Microsoft

From IT Pro to Cloud Pro

Microsoft Office 365 and SharePoint Online

Ben Curry
Brian Laws

Microsoft

From IT Pro to Cloud Pro: Microsoft Office 365 and SharePoint Online

Ben Curry
Brian Laws

PUBLISHED BY
Microsoft Press
A division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number: 2016941108
ISBN: 978-1-5093-0414-1

Printed and bound in the United States of America.

First Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Support at mspinput@microsoft.com. Please tell us what you think of this book at http://aka.ms/tellpress.

This book is provided "as-is" and expresses the author's views and opinions. The views, opinions and information expressed in this book, including URL and other Internet website references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

Acquisitions Editor: Karen Szall
Developmental Editor: Karen Szall
Project Editor: Christian Holdener, S4Carlisle Publishing Services
Editorial Production: S4Carlisle Publishing Services
Technical Reviewer: Charlie Russell; Technical Review services
provided by Content Master, a member of CM Group, Ltd.
Copyeditor: Roger LeBlanc
Indexer: Maureen Johnson, MoJo's Indexing
Cover: Twist Creative • Seattle

I dedicate this book to my family, Kimberly, Madison, and Bryce. They gave up time with "Hubby B" and Daddy to allow this book to make it to you. I love you all as high as the sky!

—Ben Curry

I dedicate this book to my incredible wife (Kathy) and kids (Daniel, Benjamin, and Isabella) for supporting me, giving me up for so long, picking up my slack, and loving me through all of it. They are my greatest blessing.

—Brian Laws

This page intentionally left blank

Contents

Introduction

From the beginning of this project, Brian and I wanted to create a book that would highlight how IT Pros would need to become Cloud Pros. At the same time, we wanted to give newer Microsoft Office 365 professionals a solid guide for implementation. We knew this book couldn't be a comprehensive guide, or it would turn into a large volume that would soon be outdated. Instead, we wanted to keep the book fresh and relevant, and only discuss topics that aren't likely to change or are brand new. Where technologies are likely to change, we referenced the best starting points for you, the reader. Some topics are very technical in nature and highlight the technical skill needed to transition from IT Pro to Cloud Pro. Others, such as migration, are more process-focused and underscore the need for Cloud Pros to be aware of proper project planning and methodologies. The cloud requires us to be both technical and business savvy.

Although it's focused on IT Pros, this book is for anyone who is responsible for designing, configuring, implementing, or managing an Office 365 deployment. This book will help you understand what your team is up against when it comes to your Microsoft SharePoint Online, OneDrive for Business, or hybrid deployment. Also, it will discuss additional technologies and concepts that underlie Office 365 but aren't readily apparent, such as Azure Active Directory (Azure AD), Security & Compliance, and lots of Windows PowerShell. However, this book rarely issues prescriptive guidance—you should use the online Microsoft resources for that. Our goal is to help you think through the various design points so that you can make the right decisions for your company.

This book assumes you have a working knowledge of SharePoint Server 2013 and SharePoint Online administration, or that you have access to that information. It also assumes you have a working knowledge of Office 365 fundamentals, such as how to work with users and navigate the administration centers. We assume you understand most networking concepts, such as DNS, firewalls, routing, and proxy servers, along with the how-to of those concepts. Newer areas of technology (many of which many IT Pros lack experience with or are weak in) are covered in more technical detail than other, better-known topics.

Finally, we often "deep dive" into a single, specific area to show the logic of being a Cloud Pro. Because there isn't room in any book to dive deeply into every area of Office 365, you should pay attention to how, in these deep dives, we go about solving problems and not just focus on how we solve that specific problem.

Acknowledgments

Writing a book requires support from many people. We had several people who were more than willing to help in areas in which we needed additional feedback. Sometimes, we simply wanted a top pro's advice. First, we want to thank the two contributing authors, Jason Batchelor and Jay Simcox. They have both contributed to Microsoft books before, and it was a pleasure to work with them again. They brought expertise in areas that weren't strengths for Brian and me, thus making those sections deeper and more relevant. Next, James Curry helped with developing some PowerShell for the book. As always, his code is stellar!

Matt Whitehorn provided valuable insights into governance and process management. Thanks, Matt. Also, we want to say thanks to Rob de Jong for his input into managing users and groups with Azure AD. We had some graphics advice from Summit 7 Systems' Robin Williams. Thanks to her for helping lay out some challenging sketches from me.

Last, but not least, thanks to Neil Hodgkinson and Manas Biswas for their expert insights into hybrid Office 365 design and configuration. If you ever get the chance to attend their sessions or read their books, you'll be very glad you did.

Free ebooks from Microsoft Press

From technical overviews to in-depth information on special topics, the free ebooks from Microsoft Press cover a wide range of topics. These ebooks are available in PDF, EPUB, and Mobi for Kindle formats, ready for you to download at:

http://aka.ms/mspressfree

Check back often to see what is new!

Errata, updates, & book support

We've made every effort to ensure the accuracy of this book and its companion content. You can access updates to this book—in the form of a list of submitted errata and their related corrections—at:

http://aka.ms/CloudPro/errata

If you discover an error that is not already listed, please submit it to us at the same page.

If you need additional support, email Microsoft Press Book Support at mspinput@microsoft.com.

Please note that product support for Microsoft software and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to *http://support.microsoft.com*.

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

http://aka.ms/tellpress

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter: *http://twitter.com/MicrosoftPress*.

This page intentionally left blank

CHAPTER 3

Working with Azure Active Directory for Office 365

In this chapter, we talk about Microsoft Azure Active Directory (Azure AD) and identity in Microsoft Office 365. First, we discuss the features of Azure AD, compare it to the on-premises Active Directory (AD DS) you're familiar with, and review the various pricing tiers available. Next, we dive into the three identity scenarios: cloud-only, synchronized, and federated.

We then discuss synchronizing your on-premises Active Directory users with your tenant using Azure AD Connect (AD Connect) product. This step is important not only to provide your users with a better sign-in experience but also to enable various hybrid scenarios, which we discuss in a later chapter. Identity synchronization might be the most difficult step of the hybrid journey, but AD Connect is making it easier. Finally, we discuss deploying Active Directory Federation Services (AD FS) to enable the federated identity scenario and give your users single sign-on (SSO).

There is a tremendous amount to discuss on these topics; entire books can be written on them. Because we don't have the luxury of diving deeply into this subject, our focus will be on giving you what you need to understand the concepts and get started with these technologies. There are terrific resources online for diving deeper, and we're not going to insult your intelligence by pretending you can't find them. Instead, we want to equip you to approach these subjects holistically as a Cloud Pro and get your mind around what it takes to cloud-enable your company. Outside of networking, identity management is perhaps the most important and fundamental aspect of cloud computing.

Azure Active Directory

The core of Office 365 is Azure AD. To borrow a phrase from one of our favorite movies (*Star Wars*), Azure AD surrounds Office 365 and penetrates it. It binds the services together. Let's take a closer look.

What is Azure Active Directory?

Azure AD is a Microsoft Azure Platform as a Service (PaaS) offering that is included in every Office 365 tenant. Azure AD is a free Azure service, and any developer can take advantage of it as an identity store. You can access your Azure AD tenant by selecting Azure AD under the Admin Centers menu in the Office 365 Admin Center, as shown in Figure 3-1.

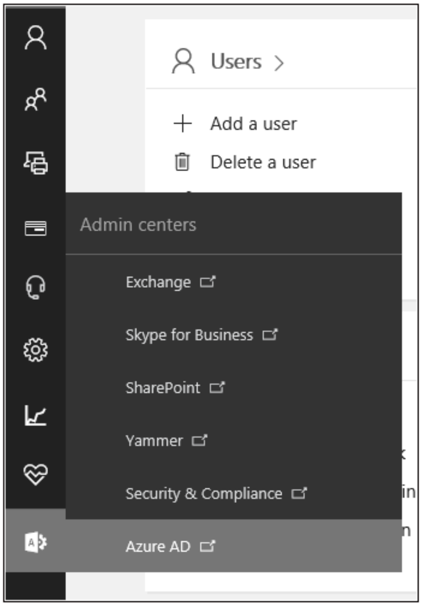
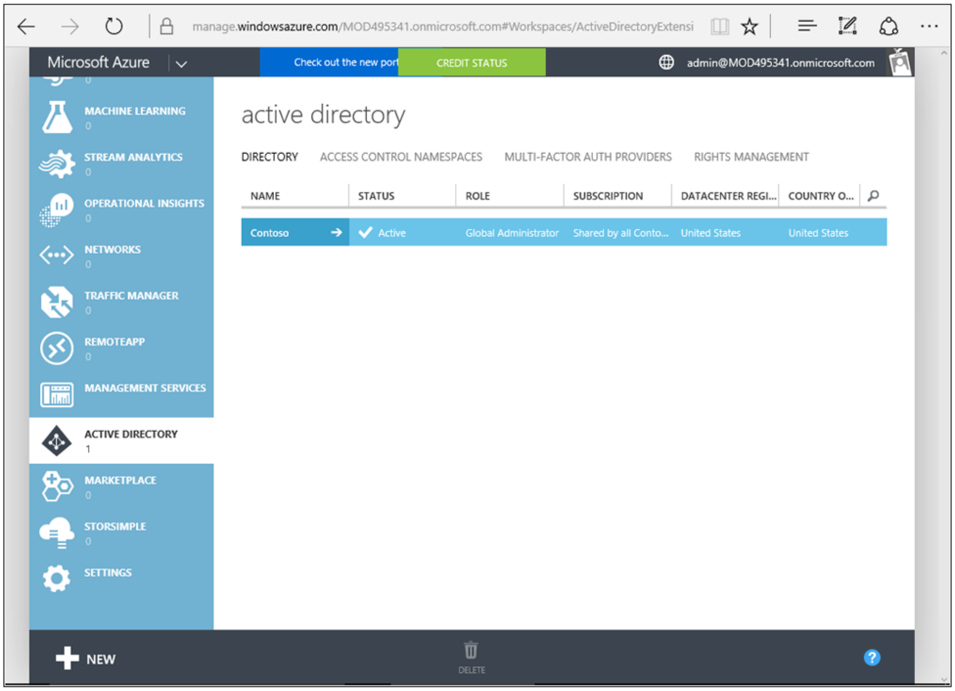


FIGURE 3-1 Accessing your Azure AD tenant

The first time you open Azure AD from the Admin Center, you are prompted to set up your Azure subscription. The subscription is free and only takes a few clicks for you to set it up. After a couple of minutes, the subscription will be ready and you can start exploring. At this point, there is nothing in the Azure subscription other than the single Azure AD directory, but you can add additional Azure services if you want. At the time of this writing, Azure AD is still managed through the old Azure portal. Scroll down in the left pane and select Active Directory. Your directory is shown in the right pane, as you can see in Figure 3-2. Click the cell with your tenant's name to start working with your directory.



NAME	STATUS	ROLE	SUBSCRIPTION	DATACENTER REGI...	COUNTRY O...
Contoso	✓ Active	Global Administrator	Shared by all Conto...	United States	United States

FIGURE 3-2 Accessing your tenant in the Azure portal

You can work with Users, Groups, Applications, and Domains, and you can perform advanced configuration, view the status of your on-premises Active Directory integration, and view reports. You also might want to work with your users and groups in this portal. That's certainly an option, and it is supported. However, chances are good that you'll need to do additional Office 365–related actions with your users and groups, such as assigning licenses, configuring services for the users (like setting up mailboxes), and assigning service-level admin roles (like SharePoint Administrator). Those activities need to be done through the Office 365 Admin Center, so it likely will be more efficient to simply work there. However, working in the Azure AD Admin Center is akin to managing via Microsoft Windows PowerShell, so feel free to use it in that way if it is an experience you prefer.

So what does Azure AD actually do, and why should you care? The various Microsoft cloud services require users to have an identity in order to interact with the service, just as your on-premises systems do. Resources and systems need to be secured from public access. On-premises, we have Active Directory or some other Lightweight Directory Access Protocol (LDAP) directory service for this. The on-premises systems trust the directory and rely on it to identify and manage the users. But what do you do in the cloud, when services are offered by external companies? Enter Azure AD.

