

During this lesson you explored how to:

- Plan for access reviews.
- Create access reviews for groups and apps.
- Monitor access review findings.
- Manage licenses for access reviews.
- Automate access review management tasks.
- Configure recurring access reviews.

Plan and Implement Privileged Access

During this lesson you explored how to:

- Define a privileged access strategy for administrative users (resources, roles, approvals, thresholds).
- Configure PIM for Azure AD roles.
- Configure PIM for Azure resources.
- Assign roles.
- Manage PIM requests.
- Analyze PIM audit history and reports.
- Create and manage emergency access accounts.

Monitor and Maintain Azure AD

During this lesson you explored how to:

- Analyze and investigate sign-in logs to troubleshoot access issues.
- Review and monitor Azure AD audit logs.
- Enable and integrate Azure AD diagnostic logs with Log Analytics / Azure Sentinel.
- Export sign-in and audit logs to a third-party SIEM tool.
- Review Azure AD activity by using Log Analytics / Azure Sentinel, excluding KQL use.
- Analyze Azure AD workbooks/reporting.
- Configure notifications.

Supplemental Resources

Use these resources to discover more:

- FAQs [https://docs.microsoft.com/azure/active-directory/conditional-access/terms-of-use#frequently-asked-questions²⁰](https://docs.microsoft.com/azure/active-directory/conditional-access/terms-of-use#frequently-asked-questions)

²⁰ <https://docs.microsoft.com/azure/active-directory/conditional-access/terms-of-use>

Answers

Review Question 1

What is the purpose of the audit logs?

- Azure AD audit logs provide a comparison of budgeted Azure usage compared to actual.
- Azure AD audit logs provide records of system activities for compliance reporting.
- Azure AD audit logs allow customer to monitor activity when provisioning new services within Azure.

Explanation

An audit log has a default list view that shows data, like the date and time of the occurrence, the service that logged the occurrence, the category and name of the activity (what), the status of the activity (success or failure), the target, and the initiator/actor (who) of an activity.

Review Question 2

Can Azure export logging data to third-party SIEM tools?

- Yes, Azure supports exporting log data to several common third-party SIEM tools.
- No, Azure only supports the export to Azure Sentinel.
- Insert the second incorrect answer text in this cell.

Explanation

Azure can export to many of the most popular SIEM tools. The most common are Splunk, IBM QRadar, and ArcSight.

Review Question 3

John wants to configure email notifications to be sent from Azure AD Domain Services when issues are detected. In Azure, where this would be configured?

- Azure Microsoft Portal > Azure Active Directory > Monitoring > Notifications > Add email recipient.
- Azure Microsoft Portal > Azure AD Domain Services > Notification settings > Add email recipient.
- Azure Microsoft Portal > Notification Hubs > Azure Active Directory > Add email recipient.

Explanation

The health of an Azure Active Directory Domain Services (Azure AD DS) managed domain is monitored by the Azure platform. The health status page in the Azure Microsoft Portal shows any alerts for the managed domain. To make sure issues are responded to in a timely manner, email notifications can be configured to report on health alerts as soon as they're detected in the Azure AD DS managed domain.

Review Question 4

How can Discovery and insights for privileged identity management help an organization?

- Discovery and insights can find privileged role assignments across Azure AD, and then provide recommendations on how to secure them using Azure AD governance features like Privileged Identity Management (PIM).
- Discovery and insights can find when guest's access resources across Azure AD.
- Discovery and insights can find security group assignments across Azure AD, and then provide recommendations on how to secure them using Azure AD governance features like Privileged Identity Management (PIM).

Explanation

Discovery and insights can find privileged role assignments across Azure AD, and then provide recommendations on how to secure them using Azure AD governance features like Privileged Identity Management (PIM).

Review Question 5

Whether to assign a role to a group instead of to individual users is a strategic decision. When planning, consider assigning a role to a group to manage role assignments when the desired outcome is to delegate assigning the role and what else?

- You want to use conditional access policies.
- Many Azure resources need to be managed.
- Many users are assigned to a role.

Explanation

Management of one group is much easier than management many individual users.

Review Question 6

Which roles can only be assigned using Privileged Identity Management?

- Permanently active roles.
- Eligible roles.
- Transient roles.

Explanation

Permanently active roles are the normal roles assigned through Azure Active Directory and Azure resources while eligible roles can only be assigned in Privileged Identity Management.

Review Question 7

Who should be engaged when planning a technology project?

- Engage the right stakeholders.
- Start planning with a small team to avoid extra work for others.
- Keep your team small to avoid project creep.

Explanation

When technology projects fail, they typically do so due to mismatched expectations on impact, outcomes, and responsibilities. To avoid these pitfalls, ensure that you're engaging the right stakeholders and that project roles are clear.

Review Question 8

What is one reason to regularly review Azure role assignments?

- To ensure naming conventions are properly applied.
- To reduce the risk associated with stale role assignments.
- To eliminate extra distribution groups that are no longer used.

Explanation

You should regularly review access of privileged Azure resource roles to reduce the risk associated with stale role assignment

Review Question 9

What is an access package?

- An access package is a group of users with the access they need to work on a project or perform a task.
- An access package is a bundle of all the resources with the access a user needs to work on a project or perform their task.
- An access package is used to create a transitive trust between B2B organizations.

Explanation

An access package is a bundle of all the resources with the access a user needs to work on a project or perform their task. For example, you may want to create an Access Package that includes all the applications that developers in your organization need, or all applications to which external users should have access.

Review Question 10

What do catalogs contain?

- Device registrations
- Resources and access packages
- User lists

Explanation

Catalogs are used to group related resources and access packages.

Review Question 11

How long are deleted users retained by Azure AD by default?

- 14 days
- 30 days
- 60 days

Explanation

By default, a deleted user is in a deleted state in Azure AD for 30 days, during which time they can be restored by an administrator if necessary.

Review Question 12

When should access packages be used?

- To allow one organization access when collaborating on a project.
- An employee requires permanent permissions to perform their job role.
- For access that requires the approval of an employee's manager or other designated individuals.

Explanation

Two or more organizations are collaborating on a project, and as a result, multiple users from one organization will need to be brought in via Azure AD B2B to access another organization's resources.