**Agenda: Azure Active Directory**

- Azure AD Introduction
- Azure AD Editions
- Managing Active Directories
- Adding a custom domain name to Azure AD
- Managing Azure AD Users, Groups and Devices
- Adding Partner Users from other organization
- Configure Windows 10 with Azure AD domain join
- Configuring Role Based Access Control
- Synchronizing On-Premise AD Identities with Azure AD
- Azure AD Connect
- Azure AD User Sign-In Options
    - Password Synchronization
    - Passthrough Authentication
    - Federated SSO
- Using SSO with Azure AD
- Mul
- Integrating SaaS Applications with Azure AD for SSO
    - Add Users and Groups to Application
    - Revoke access to SaaS Applications

## Azure Active Directory Introduction

- Microsoft Azure Active Directory (Azure AD) is a multi-tenant cloud-based **identity and access management solution** for the resources that exist in the cloud.
- **For IT Admins**, Azure AD provides an affordable, easy to use solution to give employees and business partners **Single Sign-On (SSO)** access to thousands of cloud SaaS Applications like Office365, Salesforce.com, Dropbox, and Concur.
- **For application developers**, Azure AD lets you focus on building your application by making it fast and simple to integrate with a world class identity management solution used by millions of organizations around the world.
- Organizations can use Azure AD to improve employee productivity, streamline IT processes, and improve security for adopting various cloud services. Employees can access online applications by using a single user account.
- Azure AD is highly scalable and highly available by design. Therefore, organizations do not have to maintain related infrastructure or worry about disaster recovery. Running out of 28 data centers around the world with

**automated failover**, you'll have the comfort of knowing that Azure AD is highly reliable and that even if a data center goes down, copies of your directory data are live in at **least two** more regionally dispersed data centers and available for instant access.

- Many applications built on different platforms such as .Net, Java, Node.js, and PHP can use industry standard protocols such as Security Assertion Markup Language (SAML) 2.0, WS-Federation, and OpenID Connect to integrate the identity management provided by Azure AD into the application logic. Through the support of OAuth 2.0, developers can develop mobile and web service applications that integrate with Microsoft's identity platform for cloud authentication and access management.
- Azure AD provides access to its content via **REST-based Graph API**, rather than by Lightweight Directory Access Protocol (LDAP), on which Active Directory relies.

**You can use Azure AD to:**

- Provide an identity management solution.
- Manage users and groups.
- Role based Access Control.
- Enable federation between organizations.
- Identify irregular sign-in activity.
- Configure SSO to cloud-based SaaS applications like Office365, Salesforce.com, DropBox etc…
- Configure access to the on-premise applications.
- Configure multi-factor authentication.
- Extend existing on-premises Active Directory implementations to Azure AD.

### Azure AD has many benefits.

- **Single sign-on to any cloud or on-premises web app.** Azure Active Directory provides secure single sign-on to cloud and on-premises applications including Microsoft Office 365 and thousands of SaaS applications such as Salesforce, Workday, DocuSign, ServiceNow, and Box.
- **Works with iOS, Mac OS X, Android, and Windows devices.** Users can launch applications from a personalized web-based access panel, mobile app, Office 365, or custom company portals using their existing work credentials—and have the same experience whether they're working on iOS, Mac OS X, Android, and Windows devices.
- **Protect on-premises web applications with secure remote access.** Access your on-premises web applications from everywhere and protect with multi-factor authentication, conditional access policies, and group-based access management. Users can access SaaS and on-premises web apps from the same portal.
- **Easily extend Active Directory to the cloud.** Connect Active Directory and other on-premises directories to Azure Active Directory in just a few clicks and maintain a consistent set of users, groups, passwords, and devices across both environments.

- **Protect sensitive data and applications.** Enhance application access security with unique identity protection capabilities that provide a consolidated view into suspicious sign-in activities and potential vulnerabilities. Take advantage of advanced security reports, notifications, remediation recommendations and risk-based policies to protect your business from current and future threats.
- **Reduce costs and enhance security with self-service capabilities.** Delegate important tasks such as resetting passwords and the creation and management of groups to your employees. Providing self-service application access and password management through verification steps can reduce helpdesk calls and enhance security.