Azure AD is the identity store of the Microsoft cloud. If you think of the various cloud services as traditional server applications in an on-premises domain, Azure AD is the Active Directory that these servers are hooked into for their identity needs. Like an on-premises

Active Directory, it holds the user and group objects and manages user authentication. Office 365 services (such as Microsoft SharePoint Online) act like server-based systems and use Azure AD identities for authorization. If you think of Azure AD as the domain controller for the Microsoft cloud, you're not far off.

Azure Active Directory vs. on-premises Active Directory

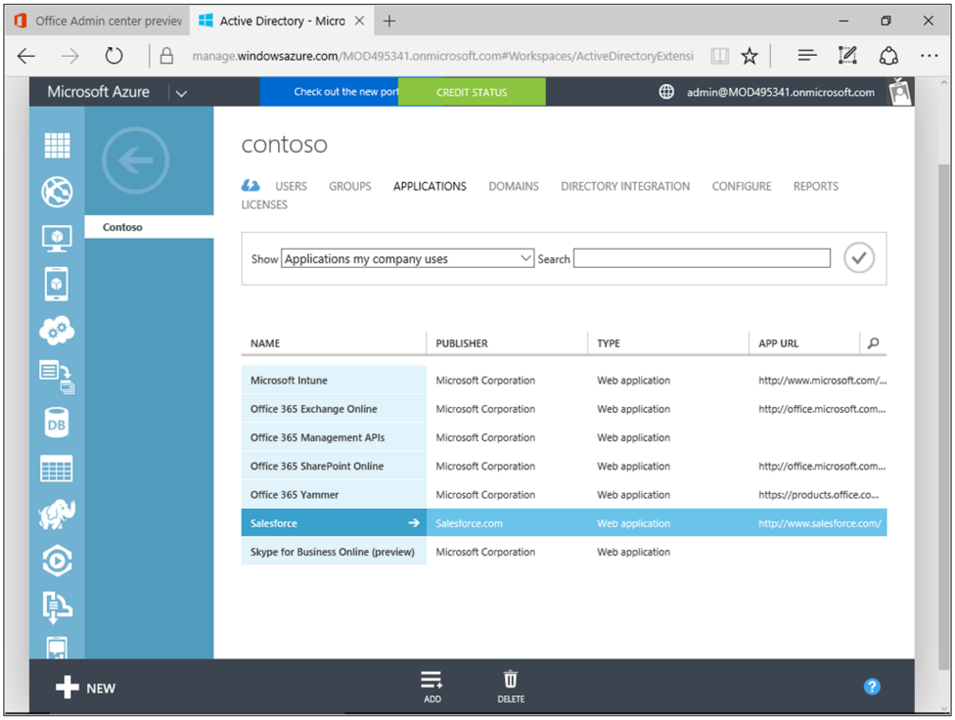
Azure AD is fundamentally different from on-premises Active Directory in that it was born in the cloud. It was designed and built as a tenant service. This means it has all the benefits cloud-native systems have, such as massive scale, high resiliency, and a tenanted service model. Although in many ways it's similar to a traditional Windows Server Active Directory, it has a radically different architecture under the hood.

Both Azure AD and Windows Active Directory manage users, groups, and contacts. There is a data schema that stores properties about the objects stored in the directories. Both have, for example, first name, last name, office location, manager, geographical information, and group membership. Your Azure AD, however, has no concept of organizational units (OUs). It's a flat hierarchy, with each object being a peer to the others. Additionally, until the advent of Azure Active Directory Domain Services (Azure AD DS) (which is discussed in the next section), there really wasn't the concept of a domain that is a boundary encapsulating objects. Instead, Azure AD uses the concept of *tenants*, wherein a company is able to securely manage a collection of objects that it owns. You can almost think of your tenant as a forest, but that's not quite accurate.

Until Azure AD DS (which is an add-on service), there was also no concept of machine objects in Azure AD. Machine objects are critical in Windows Active Directory because they establish membership within a domain and allow such things as Windows Authentication. The normal Azure AD services do not have machine objects; therefore, it's not possible for servers or virtual machines to be joined to Azure AD like an on-premises server would be joined to a local Active Directory instance. Microsoft has begun adding additional services to Azure AD and Office 365, changing the situation a bit. You can, for example, join Windows 10 devices to Azure AD. However, the devices are associated with users and managed through the Mobile Device Management service (Microsoft Intune), not directly joined to Azure AD.

Azure AD provides a number of additional, powerful services that are unavailable in Active Directory. For one, companies can register applications in Azure AD as service principals. These applications can then have permission (if approved) to the suite of Azure services, including users in Azure AD. These applications can be published to users and provide a kind of new distribution model. This is how core Office 365 services like Exchange Online and SharePoint Online work and gain access to Azure AD. But you can also use Azure AD for single sign-on (SSO) to a wide array of third-party cloud services, such as Salesforce. As shown in Figure 3-3, Salesforce has been added to the Azure AD tenant, allowing users to sign in to Salesforce from the Office 365 App Launcher using their Azure AD account.

As a Cloud Pro, you can hook up various Software as a Service (SaaS) applications with Azure AD and give your users a simple sign-on experience with just their one Azure AD account. This can help your users be more efficient and reduce the number of accounts and passwords they need to memorize. Adding these SaaS applications is simple and easy to do.


Office Admin center previev
Active Directory - Micro
manage.windowsazure.com/MOD495341.onmicrosoft.com#Workspaces/ActiveDirectoryExtensi
Microsoft Azure
Check out the new port
CREDIT STATUS
admin@MOD495341.onmicrosoft.com
Contoso
NEW
ADD
DELETE

contoso

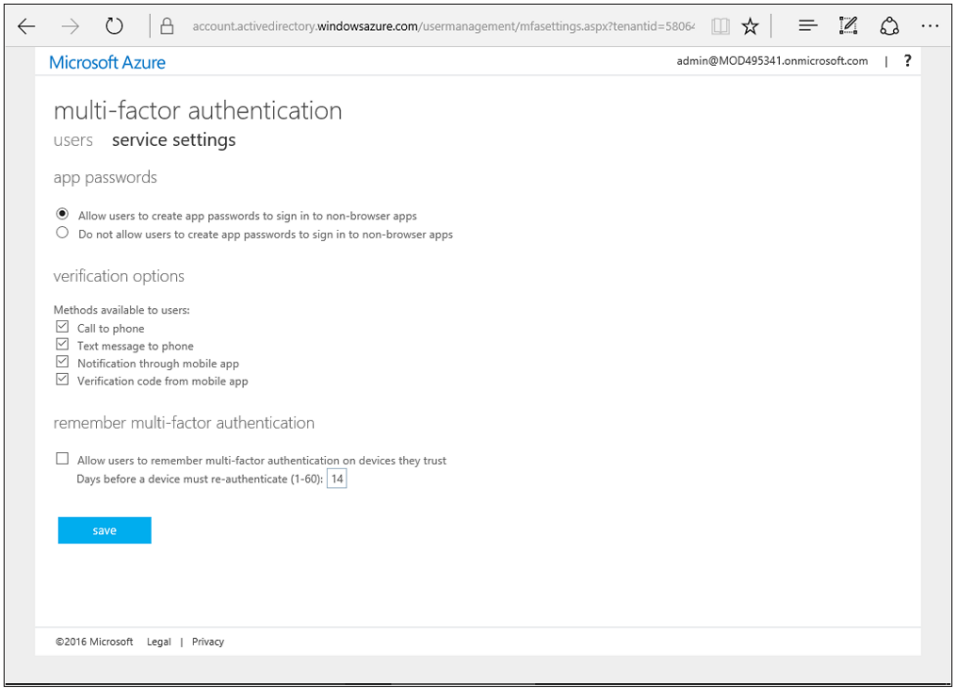
USERS GROUPS APPLICATIONS DOMAINS DIRECTORY INTEGRATION CONFIGURE REPORTS LICENSES

Show Applications my company uses Search

NAME	PUBLISHER	TYPE	APP URL
Microsoft Intune	Microsoft Corporation	Web application	http://www.microsoft.com/...
Office 365 Exchange Online	Microsoft Corporation	Web application	http://office.microsoft.com...
Office 365 Management APIs	Microsoft Corporation	Web application	
Office 365 SharePoint Online	Microsoft Corporation	Web application	http://office.microsoft.com...
Office 365 Yammer	Microsoft Corporation	Web application	https://products.office.co...
Salesforce	Salesforce.com	Web application	http://www.salesforce.com/
Skype for Business Online (preview)	Microsoft Corporation	Web application	

FIGURE 3-3 Salesforce as an application in Azure AD

Additionally, if you purchase Azure Premium, you get advanced features such as multi-factor authentication (MFA). With only a few clicks, you can greatly enhance the security of all systems using Azure AD by implementing MFA. This benefit applies to Exchange Online, SharePoint Online, Skype for Business, OneDrive for Business, Power BI, the administrator portals, and any third-party SaaS application you configured. MFA can even include on-premises applications you publish to Azure AD through the Azure AD Application Proxy (which will be discussed in Chapter 6, “Hybrid Office 365”). The Azure AD Application Proxy can securely publish to the internet that 10-year-old legacy, on-premises, web-based system that everyone is afraid to touch—and the Azure AD Application Proxy can do it without having to deploy Active Directory Federation Services or modifying the code. Figure 3-4 shows the options available in the Azure AD MFA configuration.



account.activedirectory.windowsazure.com/usermanagement/mfasettings.aspx?tenantid=5806

Microsoft Azure

admin@MOD495341.onmicrosoft.com | ?

multi-factor authentication

users   service settings

app passwords

● Allow users to create app passwords to sign in to non-browser apps
○ Do not allow users to create app passwords to sign in to non-browser apps

verification options

Methods available to users:
☑ Call to phone
☑ Text message to phone
☑ Notification through mobile app
☑ Verification code from mobile app

remember multi-factor authentication

□ Allow users to remember multi-factor authentication on devices they trust
Days before a device must re-authenticate (1-60): 14

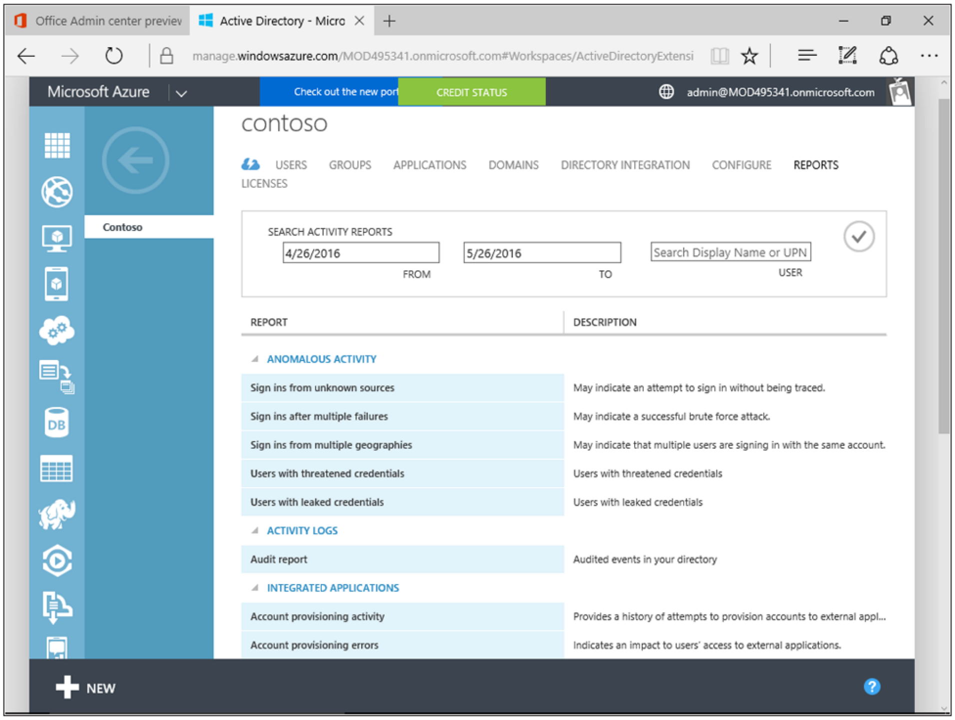
save

©2016 Microsoft   Legal | Privacy

FIGURE 3-4 Azure AD multi-factor authentication settings

Finally (and this list of services has not been exhaustive), Azure AD provides powerful capabilities around threat detection and reporting that is unavailable with on-premises Active Directory. The Azure AD service proactively watches for threats, both known and developing, and uses machine learning to determine whether or not your tenant (or individual users in your tenant) is under attack. If it sees that it is, it can take automatic actions to protect your tenant and your users.