**Azure AD editions:**

- **Free edition** provides
  - User and group management,
  - Self-service **password change** for cloud users.
  - **Synchronize** with on-premises directories,
  - Get **single sign-on** across Azure, Office 365, and thousands of popular SaaS applications like Salesforce, Workday, Concur, DocuSign, Google Apps, Box, ServiceNow, Dropbox, and more.
  - End-users are entitled to get single sign-on access for **up to 10 applications**.
- **Basic edition** extends the free edition's capabilities. Additionally, this edition has a Microsoft high availability service level agreement (SLA) uptime of 99.9%. It supports cost reducing features like
  - Group-based access management,
  - Self-service password reset for cloud users.
  - Company Branding (Logon Pages / Access Panel customization)
  - Azure Active Directory Application Proxy (to publish on-premises web applications using Azure Active Directory)
- **Premium P1 edition** is designed for task workers with cloud-first needs.It supports
  - Multi-Factor Authentication
  - Self-service identity and access management (IAM),
  - Advanced reports for security and usage information.
  - Dynamic groups and self-service group management.
  - Microsoft Identity Manager (an on-premises identity and access management suite)
  - Self-service password reset with password writeback for on-premises users.
- **Premium P2 edition** is designed to accommodate organizations with more demanding identity and access management needs.It supports
  - All features of Azure AD Premium P1
  - Azure Active Directory **Identity Protection** leverages billions of signals to provide risk-based conditional access to your applications and critical company data.

- We also help you manage and protect privileged accounts with Azure Active Directory **Privileged Identity Management** so you can discover, restrict and monitor administrators and their access to resources and provide just-in-time access when needed.

**Activate Azure AD Premium P2 trial**

1. In the Azure portal, while signed in by using the Microsoft account that has the **Owner role** in the Azure subscription and is a **Global Administrator** of the Azure AD tenant associated with that subscription, navigate to the Azure AD tenant blade.
2. From the Azure AD tenant blade, navigate to the **Licenses** blade.
3. From the **Licenses** blade, navigate to the **Licenses - All products** blade.
4. From the **Licenses - All products** blade, navigate to the **Activate** blade and activate the **Azure AD Premium P2** trial.

**Assign Azure AD Premium P2 licenses**

1. Navigate to the **Users - All users** blade of the Azure AD tenant associated with your Azure subscription.
2. From the **Users - All users** blade, display the **admin@sandeepsonideccansoft.onmicrosoft.com - Profile** blade.
3. **Edit Settings → Usage location** matching the location of the Azure AD tenant.
4. Navigate back to the **Licenses - Overview** blade of the Azure AD tenant associated with your Azure subscription.
5. From the Azure Active Directory → **Licenses - Overview** blade, navigate to the **Products** blade.
6. From the **Products** blade, navigate to the **Azure Active Directory Premium P2 - Licensed users** blade.
7. From the **Azure Active Directory Premium P2 - Licensed users**, navigate to the **Assign license** blade.
8. From the **Assign license** blade, assign an Azure AD Premium P2 license to the **admin@sandeepsonideccansoft.onmicrosoft.com** user account.

**Initial domain name**

By default, when you create an Azure subscription an Azure AD domain is created for you. This instance of the domain has initial domain name in the form domainname.onmicrosoft.com. The initial domain name, while fully functional, is intended primarily to be used as a bootstrapping mechanism until a custom domain name is verified.

**Tenants**

A tenant is simply a dedicated instance of Azure AD that your organization receives and owns when it signs up for a Microsoft cloud service such as Azure or Office 365. For example, contosogold.onmicrosoft.com, is a tenant.

A tenant houses the users in a company and the information about them - their passwords, user profile data, permissions, and so on. It also contains groups, applications, and other information pertaining to an organization and its security.

You can have multiple tenants within your organization. Each tenant can have a different purpose and fulfill a different scenario. For example, you might have tenant for Testing, Office365, and Production.
Can you think of reasons why you might want different tenants?

- **Isolation**. Each tenant is isolated with different policies, users, groups, and roles.
- **Resources**. Each tenant can have different resources specific for their functionality.
- **Administration**. Each tenant can have different administrator roles.
- **Synchronization**. Each tenant can implement synchronization in a different way.

**Multiple directory support means that an administrator can:**

- Add a new directory for testing or other non-production usage, or for managing data synced from another AD forest.
- Manage all existing Azure AD directories, such as Azure, Office 365, Microsoft Intune, by using the same account—as long as the same account is a Global Administrator for all the directories.

**Adding a New Directory:**

**1.** Azure Portal → +New → Active Directory → **Create**
2. Add Directory Dialog,
    - Name = Dummy Organization
    - Domain Name = **DummyOrg**.onmicrosoft.com
    - Country = INDIA

**To change directory of Azure Subscription:** (So that the users of that directory can be given access to manage resources of the subscription)
Azure Portal → Subscription menu → select your subscription → Change Directory, and then select any existing directory for your subscription.

**To change Directory Role of existing User:**

1. Azure portal → Azure Active Directory → Users and groups → All users → select User → Directory role
2. Directory role = User / Global administrator / Limited administrator (check the administrator roles that you want to assign to this user).
3. Save

**Note: Following link contains information about each role and what they can do and cannot do.**
**https://azure.microsoft.com/en-us/documentation/articles/active-directory-assign-admin-roles/**

**Deleting an Azure AD directory:**

By using a user account with global administrative rights, you can delete an Azure AD directory if the following conditions are met:

1. You deleted all the users in the directory except the Global Administrator for the directory that you want to delete. The Global Administrator's name cannot have the same suffix as the directory you intend to delete.
2. All applications configured for SSO are removed from the directory.
3. The directory is not associated with any of the cloud services such as Azure, Office 365, or Azure AD Premium.
4. No multi-factor authentication providers are linked to the directory.

**Steps:** Azure Portal → Switch to the Directory from the Menu on top right → Azure Active Directory → Delete directory (in menu of overview blade)

## Add a custom domain name to Azure AD

Although the initial domain name for a directory can't be changed or deleted, you can add any routable custom domain name you control. This simplifies the user sign-on experience by allowing user to logon with credentials they are familiar with.

**Practical information about domain names**

- Only a global administrator can perform domain management tasks in Azure AD.
- Domain names in Azure AD are globally unique. If one Azure AD directory has verified a domain name, then no other Azure AD directory can verify or use that same domain name.
- Before a custom domain name can be used by Azure AD, the custom domain name must be added to your directory and verified

**Adding and Verifying Custom Domain Names**

1. Azure Portal → **Azure Active Directory** → Domain names → + **Add domain name**.
2. Enter the name of your custom domain, such as **'deccansoft.net**'.
   Be sure to include the .com, .net, or other top-level extension, and leave the checkbox for "single sign-on" (federation) cleared → Add
3. In the Microsoft cloud service portal, note the DNS records that will need to be created at your domain registrar or DNS hosting provider.
4. Sign-in to your domain registrar or DNS hosting provider, and create the DNS records.

Note: A domain name can be verified in only a single directory. If a domain name was previously verified in another directory, it must be deleted there before it can be verified in your new directory.

You can add up to **900 custom domain** names to each Azure AD directory.

Azure AD provides the required DNS information, either TXT (preferably), or MX records if your DNS provider does not support TXT records.

The following is an example of a TXT record used for custom domain verification:

Alias or Host name: **@**

Destination or Points to Address: **MS=ms96744744**

TTL: **1 hour**

After verification, the administrator can make the domain the primary domain for the Azure tenant. For example, you can replace adatum12345.onmicrosoft.com with adatum.com, so that new users will be automatically created in this directory.

**To add company branding to your directory**

1. Azure Portal → **Azure Active Directory** → Users and groups
2. On the **Users and groups - Company branding** blade, select the **Edit** command.
3. Modify the elements you want to customize. All elements are optional.
4. Click **Save**.

Note: This feature is available in Azure AD Premium only.