Azure AD also provides you with useful reports that can help you manage your identities. (See Figure 3-5.) For example, there is a report that can identify sign-ins from multiple geographies, which might be an indication that an account has been hacked. Premium Azure AD subscribers receive additional reports, such as one that uses machine learning to determine irregular sign-in activities for individual users. All of this would be extremely difficult, if not impossible, for most companies to do through a traditional on-premises Active Directory instance.



SEARCH ACTIVITY REPORTS

FROM: 4/26/2016
TO: 5/26/2016
USER: Search Display Name or UPN

REPORT	DESCRIPTION
ANOMALOUS ACTIVITY	
Sign ins from unknown sources	May indicate an attempt to sign in without being traced.
Sign ins after multiple failures	May indicate a successful brute force attack.
Sign ins from multiple geographies	May indicate that multiple users are signing in with the same account.
Users with threatened credentials	Users with threatened credentials
Users with leaked credentials	Users with leaked credentials
ACTIVITY LOGS	
Audit report	Audited events in your directory
INTEGRATED APPLICATIONS	
Account provisioning activity	Provides a history of attempts to provision accounts to external appl...
Account provisioning errors	Indicates an impact to users' access to external applications.

FIGURE 3-5 Azure AD reports

Azure Active Directory Domain Services

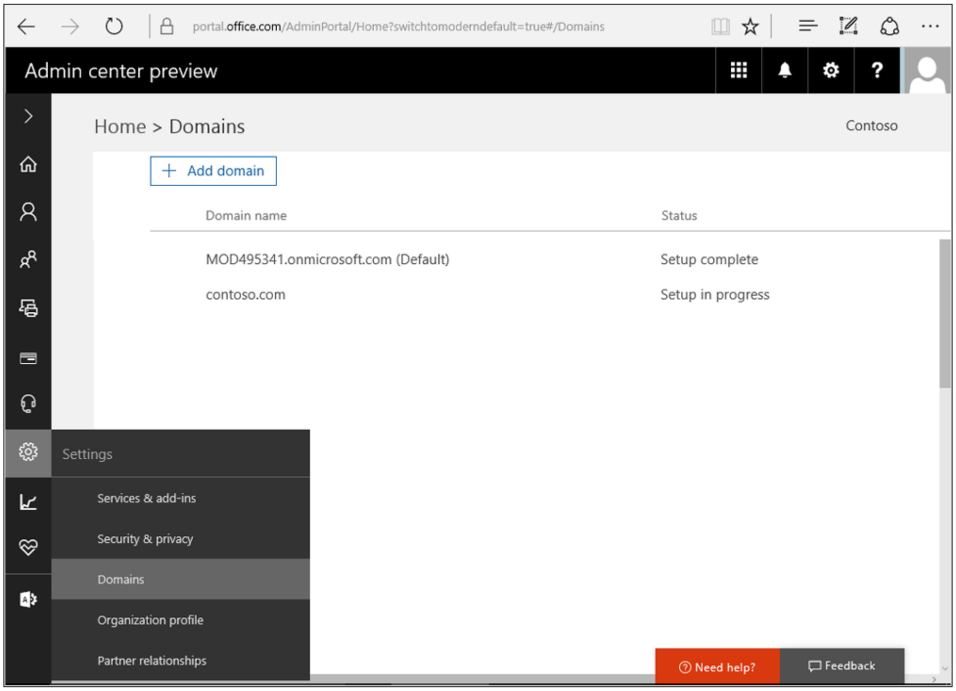
As we mentioned previously, when Azure AD first appeared, there was no concept of a computer object as there is in Active Directory. You were unable to join servers to Azure AD like you would with an on-premises Active Directory instance. In 2016, Microsoft added a new add-on service called *Azure Active Directory Domain Services (Azure AD DS)*. This subscription service enables Azure virtual machines (VMs) to join an Azure domain without needing to deploy a domain controller. This service is not used by Office 365, but it's good for you as a Cloud Pro to be aware of this capability in Azure AD.

When you sign up for Azure AD DS, Microsoft creates a domain that your Azure VMs can then join. Only Azure VMs can join the domain; it's not possible to join on-premises servers. Like Azure AD, it is a tenant service, so you do not have complete ownership of it. You cannot have domain administrator privileges, for example, and you're restricted in what you can do. However, it can be an excellent way to shift applications to Azure without having to configure and support a domain-controller infrastructure. Although you cannot set up a trust with your on-premises domains, you can use AD Connect to synchronize your domain to Azure AD. Once your domain is synced, you can then use your corporate credentials to access resources and services hosted in Azure VMs.

Custom domains

When you first sign up for Office 365, you choose your tenant name. You need to choose this well, because it will be the basis for some important things going forward. For example, your SharePoint Online URL will be based on it (*https://*<tenant>*.sharepoint.com*). You won't be able to change it after it has been set. This will be your default tenant name, and all users will be given a user name based on it. They're always in the format of <tenant>*.onmicrosoft.com*. In the examples we've used so far, we've used *mod495341.onmicrosoft.com* (from the Microsoft Demos service available to Microsoft partners and staff).

The vast majority of the time, you'll want to add your own custom domain name. Although you absolutely can (and some people do) simply use the *onmicrosoft.com* address, it will likely make more sense to associate it with your company's domain. For example, if your company is named Contoso and uses *contoso.com*, you'll want your users sending email using an *@contoso.com* address. To do this, you need to add it to your tenant as a custom domain. You do so by selecting Domains under the Settings menu in the Office 365 Admin Center, as shown in Figure 3-6.

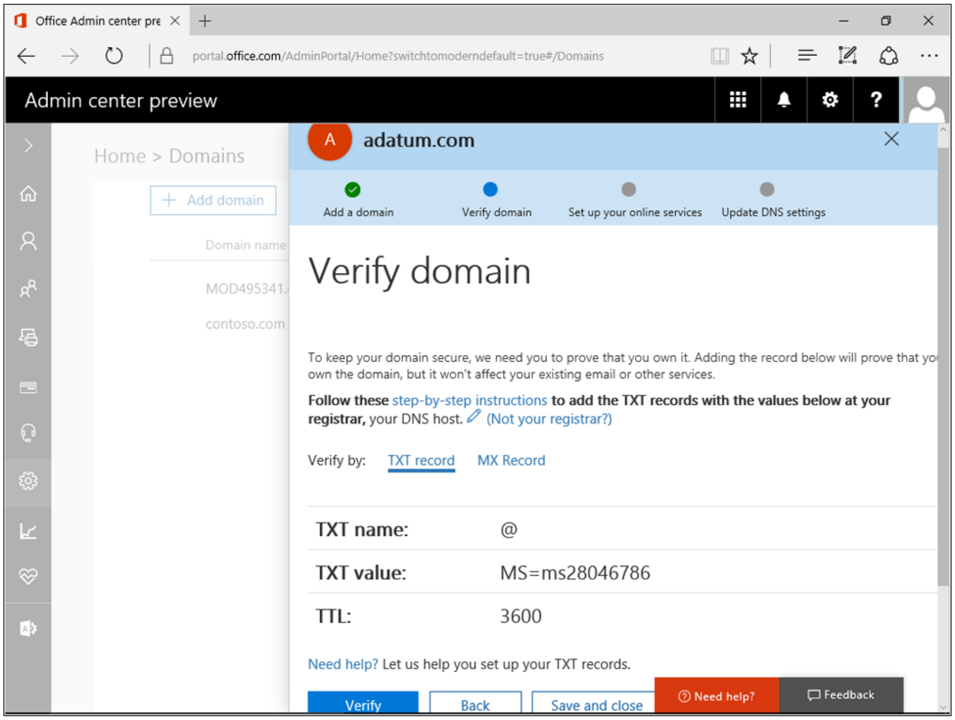


Domain name	Status
MOD495341.onmicrosoft.com (Default)	Setup complete
contoso.com	Setup in progress

FIGURE 3-6 Accessing your Office 365 domains

To add a custom domain to Office 365, you must have ownership of it. You also have to verify that ownership. This means you must be able to add and modify DNS records for the domain at your domain registrar. The Add Domain Wizard in the Admin Center will tell you what you need to do to verify ownership.

In the following example, you'll see that we're adding *adatum.com* as a custom domain. To verify the domain, you need to create a TXT or MX record in your DNS. As shown in Figure 3-7, the wizard will tell you exactly what to add. (It will be different for each domain.) After you do this and give DNS a little time to propagate, click the Verify button.



Verify domain

To keep your domain secure, we need you to prove that you own it. Adding the record below will prove that yo own the domain, but it won't affect your existing email or other services.

Follow these step-by-step instructions **to add the TXT records with the values below at your registrar,** your DNS host. (Not your registrar?)

Verify by: TXT record MX Record

TXT name:	@
TXT value:	MS=ms28046786
TTL:	3600

Need help? Let us help you set up your TXT records.

FIGURE 3-7 The Verify Domain page of the Add Domain Wizard

After Office 365 verifies the existence of the records in your registrar, it helps you set up the rest of the DNS records you need to support the various Office 365 services. If your domain is hosted with GoDaddy, Microsoft sets up the services automatically for you. If your domain is not hosted with GoDaddy, you can still choose to have Microsoft manage the DNS records. Don't do it, though, if you have a website registered under your DNS name.

We recommend that you choose the option to manually configure your DNS. If you choose this option, Microsoft gives you all the DNS records you need to add to support the Office 365 services. This includes records for email, Exchange Online, Skype for Business, and mobile device management (MDM). Once you get this list of records, add them to your DNS registrar to fully configure your custom domain for Office 365. Do not, however, change your MX record if you still need to keep your current email solution. You should wait to change the MX record until you're ready to migrate your email to Exchange Online.

Although adding a custom domain isn't necessary for most scenarios, it is required for implementing identity federation. This requirement exists because you must be able to add to the domain registrar a DNS record for the public endpoint of your AD FS proxy.

MORE INFO **For more information about how to add a custom domain, see "Add your users and domains to Office 365" at** ***https://support.office.com/article/Add-your-users-and-domain-to-Office-365-6383f56d-3d09-4dcb-9b41-b5f5a5efd611*****.**

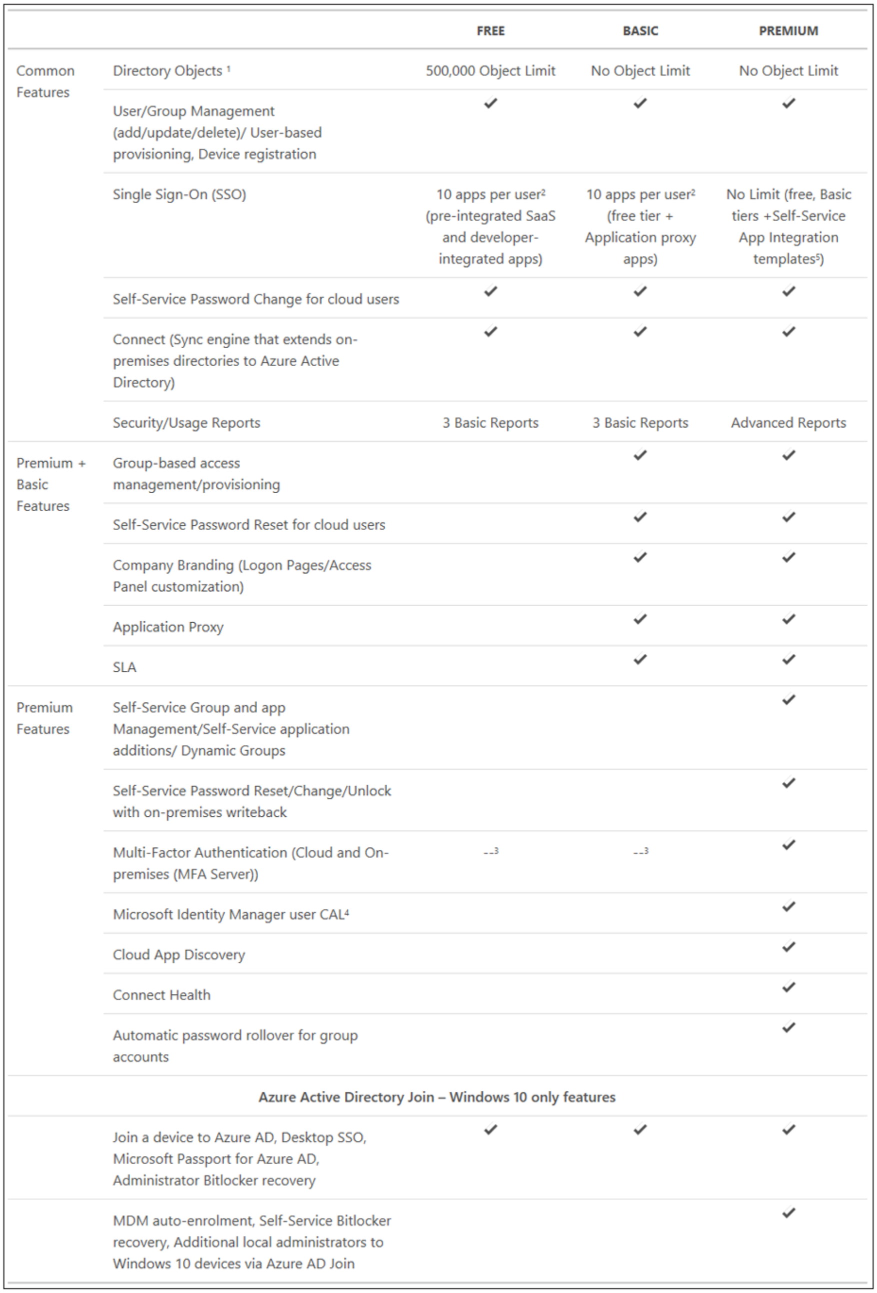
Azure Active Directory service tiers

There are three tiers available for Azure AD: Free, Basic, and Premium. The Free tier is exactly that: free. It is available in all Azure subscriptions, including Office 365. The Free tier will be sufficient for most users because it supports all the basic Office 365 workloads. The Basic and Premium tiers, though, add some powerful capabilities to your tenant.