Likewise Password reset and Sign-ins is also available to AD Premium only.

**Delete a custom domain name**

To delete a custom domain name, you must first ensure that no resources in your directory rely on the domain name.

You can't delete a domain name from your directory 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.

**Step:** Azure Portal → **Azure Active Directory** → Custom Domain names → Select Domain name → Delete

## Managing Azure AD Users, Groups and Devices

A directory can consist of the following three types of identities:

- Users added manually to the directory (cloud only identities)
- Third-party accounts (third-party identities)
- Users synced from existing Active Directory installations (on premise identities)

**There are essentially two ways to create and manage your users:**

- As cloud identities by using only Azure AD. This is the quickest and most straightforward method.
- As directory-synchronized identities by using an on-premises directory service to synchronize with Azure AD. This method has the added complexity of installing and configuring synchronization software to ensure that directory objects synchronize successfully with Azure AD.

**Types of User**

1. New user in your organization.
2. User with existing Microsoft account (any email id registered with https://signup.live.com)

**Creating new user in your organization:**

1. Azure Portal → Azure Active Directory → Users and groups
2. All users → + Add
3. Create the following user in the directory:
    - **Name**: the display name
    - **User name**: unique name within the domain name associated with the current Azure AD tenant that the user will provide when signing in
    - **Profile**: first name, last name, job title, and department
    - **Properties**: Source of Authority (Azure Active Directory)
    - **Groups**: groups that the user should be a member of
    - **Directory role**: User
4. Create the user and record the temporary password.
5. At the top-right corner of the page, click your Azure subscription name, and then click **Sign Out**. **You have been signed out** page,
6. Click **SIGN IN** and Login again as JSmith.

**Note that by default this user will not have access to any resources.**

**Add Google as an identity provider for B2B guest users**

https://docs.microsoft.com/en-us/azure/active-directory/b2b/google-federation

**To add Users in Bulk:**

Install-Module AzureAD

https://docs.microsoft.com/en-us/powershell/azure/active-directory/importing-data?view=azureadps-2.0

**Manage groups by using the Azure portal**

1. GROUPS → ADD A GROUP.

2. In the Add Group dialog box, enter the following settings, and then click Complete:
   - NAME: Sales
   - DESCRIPTION: Sales team
3. Click **Sales** → Click **ADD MEMBERS**.
4. In the **Add members** dialog box, click required Users, and click **Complete**.

**Managing devices in the Azure portal**

Users can join Windows 10 devices to Azure AD by themselves during the first-run experience or from the system settings. If users sign in to Windows 10 by using their Azure AD credentials, they can experience SSO to Office 365 and any other applications that use Azure AD for authentication, including the Azure AD Access Panel (at myapps.microsoft.com).

In Azure AD, you need to **enable the option** for users to join their devices to Azure AD

1. Azure Active Directory → Devices → Device Settings → . . . → Save

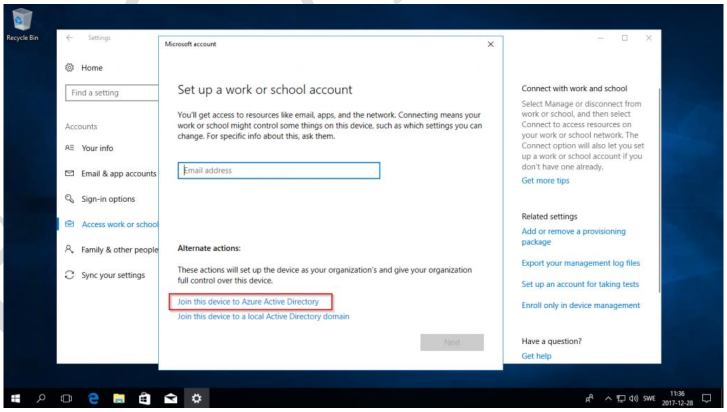
   Users may join devices to Azure AD = All

   Users may register their devices with Azure AD = All

**To join a device (Windows 10) to Azure AD:**

https://tech.xenit.se/join-windows-10-computer-azure-active-directory/

1. **Windows10 OS**: Start → Settings → Account → Access Work or School → +Connect



Microsoft account

Set up a work or school account

You'll get access to resources like email, apps, and the network. Connecting means your work or school might control some things on this device, such as which settings you can change. For specific info about this, ask them.

Email address

Alternate actions:

These actions will set up the device as your organization's and give your organization full control over this device.

Join this device to Azure Active Directory

Join this device to a local Active Directory domain

Next

Connect with work and school

Select Manage or disconnect from work or school, and then select Connect to access resources on your work or school network. The Connect option will also let you set up a work or school account if you don't have one already.

Get more tips

Related settings

Add or remove a provisioning package

Export your management log files

Set up an account for taking tests

Enroll only in device management

Have a question?

Get help

2. At this point, you would be able to sign in to the local computer by using Azure AD credentials.

**After a device is registered in Azure AD, you can control its usage**. For example, if you determine that the device has been lost or compromised, you can delete or disable its Azure AD object from the portal. If Microsoft Intune or another mobile device management (MDM) system manages the device, you can implement additional capabilities such as policy-based configuration and software deployment.

## Configuring Self-Service Password Reset

1. Azure AD → Password Reset → Properties
2. Self-service password reset enabled: None, Selected, and All.
3. Select the Group if Applicable
4. Save.

After enabling password reset for user and groups, you pick the number of authentication methods required to reset a password and the number of authentication methods available to users.

5. Azure AD → Password Reset → Authentication methods
6. Number of methods required to reset: 1
7. Methods available to users:
   - **Mobile phone**
   - **Office phone**
8. Save

From the **Registration** page, make the following choices:

9. Require users to register when they sign in: **Yes**
10. Set the number of days before users are asked to reconfirm their authentication information: **365**

**Test self-service password reset**

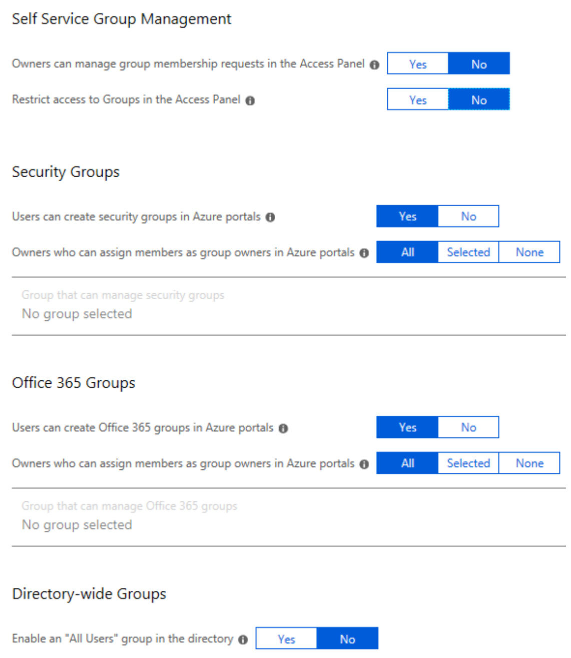
11. Open a new browser window in InPrivate or incognito mode, and browse to https://aka.ms/ssprsetup.
12. Sign in with a non-administrator test user, and register your authentication phone.
13. Once complete, click the button marked **looks good** and close the browser window.
14. Open a new browser window in InPrivate or incognito mode, and browse to https://aka.ms/sspr.
15. Enter your non-administrator test users' User ID, the characters from the CAPTCHA, and then click **Next**.
16. Follow the verification steps to reset your password

**Self Service Group Creation:**

It's is another feature in Azure Active Directory **Premium** that allows users to create and manage their own security groups or Office 365 groups in Azure Active Directory (Azure AD).