One of these is the Azure AD Application Proxy (one of our favorites). You use this feature to install a little service on-premises through which you can securely publish your on-premises web-based applications (via Windows or Claims authentication)—all without opening an inbound port in the firewall. You can use all the features of Azure AD, such as MFA, with the published app. With Premium, you get the ability to do self-service password reset, change, and unblock, and have the updated password write back to your on-premises Active Directory. This alone can significantly reduce help-desk calls.

MORE INFO **For more information on using Azure AD to securely access on-premises applications, see "How to provide secure remote access to on-premises applications" at** ***https://azure.microsoft.com/documentation/articles/active-directory-application-proxy-get-started*****.**

Figure 3-8 shows the current set of features available across the three tiers, as taken from *https://azure.microsoft.com/pricing/details/active-directory*. See that webpage for more details.



		FREE	BASIC	PREMIUM
Common Features	Directory Objects [1]	500,000 Object Limit	No Object Limit	No Object Limit
	User/Group Management (add/update/delete)/ User-based provisioning, Device registration	✓	✓	✓
	Single Sign-On (SSO)	10 apps per user[2] (pre-integrated SaaS and developer-integrated apps)	10 apps per user[2] (free tier + Application proxy apps)	No Limit (free, Basic tiers +Self-Service App Integration templates[5])
	Self-Service Password Change for cloud users	✓	✓	✓
	Connect (Sync engine that extends on-premises directories to Azure Active Directory)	✓	✓	✓
	Security/Usage Reports	3 Basic Reports	3 Basic Reports	Advanced Reports
Premium + Basic Features	Group-based access management/provisioning		✓	✓
	Self-Service Password Reset for cloud users		✓	✓
	Company Branding (Logon Pages/Access Panel customization)		✓	✓
	Application Proxy		✓	✓
	SLA		✓	✓
Premium Features	Self-Service Group and app Management/Self-Service application additions/ Dynamic Groups			✓
	Self-Service Password Reset/Change/Unlock with on-premises writeback			✓
	Multi-Factor Authentication (Cloud and On-premises (MFA Server))	--[3]	--[3]	✓
	Microsoft Identity Manager user CAL[4]			✓
	Cloud App Discovery			✓
	Connect Health			✓
	Automatic password rollover for group accounts			✓
	Azure Active Directory Join – Windows 10 only features			
	Join a device to Azure AD, Desktop SSO, Microsoft Passport for Azure AD, Administrator Bitlocker recovery	✓	✓	✓
	MDM auto-enrolment, Self-Service Bitlocker recovery, Additional local administrators to Windows 10 devices via Azure AD Join			✓

FIGURE 3-8 Comparing Azure AD service tiers

At the time of this writing, the retail Basic license is $1 per user per month and the retail Premium license is $6 per user per month. If you own Enterprise Mobility Suite (EMS) licenses, you already have Azure AD Premium licenses. If not, you need to purchase a license for each user who will be using the extra features. If a user somehow receives a benefit from the feature, either directly or indirectly (such as being in a report), that user needs a license. Although you don't *necessarily* have to license every user, you likely will. Carefully consider what your users will be doing (or what will be done for or to them) when determining the number of licenses to purchase.

The Basic and Premium tiers are available through standard means: Microsoft Enterprise Agreement or Open Volume License. You can also purchase licenses through a Cloud Solution Provider partner. Perhaps the easiest way to purchase licenses, though, is to simply go through the Office 365 Admin Center. You'll find Basic and Premium licenses listed as "Microsoft Azure Active Directory Basic" and "Microsoft Azure Active Directory Premium." Previously, Azure AD Basic was not available for purchase online, but we're pleased that it is now an online option.

Identity scenarios

We talked about Azure AD and what it has to offer. Now, let's take the discussion a step back and talk about the broader subject of *identity*. It's a large and important subject, and it's one we'll only scratch the surface of. The topic is critical, though, to the rest of the chapter, and it's incredibly important to understand as a Cloud Pro. Although you're likely familiar with on-premises identities via Active Directory, and although you might have some experience with AD FS and Security Assertion Markup Language (SAML) identities, new identity models and single sign-on (SSO) will become very real, accessible, and needed in the cloud. Identity is what binds disparate cloud services together, and implementing it well can greatly improve the end-user experience.

Before we go further, it will be helpful to define "identity." An *identity* is an object that represents a user. It can also represent a device, application, or service that can act like a user. Typically, this object is registered and owned by a directory of some sort, but this doesn't have to be the case (such as with an anonymous user). An identity is usually associated with a name and a password to allow sign-in to digital systems. Identities are unique within their directories, and this uniqueness is usually ensured by a unique key (like a user principal name, GUID, or both). The directory is responsible for providing the means of authenticating the identity, and the digital system is responsible for associating the identity with a resource (authorization). In simplistic terms, just think of an identity as a user account.

Cloud-only identities

The first and simplest identity scenario in the cloud is *cloud-only identities*. This model resembles the one you're already familiar with on-premises. The concept applies to virtually any cloud service, but we'll narrow our focus to Azure AD. With cloud-only identities, users are given a user account in Azure AD. It is the directory of record for the identity. It has lots of properties related to it, such as the person's name, manager, email address, phone number, and other useful details. The difference, though, is that the identities exist only in the cloud.

When you sign up for Office 365 and create users, these are cloud-only identities. They are uniquely identified in Office 365 by the User Principal Name (UPN), which is used to sign in to Office 365. The default UPN is a <tenant>.*onmicrosoft.com* user name, but you can change it to your custom domain if you add one. Typically, the UPN will look like an email address (*username@tenant.microsoft.com* or *username@contoso.com*), but the UPN doesn't actually have to be the user's email address. It can be simpler and easier for users to work with if it is, though.

Any system that trusts Azure AD as an identity provider can use this identity to gain access. All of the Microsoft cloud trusts Azure AD, meaning you can use your Azure AD account to access any number of Microsoft cloud services. When you access SharePoint Online, for example, you do so using your Azure AD identity.

Again, in a cloud-only identity scenario, a user has an account only in the cloud. The user does not have an on-premises identity because that user might not be using Active Directory.

Synchronized identities

Most companies have an on-premises instance of Active Directory (or other directory). This Active Directory instance is usually the company's system of record for identities. When a company enables Office 365 for its users, it faces an identity dilemma. By default, Office 365 doesn't know anything about the company's on-premises implementation of Active Directory and doesn't allow users to sign in with their Active Directory account. Therefore, a company needs to create accounts in Office 365 for its users. As a result, users have two accounts to manage, including two passwords to memorize. The users will use their Active Directory account credentials on-premises and their Azure AD account in Office 365. This situation can be made even more confusing when the user signs in to both accounts using the same user name but then the passwords don't match.

Thankfully, this issue is not difficult to solve. You do so using *synchronized identities*. In this scenario, the on-premises Active Directory instance is still the master of identities, but a service runs to regularly push on-premises user data to Azure AD. All changes are made on-premises and the service, AD Connect, runs regularly to make sure Azure AD looks the same as the on-premises Active Directory. This synchronization process can include passwords. Technically, the passwords are not synchronized with Azure AD, just the password hashes. If you choose to include the password hash, your users will be able to sign in to both environments using what looks to them like the same credentials with the same password. Syncing the password hashes is not required, though, and you might have a policy that prohibits it. However, we highly recommend that password sync be implemented.

Note that this is not the same as single sign-on (SSO). It's frequently called *same sign-on*. With SSO, a single identity is used to access multiple systems. With same sign-on, although it might look like they're only using one identity, the user actually has two separate identities in two separate directories. The magic of synchronization, though, hides this reality from them. The drawbacks are that because there are two accounts, there technically can be additional maintenance and the potential for identity drift. Tools are available that can help you avoid these drawbacks. Unfortunately, we don’t have the space in this book for a deep dive into the technical implications of same sign-on vs. SSO.

Federated identities

The next scenario involves *federated identities*. This is the single sign-on model you likely have heard much about. It is far and away the most complicated (and costly) model, but its coolness and utility is commensurate with that cost.

With federated identities, one directory trusts another one for authentication. More specifically, it trusts the tokens that the other one provides. We’ll get to tokens in a moment. Although it’s not an approach required in all federated identity scenarios, Office 365 layers the federated identity on top of synchronized identities. In its case, Office 365 has all the information about the user and is able to leverage that information to provide rich experiences when the user interacts with its various services.

The big difference between federated identities and synchronized identities with Office 365 is that Office 365 does not do the authentication. Instead, it relies on an external federated-identity provider to do the authentication. This provider most commonly is a customer’s Active Directory Federation Service (AD FS), although third-party federation services are supported as well. Our focus is going to be on AD FS.

When an AD FS farm is deployed, it’s connected to the on-premises Active Directory. When a user signs in to Office 365, instead of using the regular sign-in page, the user actually is redirected to the company’s AD FS servers, where she then signs in. This user is signing in to the on-premises domain. The AD FS server then redirects the user back to Office 365, except this time it assigns a security token telling Office 365 that the user has successfully been authenticated. Because a trust has been established (technically, they both trust the same certificate that signs the token), Office 365 lets the user through to wherever she was trying to go. Office 365 has delegated the authentication to the on-premises Active Directory.

REAL WORLD **Not everybody will be using AD FS with Office 365 for identity federation. At this time, Microsoft supports federation with 18 other federation providers in addition to AD FS. To see a list of list of these providers and a description of what support exists for clients, see the “Azure AD federation compatibility list” at *https://azure.microsoft.com/documentation/articles/active-directory-aadconnect-federation-compatibility.***

The security token that AD FS creates and sends to Office 365 is a short-lived cookie that is stored in the user’s session. By default, the lifetime for an AD FS security token is 60 minutes long. This security token is actually an SAML token that makes certain claims about the user. A *claim* is a piece of information about the user, like the user’s User Principal Name (UPN), name, and group membership. The receiving system takes a look at the token, verifies that it is signed by the correct certificate and, because it also trusts that certificate, allows the user into the system.

Think of the token as if it was a passport. To enter another country, you must provide proof of your identity. It must be in a specific format that countries will recognize as official. Before letting you into the country, the border official will ask you for your passport. The official checks to make sure it's valid and that you look like the picture on the passport. If everything checks out, the border official allows you into the country. If not, you're turned away and are unable to enter the country legally.

The process is depicted in Figure 3-9. In step 1, the border official stops you from reaching your destination. You've previously received your passport from your home government, as shown in step 2. In step 3, you present your password to the border official, who then lets you through.