## Configure Self Service Group Management

1. Azure AD → Groups → General →



Self Service Group Management

Owners can manage group membership requests in the Access Panel ⓘ Yes [✓ No]

Restrict access to Groups in the Access Panel ⓘ Yes [✓ No]

Security Groups

Users can create security groups in Azure portals ⓘ [✓ Yes] No

Owners who can assign members as group owners in Azure portals ⓘ [✓ All] Selected None

Group that can manage security groups
No group selected

Office 365 Groups

Users can create Office 365 groups in Azure portals ⓘ [✓ Yes] No

Owners who can assign members as group owners in Azure portals ⓘ [✓ All] Selected None

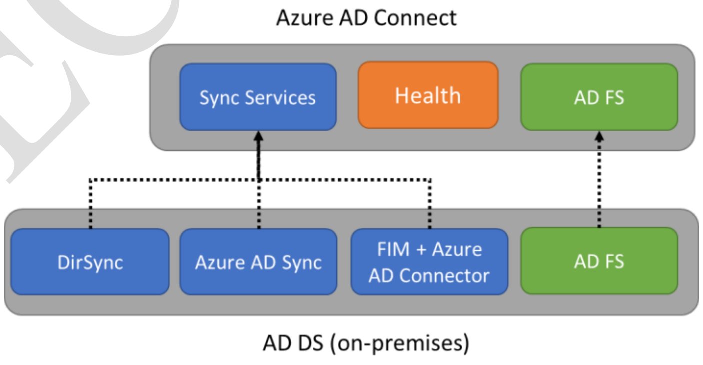
Group that can manage Office 365 groups
No group selected

Directory-wide Groups

Enable an "All Users" group in the directory ⓘ Yes [✓ No]

## Integrating On-Premises AD identities with Azure AD

- **Azure AD Connect** will integrate your on-premises directories with Azure Active Directory. This allows you to provide a common identity for your users for Office 365, Azure, and SaaS applications integrated with Azure AD.
- 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.
- The beauty of this approach is that any time your organization adds or deletes a user, or a user changes a password, you use the same process that you use today in your on-premises environment. All of your on-premises AD changes are automatically propagated to the cloud environment.



**Azure Active Directory Connect** is made up of three primary components:

1. **Sync Service** - This component is responsible for creating users, groups, and other objects. It is also responsible for making sure identity information for your on-premises users and groups is matching the cloud.

2. **Health Monitoring** - Azure AD Connect Health can provide robust monitoring and provide a central location in the Azure portal to view this activity. For additional information, see Azure Active Directory Connect Health.
3. **AD FS - 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. This can be used by organizations to address complex deployments, such as domain join SSO, enforcement of AD sign-in policy, and smart card or 3rd party MFA.

**Express Settings:**

- If you have a single forest AD then this is the recommended option to use.
- User sign in with the same password using password synchronization.
- It's the default option and mostly used for common deployed scenario.

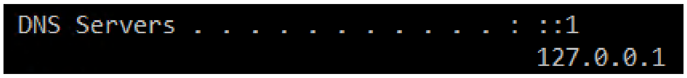
**Customized Settings**

- Used when you have multiple forests. Supports many on-premises topologies.
- Customize your sign-in option, such as ADFS for federation or use a 3rd party identity provider.
- Customize synchronization features, such as filtering and writeback.

## Creating a Domain Controller and Join Azure virtual machines to a domain

1. Create a new VM (DemoVM1) to be used as Domain Controller and DNS Server.
2. Change the Private IP to static: VM → Networking → click on Network Interface → IP configurations → ipconfig1 → **Private IP Address settings**, Assignment = **Static** → Save (note the IP address)
3. Virtual Network → Select the VNet → **DNS servers** → Select Custom and provide **static IP** of VM from previous step.
4. Restart your VM
5. Promote the VM as Active Directory Domain Controller and DNS Server.
   **a)** RDP to VM
   **b)** Server Manager → Dashboard → Add Roles andFeatures → Next → Next
   **c)** Check **Active Directory Domain Service** and **DNS** → Next → . . . → Finish
   **d)** From Notification in Server Manager Window (Top Right) → Click on ***Promote this server to a Domain Controller***
   **e)** Add a New Forest → Root domain name: ***bestazuretraining.com*** *(Custom Domain name created earlier)* → Next, Provide DSRM Password → Next → . . . → Finish
   **f)** Restart your machine.
   **g)** Server Manager → Tools → **DNS** → Right Click on Server → Properties → Forwarders → Edit → **Delete existing IP and replace with 8.8.8.8** → OK
   **h)** Restart your machine
   **i)** In VM type the following command to verify that this Machine is DNS Server