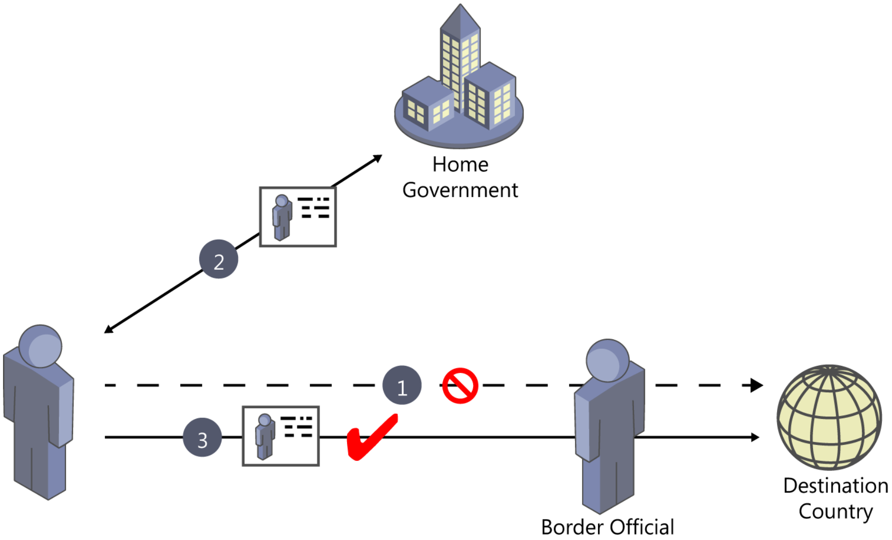


FIGURE 3-9 An illustration of border security requiring a passport

With identity federation and SAML claim-based authentication, the security token is like the passport. This is depicted in Figure 3-10.

The user asks to get a document in Office 365, just as a traveler asks to enter a country. Like the border official, Office 365 asks for proof of identity before it allows the transaction to occur (step 1). The user goes to AD FS to get the token (step 2), like the traveler in the previous illustration went to her government for a passport. AD FS authenticates the user against Active Directory (step 3). The user receives the token from AD FS (step 4), which is the equivalent to the passport, and presents it to Office 365 (step 5). Because Office 365 has been configured to trust the signature of the AD FS that issued it (just as the border official trusts the country that issued the passport), the security token is recognized as official. It then knows about the user based on the claims in the token, such as the user's name and UPN. Office 365 looks in Azure AD for the UPN (step 6), finds a match, associates the user's session with the Azure AD account, and gives the user access to the document (step 7).

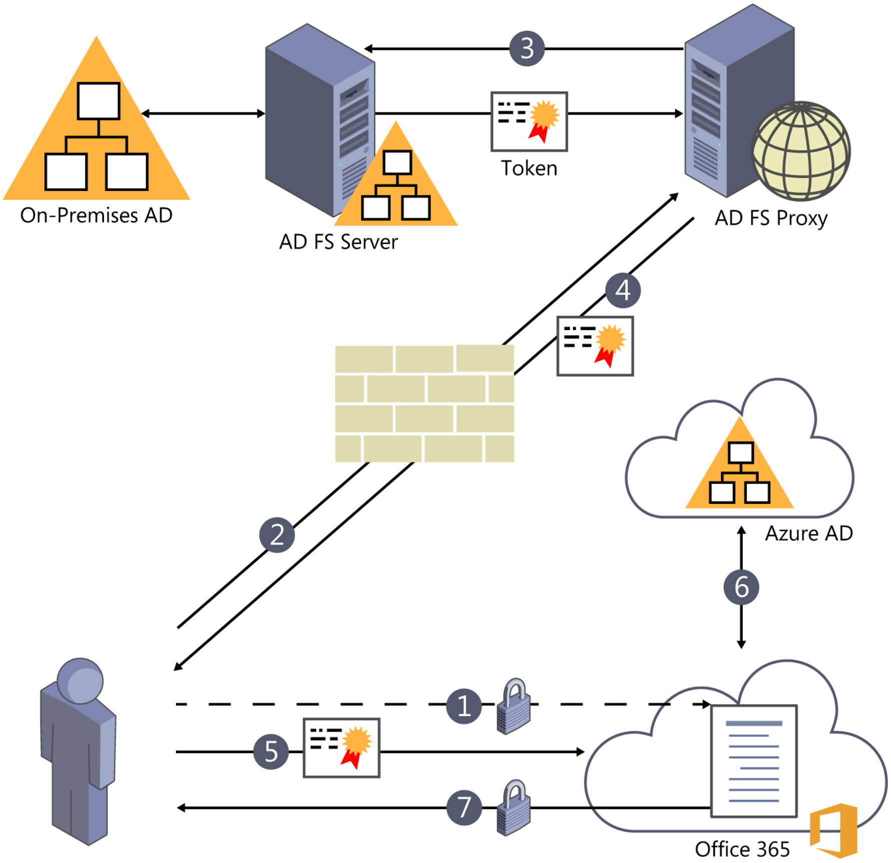


FIGURE 3-10 The federated-identity authorization process

So why bother? Why go through all that extra work, when you can just do identity sync? There are a number of good reasons to do the extra work, and Microsoft outlines some of them in the page "Determine which directory integration scenario to use" at *https://msdn.microsoft.com/library/azure/jj573649.aspx*. Table 3-1 is taken from that article.

TABLE 3-1 Decision matrix for password sync vs. single sign-on

I need to	Dirsync with Password sync	Dirsync with single sign-on
Sync new user, contact, and group accounts created in my on-premises Active Directory to the cloud automatically	Y	Y
Sync incremental updates made to existing accounts in my on-premises Active Directory to the cloud automatically	Y	Y
Set up my tenant for Office 365 hybrid scenarios	Y	Y
Enable my users to sign in and access cloud services using their on-premises password	Y	Y
Reduce password administration costs	Y	Y

I need to	Dirsync with Password sync	Dirsync with single sign-on
Control password policies from my on-premises Active Directory	Y	Y
Enable cloud-based multi-factor authentication solutions	Y	
Enable on-premises multi-factor authentication solutions		Y
Ensure user authentications occur in my on-premises Active Directory		Y
Implement single sign-on using corporate credentials		Y
Customize the user Sign-In page		Y
Limit access to cloud services based on the location, client type, or Exchange endpoint of the client		Y

Companies usually choose AD FS for four primary reasons:

- **Passwords are kept on-site** Because users sign in to AD FS, which is on-site and controlled by the company, external systems don't need to have the users' passwords. Passwords stay within the firewall, which is considered safer. However, with Office 365 and Azure AD, the plain-text password itself is not actually synced, only the password hash.
- **AD FS can provide some additional authentication controls** This approach might include an extra level of filtering, such as basing it on IP addresses. It can, for example, be used to prevent OneDrive for Business from syncing to noncorporate locations. It also can provide MFA, although Azure AD also can do it with Premium licenses.
- **AD FS can be a single-identity provider that can be used by many services, both on-premises services and services in the cloud** Any system that is configured to trust AD FS identities can be accessed by a single log-on. With the token, the user potentially can sign in to hundreds of systems and services but only have to sign in once. This reason is often the primary one for choosing AD FS.
- **AD FS can be configured so that users don't have to sign in to Office 365 when they're using a domain-joined device** Saving users from having to log on to Office 365 can reduce friction and increase productivity.

If you want to implement federated identities, you need to deploy at least one AD FS server and at least one AD FS Proxy server, although you'll likely want two of each to prevent a single point of failure. The AD FS Proxy will often live in your perimeter network because it will require inbound connectivity from the internet. If you already have an AD FS infrastructure in place, you can take advantage of this investment and use it with Office 365. If you do not, AD Connect can easily deploy everything you need through a simple interface. You also need to configure identity sync using AD Connect. Some companies who configure federation choose not to sync the password hashes. This will be especially true for companies with a policy requiring passwords to stay on-premises.

REAL WORLD If you don't have a policy to keep all passwords on-premises and are looking to implement federation, we highly recommend including the password hash when you configure identity sync. Because federation introduces a dependency on the on-premises AD FS system, if that system goes down, users will be unable to sign in to Office 365. If the password hash is synced, you could turn off federation temporarily (or even permanently) and there will be less of an impact on the users. Otherwise, you'll have to assign new passwords to all users if you decide to turn off federation. Including the password hash with your sync is a good tool to have in your back pocket.

Identity sync with Azure Active Directory Connect

We mentioned Azure Active Directory Connect (AD Connect) multiple times, and now it's time to start diving in. Our goal in this section is to equip you to understand the product so that you can make the important design and configuration decisions needed for your organization. You need to build it right the first time and then make sure it runs well.

MORE INFO For details on how to actually install and configure AD Connect, see "Set up directory synchronization for Office 365," at *https://support.office.com/article/Set-up-directory-synchronization-for-Office-365-1b3b5318-6977-42ed-b5c7-96fa74b08846*, and "Custom installation of Azure AD Connect," at *https://azure.microsoft.com/documentation/articles/active-directory-aadconnect-get-started-custom*.

Azure AD Connect overview

Azure AD Connect is the third generation of tools Microsoft has given us to sync identities with Azure AD. The first one was simply called the Directory Synchronization tool, or DirSync. Although that tool is still supported, DirSync is not released individually anymore. However, because it had such a great impact, you'll still hear the synchronization process referred to as "DirSync" even though AD Connect is being used. After DirSync, Microsoft released Azure AD Sync, or AAD Sync. Although that's still supported, like DirSync, AAD Sync is no longer being updated and will be retired. Both are being deprecated, and support ends on April 13, 2017.

In place of these two tools, Microsoft released AD Connect, which incorporates elements of both and adds some significant capabilities. The most significant component AD Connect adds is a wizard that can deploy, and integrate with, an AD FS farm for federated identities. Excellent.

AD Connect is a standalone service you install on a machine in your organization that will act as the middleman between your on-premises Active Directory and Azure AD. It will regularly read from Active Directory and synchronize users, groups, contacts, distribution lists, and other items with Azure AD. If AD Connect is enabled and you have Azure AD Premium licenses, it can synchronize passwords from Azure AD back to the on-premises Active Directory. AD Connect also can be used to deploy an AD FS farm to support federated identities. AD FS can be quite complicated to deploy, but AD Connect makes it simple through an easy-to-use wizard interface. Finally, it also provides a means to monitor the health of your sync and AD FS. AD Connect Health tracks usage and prevents critical issues from occurring. AD Connect Health is a Premium feature, though.

If you already have an AD FS farm deployed that you'd like to use, you can still use AD Connect. If it's a Windows 2012 R2 AD FS farm, you can integrate it with AD FS. If the farm is AD FS 2.0, you need to hook up the federation manually.

MORE INFO **For an extremely good reference for AD Connect and an index of related material, see the following articles:**

- "Integrating your on-premises identities with Azure Active Directory" at *https://azure.microsoft.com/documentation/articles/active-directory-aadconnect*
- "Azure AD Connect sync: Understand and customize synchronization" at *https://azure.microsoft.com/documentation/articles/active-directory-aadconnectsync-whatis*

Before you deploy AD Connect, you need to understand some important concepts and be aware of key decisions that need to be made. The default settings provided in the tool should be sufficient for most organizations. The tool even includes an Express Settings option that deploys a full sync of all the needed attributes for a single forest, including the password hashes. Before you choose this option, though, make sure you read through the design choices in the next section to ensure that it's a good option for you. If you have a more complicated environment, such as you do when you have multiple Active Directory forests or multiple Office 365 tenants, make sure you review the topology options to understand which are supported.

Design choices

When designing your AD Connect deployment, there are four primary factors to consider. The first factor is your *sourceAnchor*, which uniquely identifies each object. The second is the *userPrincipalName*, which is what a user will use to sign in. The third factor is deciding what to do if your on-premises Active Directory contains a nonroutable domain (like contoso.local). The final factor is related to considerations about the availability of the AD Connect service.

Choosing the right sourceAnchor

If you're familiar with the concepts of relational databases and lookup tables, then think of the sourceAnchor as the primary key for your sync. The sourceAnchor is a single attribute that will be *immutable* for the lifetime of the object, and it's used to uniquely identify an object in both

Active Directory and Azure AD. It's used by the sync to tie the two objects together and, as such, the sourceAnchor must never change.

By default, objectGUID is used, and it's perfect for most environments. However, if you have multiple Active Directory forests and you need to move objects between them, objectGUID is not going to be a good choice. Even if you have a single forest, if you think there's a good chance that your company will be merged with another in the future, you might want to consider another attribute as well. It's important that you choose an attribute that will not change. No, email is not a good option, because some people in the organization might change their names.

You need to be sure to get the sourceAnchor correct the first time you set up AD Connect because it cannot be changed. Make sure you record which attribute you use because you will need to select it again if it becomes necessary to redeploy AD Connect. Ensure that it's recorded as part of your disaster-recovery processes.

> ***MORE INFO*** **For a thorough treatment of the sourceAnchor, see Paul Williams' blog "Windows Azure Active Directory Connector part 3: immutable ID" at *https://blog.msresource.net/2014/03/10/windows-azure-active-directory-connector-part-3-immutable-id*.**

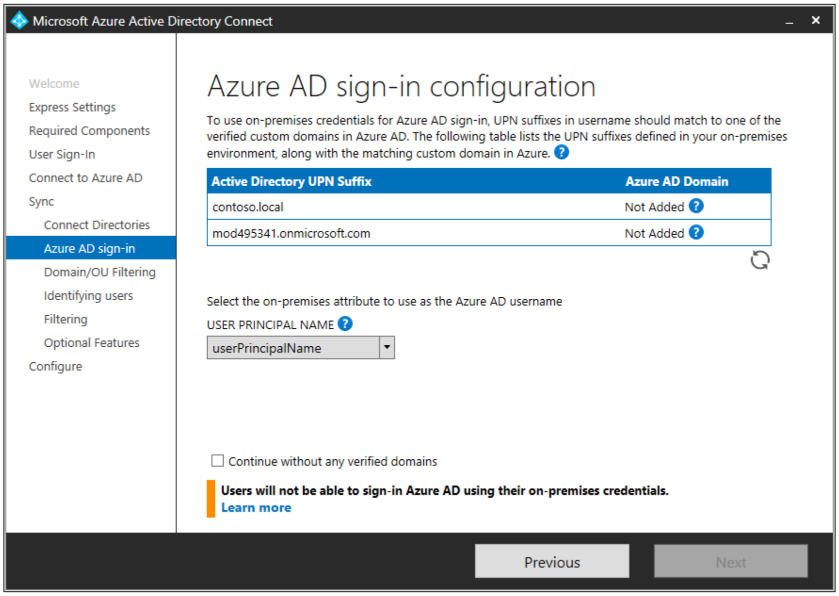
Choosing the right User Principal Name

An early decision to make is how you want your users to sign in to Office 365. Do you want them to use the same login they use to sign in to their computer? Their email address? Do you want them to use AD FS and single sign-on? The answers to these questions dictate what the User Principal Name (UPN) should be in Azure AD.

The UPN is like the sourceAnchor in that it uniquely identifies a user. However, unlike the sourceAnchor, it doesn't have to be immutable and users will need to know it. The UPN is what the users will enter when they sign in, and it comes in the form of *[username]@[domain]*, such as *brian.smith@contoso.com* or *brian.smith@contoso.onmicrosoft.com*. Although it looks like an email address, it does not have to be. Regardless, the UPN must be based on a domain name that is internet-routable.

> ***MORE INFO*** **For more information about the sign-on options and their impact, see "Azure AD Connect User Sign on options" at *https://azure.microsoft.com/documentation/articles/active-directory-aadconnect-user-signin*.**

Each user in your on-premises Active Directory implementation has a UPN. It is what your users use when they sign in to their work computers. By far, it is best if the UPN chosen for Azure AD matches the user's on-premises UPN. If for whatever reason this is not a good option (such as if the on-premises domain is not routable), you have the option of choosing something else instead of the UPN. This is known as an *alternate login ID*. You can set this up in the Azure AD Sign-In Configuration page, shown in Figure 3-11, by changing the value of the User Principal Name. Using an alternate login ID can let you choose something like the email address instead of the UPN.



Azure AD sign-in configuration

To use on-premises credentials for Azure AD sign-in, UPN suffixes in username should match to one of the verified custom domains in Azure AD. The following table lists the UPN suffixes defined in your on-premises environment, along with the matching custom domain in Azure.

Active Directory UPN Suffix	Azure AD Domain
contoso.local	Not Added
mod495341.onmicrosoft.com	Not Added

Select the on-premises attribute to use as the Azure AD username

USER PRINCIPAL NAME: userPrincipalName

□ Continue without any verified domains

Users will not be able to sign-in Azure AD using their on-premises credentials.
Learn more

FIGURE 3-11 Azure AD Sign-In Configuration page in AD Connect

> ***CAUTION*** **Before you choose to use an alternate login ID, understand that it's not compatible with all Office 365 scenarios. The issues are primarily related to Exchange Online hybrid deployments. Also, the Azure AD Application Proxy (a feature available in the Basic and Premium tiers) requires the UPN from Active Directory, so don't select an alternate login ID if you expect to use this feature. Although Microsoft might solve the issues over time, it's by no means a certainty. We recommend you read the following resources if you're considering an alternate login ID:**
>
> - "Configuring Alternate Login ID" at *https://technet.microsoft.com/library/dn659436.aspx*
> - Joe Palarchio's blog "Office 365 – The Limitations of Alternate Login ID" at *http://blogs.perficient.com/microsoft/2015/02/office-365-the-limitations-of-alternate-login-id*

Dealing with a nonroutable domain

For users to sign in to Office 365, they must use a login ID that is associated with a domain registered and validated with Office 365. This ID can be either the default <tenant>.*onmicrosoft.com* domain or a custom domain (like *contoso.com*). For a lot of companies, their Active Directory domains were not set up that way. They might have a nonroutable domain name, like *contoso.local*. If this is the case for your environment, don't worry—hope is not lost. You have several options to choose from:

- **Change your domain name** Frankly, this option is not realistic for most organizations. Changing your domain name can be a nightmare. If, however, you're about to create a new domain or you're just starting one, now is your opportunity to change it so that it's routable.
- **Use an alternate login ID** As discussed earlier in the section about UPNs, you can opt to use a property other than the UPN (email address, for example). However, be aware of the risks if you do so.
- **Change the UPN of the on-premises Active Directory instance** This option is likely your best one and will be what a majority of organizations do. With this approach, you add another UPN suffix to your domain using the Active Directory Domains and Trusts snap-in. This UPN suffix matches the domain you want to use in Office 365 (for example, *contoso.com*). Next, you update each user to use the new suffix. Although you can do it manually, you can update all users at once using the Windows PowerShell Set-ADUser cmdlet.

> ***MORE INFO*** **For more information about assigning a new UPN suffix, see "How to prepare a non-routable domain (such as .local domain) for directory synchronization" at *https://support.office.com/article/How-to-prepare-a-non-routable-domain-such-as-local-domain-for-directory-synchronization-e7968303-c234-46c4-b8b0-b5c93c6d57a7*.**

- **Do nothing** Technically, you do not have to address the issue at all. If not, users will just be synced using the *default @<tenant>.onmicrosoft.com*. Although the sync will work, expect that some Office 365 scenarios may not work as expected, such as SharePoint hybrid. You also will not be able to use single sign-on with AD FS.

You can change the UPN at a future time. However, doing so is complicated and will affect your users, so it's best to choose wisely up front.

Availability

As a Cloud Pro, you know that it's important that services remain available. It's imperative that you give your users no reason to balk at adopting cloud technologies. As such, we would be remiss if we did not touch on the topic of high availability for AD Connect.

Unfortunately, you don't have the options for high availability for AD Connect that you do with most of Microsoft's current server technology. You're limited to running only a single instance of the sync for a given tenant. There are, however, several ways to work around this limitation to maximize uptime:

- **Make SQL Server highly available** By default, AD Connect will install and use SQL Server 2012 Express LocalDB to host its databases. However, you can use a full-featured remote SQL Server instance instead. You can then configure that server for high availability using standard means, such as SQL Server AlwaysOn Availability Groups or Failover Clustering.
- **Deploy a stand-by server in Staging Mode** One of the options you have in the AD Connect Wizard is to place the server in Staging Mode. With it, the server can be fully set up for sync, except that the sync is not actually performed. As a result, a second or third AD Connect instance could be set up (such as at a disaster-recovery location) in Staging Mode and kept prepared for action. If the active instance is lost or needs to be taken offline, you simply run through the wizard on the second server to take it out of Staging Mode.
- **Take advantage of virtualization** Most, if not all, modern virtualization providers have a way to move a virtual machine between hosts, either manually or automatically (such as if there is a physical hardware failure). By simply deploying AD Connect in a virtual machine on such a system, you have a means of moving the service to another host should there be an outage of the physical host.
- **Rebuild** Deploying and configuring AD Connect is quite quick and simple. As long as the settings chosen have been documented (especially the sourceAnchor and UPN), it might be just as easy to redeploy onto a new server. Correctly setting the sourceAnchor allows the sync to reconnect to existing objects and pick up where the last one left off.

MORE INFO **For more information about these availability strategies, see "Azure AD Connect sync: Operational tasks and consideration [sic]" at *https://azure.microsoft.com/documentation/articles/active-directory-aadconnectsync-operations*.**

Topologies

Before you configure anything, you need to evaluate your Active Directory and Office 365 tenants to decide on a suitable topology. There are specific topologies Microsoft supports and some that are not supported. The general rule is you can have only one AD Connect instance per Office 365 tenant. We do not have the space to go into each topology in detail, but if anything below raises a red flag, make sure to review "Topologies for Azure AD Connect" at *https://azure.microsoft.com/documentation/articles/active-directory-aadconnect-topologies*.

Supported topologies

The following topologies are supported by Microsoft:

- **A single forest synchronizing through one AD Connect server to a single Azure AD directory** This scenario is your simplest and most likely. Express Settings in the wizard supports this topology.
- **Multiple forests synchronizing through one AD Connect server to a single Azure AD directory** Note that the AD Connect server does not need to be domain joined. It just needs to be able to communicate with the domain.
- **Multiple forests synchronizing to multiple Azure AD directories, each with their own AD Connect server.**
- **A single forest synchronizing with multiple Azure AD directories through multiple AD Connect servers** In this topology, an object can exist in only one Azure AD directory, so careful filtering must be implemented in each AD Connect instance.
- **Any of the other topologies in this list with one AD Connect server and one or more staging servers** Adding an AD Connect server in Staging Mode can provide increased availability for the service.

Unsupported topologies

The following topologies are not supported by Microsoft:

- **Single or multiple forests synchronizing through multiple AD Connect instances to the same Azure AD directory** An Azure AD tenant can work with only one AD Connect instance.
- **One object synchronizing to multiple Azure AD directories** An object can be synchronized to only one Azure AD directory at a time.

Prepare for sync

After you figure out the architecture of your sync, you need to prepare to implement it. You'll want to verify you can connect to Office 365 and run some scans. You also need to make sure your on-premises Active Directory objects are clean and ready to be synced.

Validate health and connectivity

Obviously, if you're unable to connect to Office 365, you won't be able to sync with it. Your Active Directory must also be healthy. The easiest way to do a simple validation of your environment is by running the health, readiness, and connectivity checks from the server that will host AD Connect. To run these checks, launch a wizard from the AD Connect server by either going to *https://portal.office.com/tools* or walking through the Directory Synchronization Wizard (which is launched from the Users page). The wizard will install the Microsoft Office 365 Support Assistant, which will perform the checks. It will tell you if there are any issues to resolve.

Chapter 1, “Getting started as an Office 365 Cloud Pro,” discusses networking and connectivity, and it provides some additional information on validating and preparing connectivity. It might be a good idea to go back and review this material.

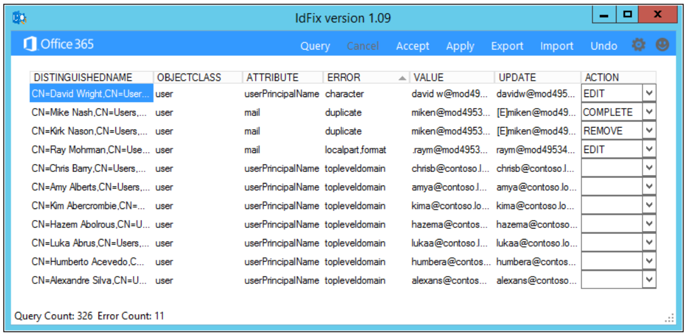
Clean up your Active Directory

If your environment is like most organizations, your Active Directory has been around for many years and has seen many changes. It has likely built up all kinds of cruft. There might be dark, dimly lit recesses of your Active Directory that haven’t been reviewed in years. For the synchronization to work, you need to make sure your Active Directory is as clean as possible.

For many Office 365 projects, this cleanup is often the most time-consuming and difficult part. This cleanup includes making sure that each synchronized user has a unique email address in the *proxyAttribute* attribute, has a valid and unique User Principal Name, and ideally has correct demographic information (such as their name, department, title, and office information). You also want to ensure each user doesn’t have any invalid characters in the synced attributes. If the domain is configured with a nonroutable name, now is the time to address that issue. All of this can be a daunting task.

> ***MORE INFO*** **For more information about what’s involved in preparing the domain for sync, see “Prepare to provision users through directory synchronization to Office 365” at *https://support.office.com/article/Prepare-to-provision-users-through-directory-synchronization-to-Office-365-01920974-9e6f-4331-a370-13aea4e82b3e*.**

Thankfully, Microsoft provides you with a tool to make all of this easier: *IdFix*. This little tool scans your Active Directory and reports any problem it finds. (See Figure 3-12.) Beware—the first time you run it, the results can be quite overwhelming. IdFix makes the cleanup work easier by automatically suggesting fixes to the objects. You can even make the changes right there in the tool. Although you don’t have to run IdFix before you sync, doing so will save you from repeatedly fixing failed syncs. Usually, the process is an iterative one in which you keep chipping away at the list of errors until none are found. When IdFix comes back clean, it’s time to go ahead with the sync.



DISTINGUISHEDNAME	OBJECTCLASS	ATTRIBUTE	ERROR	VALUE	UPDATE	ACTION
CN=David Wright,CN=User...	user	userPrincipalName	character	david w@mod49...	davidw@mod495...	EDIT
CN=Mike Nash,CN=Users,...	user	mail	duplicate	miken@mod4953...	[E]miken@mod49...	COMPLETE
CN=Kirk Nason,CN=Users,...	user	mail	duplicate	miken@mod4953...	[E]miken@mod49...	REMOVE
CN=Ray Mohrman,CN=Use...	user	mail	localpart.format	.raym@mod4953...	raym@mod49534...	EDIT
CN=Chris Barry,CN=Users,...	user	userPrincipalName	topleveldomain	chrisb@contoso.l...	chrisb@contoso.l...	
CN=Amy Alberts,CN=Users,...	user	userPrincipalName	topleveldomain	amya@contoso.l...	amya@contoso.lo...	
CN=Kim Abercrombie,CN=...	user	userPrincipalName	topleveldomain	kima@contoso.lo...	kima@contoso.lo...	
CN=Hazem Abolrous,CN=U...	user	userPrincipalName	topleveldomain	hazema@contos...	hazema@contoso...	
CN=Luka Abrus,CN=Users,...	user	userPrincipalName	topleveldomain	lukaa@contoso.l...	lukaa@contoso.lo...	
CN=Humberto Acevedo,C...	user	userPrincipalName	topleveldomain	humbera@contos...	humbera@contos...	
CN=Alexandre Silva,CN=U...	user	userPrincipalName	topleveldomain	alexans@contos...	alexans@contoso...	

Query Count: 326 Error Count: 11

FIGURE 3-12 The IdFix tool

As you can see in Figure 3-12, the IdFix tool found 326 items in the directory and 11 errors. The vast majority of these errors are related to a nonroutable domain (*contoso.local*) being used in the *userPrincipalName* field. David Wright's UPN has a space, so we're accepting the edit the IdFix tool suggested. You also can see that two users have the same email address. We've told the tool to keep it for Mike but remove it for Kirk. Finally, Ray Mohman's email address begins with a period. We removed it and told IdFix that we edited the field.

MORE INFO **For more information about IdFix, see "Install and run the Office 365 IdFix tool" at *https://support.office.com/article/Install-and-run-the-Office-365-IdFix-tool-f4bd2439-3e41-4169-99f6-3fabdfa326ac*.**

Deploying AD Connect

After you resolve all the issues found in your Active Directory, decide on the sourceAnchor and UPN, and set up your custom domain setup (if needed), it's finally time to deploy AD Connect. Although you can install it on a domain controller, it would be best to install it on its own server. AD Connect is lightweight and doesn't require many resources. If you're looking to do a small deployment, the server can double-up as an AD FS server (although just make sure you size the machine appropriately). We highly recommend you deploy on Windows Server 2012 R2, especially if you want to use AD FS.

There are great resources online that will walk you through all the details. It's all wizard-driven, so it's actually quite simple. Again, see "Set up directory synchronization for Office 365," at *https://support.office.com/article/Set-up-directory-synchronization-for-Office-365-1b3b5318-6977-42ed-b5c7-96fa74b08846*, and "Custom installation of Azure AD Connect," at *https://azure.microsoft.com/en-us/documentation/articles/active-directory-aadconnect-get-started-custom*, for information on installing Azure AD Connect.

MORE INFO **To download Azure AD Connect, go to *https://www.microsoft.com/download/details.aspx?id=47594*.**

The first thing you need to decide is whether to do an express or custom install. You can choose Use Express Settings if you have a simple Active Directory deployment and are planning to use all the defaults. Use it if you have a single forest, you want to sync all users and all attributes, and you want to keep the default sourceAnchor (objectGUID) and UPN (*userPrincipalName*). It will set up everything for you.

MORE INFO **For more information, see "Getting started with Azure AD Connect using express settings" at *https://azure.microsoft.com/documentation/articles/active-directory-aadconnect-get-started-express*.**

However, if you're like us and you like to have a little more control, select Customize on the first page of the wizard. Even if you decide to keep all the defaults, it can be good to see what your options are. With a customized deployment you can do the following:

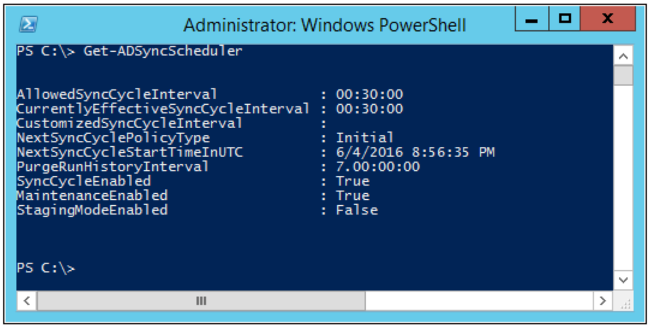
- Specify a custom installation location (such as on a data drive).
- Use an already-existing SQL Server instance (if you want high availability or greater control).
- Use an existing service account (which is not really recommended unless you're using a proxy that requires authentication or you're using an existing SQL Server instance).
- Define custom local sync groups (but it's unlikely you'll want to).
- Choose to deploy AD FS. If you do so, there will be another section of the wizard that will walk you through deploying AD FS, including specifying AD FS and AD FS Proxy servers. Deploying AD FS has never been easier.
- Add multiple Active Directory directories. (No domain trusts needed.)
- Specify the User Principal Name (either *userPrincipalName* or an alternate ID—but again, beware of doing so).
- Configure which Active Directory organizational units (OUs) to sync.
- Specify the sourceAnchor and, if users are in multiple domains, what identifies a user as unique.
- Specify a security group that contains the list of users to sync. This action is really meant only for pilot deployments, not for production.
- Specify which attributes to include in the sync. The wizard does so on a per-Office-365-application basis. If, for example, you know you're never going to use Skype for Business, you can avoid syncing the attributes it needs.
- If it's available with your licensing option, enable password or device writeback.
- If you chose AD FS as the sign-in method, specify whether or not to sync the password hashes (to act as a fallback in case AD FS isn't working or needed).
- If Exchange is deployed in the domain, specify whether or not to configure Exchange hybrid mailboxes or Office 365 Groups writeback.
- Choose additional attributes in the sync. A primary use case relates to user profiles in SharePoint Online. You can map Azure AD properties to user property fields (either existing or new ones) for use in such things as SharePoint customizations and SharePoint Audiencing. An example of this is an employee ID. Attributes can also be used for dynamic group membership.
- Place the server in or out of Staging Mode.

Note that you can run the wizard any time. You can use it to change settings, such as deploying AD FS, adding AD FS servers, switching the authentication method, adding attributes, and working with Staging Mode.

MORE INFO **For an excellent resource on installation using customized settings, see "Custom installation of Azure AD Connect" at *https://azure.microsoft.com/documentation/articles/active-directory-aadconnect-get-started-custom*.**

Running a sync

In previous tools, the syncs were scheduled using Windows Task Scheduler or a separate Windows service. Beginning with version 1.1 of the tool, AD Connect includes a scheduler that can be customized, removing the need for external sync methods. By default, the sync runs every 30 minutes, but you can change this to the frequency you need. You cannot, however, schedule the sync to run more often than what Azure allows. This limit can be determined by running the Get-ADSyncScheduler cmdlet and reviewing the *AllowedSyncCycleInterval* property. The results of the cmdlet are shown in Figure 3-13.



```
PS C:\> Get-ADSyncScheduler


AllowedSyncCycleInterval            : 00:30:00
CurrentlyEffectiveSyncCycleInterval : 00:30:00
CustomizedSyncCycleInterval         :
NextSyncCyclePolicyType             : Initial
NextSyncCycleStartTimeInUTC         : 6/4/2016 8:56:35 PM
PurgeRunHistoryInterval             : 7.00:00:00
SyncCycleEnabled                    : True
MaintenanceEnabled                  : True
StagingModeEnabled                  : False



PS C:\>
```

FIGURE 3-13 Results of Get-ADSyncScheduler

If for some reason you can't wait for the next sync (the time of which is displayed in the *NextSyncCycleStartInUTC* property) and you need to start a sync, you can do so by running the following PowerShell command:

```
Start-ADSyncSyncCycle -PolicyType Delta
```

This starts a delta sync, but if you need to do a full sync, change the *PolicyType* to *Initial*. You can also stop a sync by running the Stop-ADSyncSyncCycle cmdlet. It finishes its current connection and then stops.

MORE INFO **Learn more about the scheduler by visiting "Azure AD Connect sync: Scheduler" at *https://azure.microsoft.com/documentation/articles/active-directory-aadconnectsync-feature-scheduler.***

Configuring identity federation with Office 365

At this point, we've implemented identity sync using AD Connect, pushing our on-premises user information into Azure AD and Office 365. This might be sufficient for most companies, especially because it's free (or nearly so). As discussed previously, we have another identity option available: federated identities.

Again, federated identities give users a secure, single-sign-on experience by using a trusted external identity system that authenticates the user and issues a token that systems use to sign in the user. This is most commonly done with AD FS, which is deployed on-premises. In this section, we'll talk about how to configure identity federation with Office 365 using AD FS.

A brief intro to Active Directory Federation Services

We already talked a bit about AD FS, and we walked through an example scenario of a user needing a document in Office 365. With AD FS, you have one or more AD FS servers. These are the heart of the system, and they're what users log in to and what hands off the secure token. AD FS can be used to provide single sign-on to services (like Office 365) using SAML tokens or Windows Integrated authentication using Kerberos-constrained delegation. To do the latter, the AD FS server must be in the same domain as the service or in a domain with a trust.

When a user requests access to a system configured for AD FS, that system redirects the user to a page on the AD FS server. The user then enters his credentials and AD FS attempts to log the user into the domain. If the login is successful, AD FS creates a token signed by a trusted certificate, adds the token to the user's session, and redirects the user back to the system. At this point, the system sees that the session now has the token and, because it also trusts the same certificate that signed the token, grants access to the user. As long as the user has the token and the token hasn't expired, the user can then use that same token and sign on to other services that trust that same AD FS system. The user needs to sign in only once instead of once per system (hence, *single sign-on*). If the user is coming from a domain-joined machine and other settings are correct, that initial sign-on is done for the user automatically. He can go straight to Office 365, for example, without having to enter his credentials.

When using AD FS on-premises, you can generally stop with just the AD FS server. Actually, it's highly recommended that you deploy at least two (behind a load balancer) for redundancy. If, though, you want to use AD FS to authenticate users to systems from the internet, you need to go a step further. It's important to protect your AD FS servers as you would a domain controller, so you don't want to expose them directly to the internet. Instead, you deploy one or more AD FS Proxy servers in your perimeter network (also known as DMZ, for demilitarized zone). These servers protect the AD FS servers from direct access and channel appropriate requests to them. As with the AD FS servers, you'll probably want at least two for redundancy (also behind a load balancer).

This latter configuration with AD FS Proxy servers is what's recommended for use with Office 365. With it, you configure an entry in your public DNS that points users to the AD FS Proxy (or load-balanced endpoint).

In the example topology shown next, we use *fs.contoso.com* as the DNS name, and it points to the IP address on the load balancer, which is exposed in the perimeter firewall on port 443. The internal DNS also has an entry for *fs.contoso.com*, but because we want users to go directly to AD FS and not have to go out to the internet and back in through the proxies, we point the record directly to the load balancer in front of the AD FS servers. Both internal and public DNS records have the same name, *fs.contoso.com*, but they each point to a different IP address. This configuration is what's called *split-brained DNS*, and it's primarily used to optimize performance for internal users.

Figure 3-14 is an example of a typical, redundant AD FS topology we recommend.

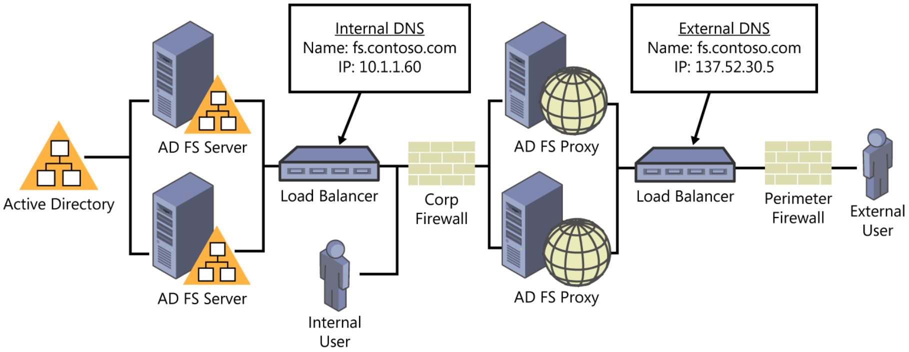


FIGURE 3-14 A typical AD FS topology

Deploying AD FS for Office 365

Unfortunately, once again, we do not have the space to go into detail about how to deploy AD FS for Office 365. If you are able to deploy AD FS on Windows Server 2012 R2 servers, you can use AD Connect to configure the servers. It will, however, still be up to you to provide any load-balancing and firewall configurations you might need. You'll also need an SSL certificate from a third-party provider (like VeriSign or DigiCert) that includes the DNS name of the AD FS Proxy endpoint (such as *fs.contoso.com*). The domain should be at the 2008 functional level or higher.

If you have access to Windows Server 2012 R2 servers for the deployment, AD Connect can deploy most of this infrastructure for you in a simple wizard. You need to have the PFX certificate ready for the wizard or already installed on the servers. Then you simply give it the name of the AD FS servers and proxies and tell it which custom domain to configure for federation. The wizard does the rest.

MORE INFO **For more details on how to do this, see the "Configuring federation with AD FS" section of "Custom installation of Azure AD Connect" at *https://azure.microsoft.com/documentation/articles/active-directory-aadconnect-get-started-custom/#configuring-federation-with-ad-fs*.**

MORE INFO **For more information about AD FS on Windows Server 2012 R2, see the "AD FS Design Guide in Windows Server 2012 R2" at *https://technet.microsoft.com/library/dn554245%28v=ws.11%29.aspx*.**

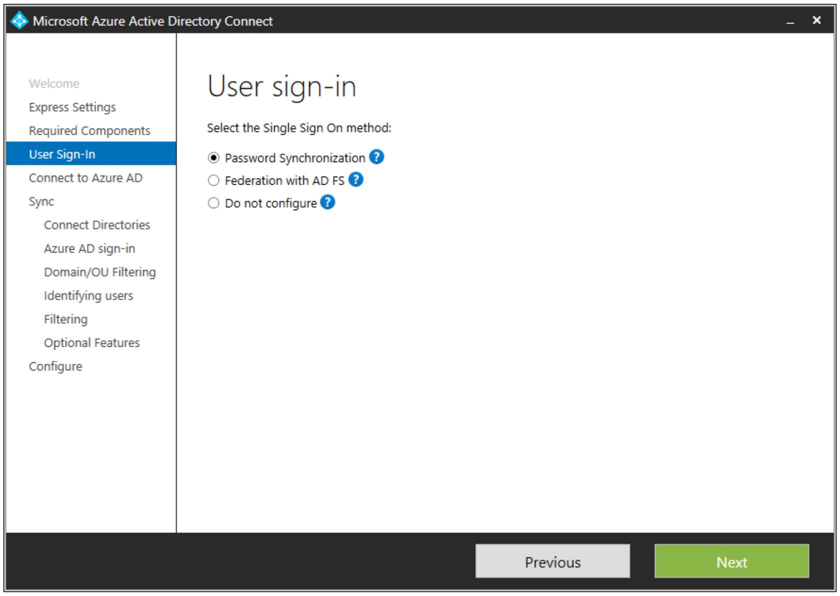
If you already have an existing AD FS farm, don't worry—you can use it instead of deploying a new one. If you don't already have an extranet scenario with AD FS Proxies accessible from the internet, plan on adding this capability as part of the rollout. If the existing AD FS farm is running Windows Server 2012 R2, you can integrate it with AD Connect using the wizard. (Choose "Use an existing Windows Server 2012 R2 AD FS farm.") If, however, it's an AD FS 2.0 farm (Windows 2008 or 2008 R2), you won't be able to use AD Connect to configure federation. Instead, you must manually configure the federation from the AD FS server using PowerShell cmdlets.

MORE INFO **For more information about AD FS 2.0 on Windows 2008 or 2008 R2, see the "AD FS 2.0 Design Guide" at *https://technet.microsoft.com/library/dd807036(ws.10).aspx*.**

Configuring Office 365 for federation

After the AD FS farm and its proxies have been fully configured, the last step is to configure Office 365 for federation. Nonfederated domains are called *managed* or *standard*. You can configure individual domains as either federated or managed. Both can exist in the same tenant side by side, and you can even have multiple domains each federated with different AD FS farms. The important thing, though, is that you verify the custom domain before you attempt to federate it. You will not be able to configure federation on a nonverified domain.

If you deployed AD FS using Windows Server 2012 R2, you can simply use AD Connect and select Federation With AD FS as the sign-in method. (See Figure 3-15.) The wizard will walk you through the rest and configure Azure AD for you.



User sign-in

Select the Single Sign On method:

● Password Synchronization

○ Federation with AD FS

○ Do not configure

FIGURE 3-15 The User Sign-In page of the AD Connect wizard

If you're using an AD FS 2.0 farm, you need to convert the domain manually via PowerShell. You can do this from an elevated PowerShell console either on the primary AD FS server or from another machine. Before doing so, make sure the Azure AD PowerShell module and the Microsoft Online Services Sign-In Assistant have been installed. If you're running the commands somewhere other than on the primary AD FS server, you first need to connect to the AD FS server by using the Set-MsolADFSContext cmdlet, giving it the name of the primary AD FS server. You will be prompted for administrator credentials. This command isn't needed if you're on the AD FS server:

```
Set-MsolADFSContext -Computer Win2008ADFS01.contoso.com
```

In the same PowerShell session, use the Convert-MsolDomainToFederated cmdlet to convert the domain. The *DomainName* parameter specifies the domain to be converted. If your Azure AD tenant needs to support federation with more than one domain, it's important to use the *SupportMultipleDomain* switch each time. If, for example, you need to federate two domains, use the *SupportMultipleDomain* switch with each of the Convert-MsolDomainToFederated commands:

```
Convert-MsolDomainToFederated -DomainName "contoso.com" -SupportMultipleDomain
Convert-MsolDomainToFederated -DomainName "adatum.com" -SupportMultipleDomain
```

Notice that the cmdlet doesn't have a parameter specifying the AD FS farm or its sign-in URL. This is because either we are on the AD FS server when we run it or we used the Set-MsolADFSContext cmdlet to connect to the AD FS server. The cmdlet reads the AD FS configuration and configures the relying party trusts for us. Pretty nice!

Once this is done, the domain should now be configured for federation with our AD FS farm. Test it out by attempting to sign in to Office 365 using a federated account. Either you should be able to sign directly into Office 365 using your currently signed-in credential or you should be redirected to your AD FS sign-in page. If after signing in you are successfully signed in to Office 365, you have successfully configured federation. Congratulations!

We recommend you consider doing one last optional thing: include the password hash in the sync. With AD FS, you do not need to sync the password hash. It's more secure. However, if you include the password hash, you have a contingency plan in case you need to temporarily disable (or fully remove) federation. Otherwise, you'll have to generate new passwords for all your users and distribute them. That certainly would not be a pleasant day. After your federation is complete, simply go back into the AD Connect Wizard, choose the Customize Synchronization Options task, and ensure that Password Synchronization is selected on the Optional Features page. With luck, you'll never need to use it, but it's good to have it as a backup if needed.

REAL WORLD **Most organizations would like to pilot a technology before they decide to adopt it. If you're just getting started with federation, implementing this strategy isn't a challenge. However, it's not simple to pilot federation if the domain is already being synced with Office 365. This is because the Convert-MsolDomainToFederated cmdlet converts all users in the domain to federated. You cannot convert just a subset of users.**

To pilot federation, you need to register a new, separate domain in Office 365 and go through the whole sync and AD FS process. Yes, this means provisioning a new certificate. From a high level, the process looks like this:

1. **Register and verify the new domain.**
2. **Obtain a certificate for the new domain.**
3. **Remove the users' Office 365 licenses before making the switch to ensure the UPN change is synced (or use *Set-MsolDirSyncFeature -Feature SynchronizeUpnForManage-dUsers-Enable $True* to enable UPN changes while licensed).**
4. **In the local Active Directory, add the new domain as a UPN suffix and switch the pilot users to that suffix.**
5. **Configure AD FS for federation using that domain.**
6. **After the users have been synced with the pilot domain and AD FS has been configured, use Convert-MsolDomainToFederated to convert the pilot domain.**

After the pilot is complete, you need to undo all that work to bring the users back to their original UPNs. As you can see, this process involves a lot of work. If you haven't yet implemented directory sync and you're looking to deploy AD FS, we recommend that you do your federation proof of concept at the beginning, before all your users have been onboarded to Office 365.

Switching back to managed identities

After you convert to federated identities, you might find it necessary to switch the domain back to managed or standard. You might do this permanently if you no longer want to use federated identities, or you might need to do this temporarily if your AD FS infrastructure is offline.

The process to do it is simple, but the method you use to do it depends on whether or not the AD FS server is available. If you're doing it because of an AD FS failure, chances are that it won't be available. If you're going to make the switch, keep in mind that it can take up to two hours for it to take effect. Chances are, though, it won't take that long, but plan for it in your decisions and timing for the conversion. The more users you have, the longer it will take.

If your AD Connect is online and it was integrated with your Windows Server 2012 R2 AD FS farm, your easiest choice is to use the wizard to change the sign-in from Federation With AD FS to Password Sync. AD Connect should take care of it for you.

If your AD FS server is available, you can use the Convert-MsolDomainToStandard cmdlet. This is the preferred method. As mentioned earlier, either run it from the AD FS server or use Set-MsolADFSContext first to connect to the AD FS server. You need to supply the *PasswordFile* parameter and give it the path to a file that will be created (for example, C:\Temp\Passwords.txt).

If password sync is operational, the passwords will be overwritten and you won't need to care about this file. (It's still required, though.) However, if you aren't syncing your passwords, you need to distribute the passwords in this file to your users. You also need to use the correct *SkipUserConversion* parameter. If you're permanently moving away from federated identities, this value should be *$false*. A value of *$false* converts all the users to managed identities. If, however, this is a temporary conversion, make sure the value is *$true*.

The following command permanently converts the *contoso.com* domain from federated to managed:

```
Convert-MsolDomainToStandard -DomainName contoso.com -SkipUserConversion $false
-PasswordFile C:\Temp\Passwords.txt
```

If your AD FS server is not available, you can't use the Convert-MsolDomainToStandard cmdlet. Instead, you need to use Set-MsolDomainAuthentication. For example:

```
Set-MsolDomainAuthentication -DomainName contoso.com –Authentication Managed
```

When AD FS is available again, either switch the sign-in method in AD Connect or run Convert-MsolDomainToFederated:

```
Convert-MsolDomainToFederated –DomainName contoso.com
```

Index