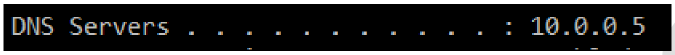
C:\> **Ipconfig** /all



```
DNS Servers . . . . . . . . . . . : ::1
                                    127.0.0.1
```

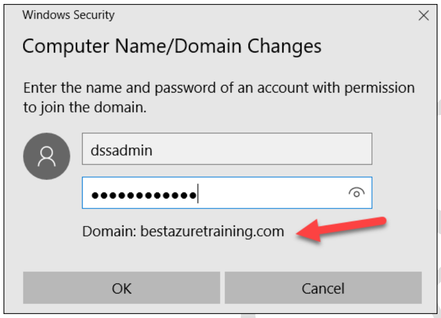
6. Create New Users (User1 and User2) and Groups in this New Domain Controller
    1. Administrative Tools → Active Directory Users and Computers
    2. Expand your Domain Name (bestazuretraining.com) → Expand Users → Right click add New User
7. Create a new VM (DemoVM2) and Join to the Domain Controller.
    1. Restart the new VM if the DNS server is changed after this VM was created.
    2. RDP to VM
    3. In VM type the following command to verify that DemoVM1 (Domain Controller) is DNS Server

       C:\> **Ipconfig** /all



```
DNS Servers . . . . . . . . . . . : 10.0.0.5
```

4. Server Manager → local server → Workgroup → Change → Domain Name = ***bestazuretraining.com***, Provide admin u/p → Your machine has now joined the domain → Restart the Machine.



Windows Security

Computer Name/Domain Changes

Enter the name and password of an account with permission to join the domain.

dssadmin

••••••••••••

Domain: bestazuretraining.com

5. To Remote Desktop Login with new Identity on this VM (DemoVM2)

   Computer Management > Local Users and Groups > Groups > Remote Desktop Users

   Users needs to be added to the above on the computer in which she's trying to remote into. The setting you changed is just to allow people to RDP into the machine, but they still need individual rights to do it.

## Azure AD Connect Express Installation Walkthrough

1. Azure Portal → Azure Active Directory → Users and Groups → All users → + New User → Username = admin@sandeepsoni.onmicrosoft.com, Directory Role = Global Admin.
2. Remote Login to VM (Primary Domain Controller)
3. Server Manager → Local Server → IE Enchanced Security Configuration = Off
4. Add few Users to its **Active Directory Users and Groups**.
5. Download Azure AD Connect. Navigate to and double-click on **AzureADConnect.msi**.
6. On the Welcome screen, select the box agreeing to the licensing terms and click **Continue**.
7. On the Express settings screen, click **Use express settings.**

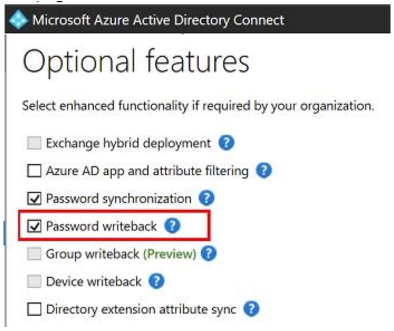
8. On the Connect to Azure AD screen, enter the username and password of a **global administrator** (admin@sandeepsonideccansoft.onmicrosoft.com) for your Azure AD. Click **Next**.
9. On the Connect to AD DS screen, enter the username and password for an **enterprise admin account (**bestazuretraining.com\dssadmin**)**. Click **Next**.
10. The **Azure AD sign-in configuration** page will only show if you did not complete verify your domains.
11. On the Ready to configure screen, click **Install**.
    1. Optionally on the Ready to configure page, you can unselect the **Start the synchronization process as soon as configuration completes** checkbox. You should unselect this checkbox if you want to do additional configuration, such as filtering. If you unselect this option, the wizard configures sync but leaves the scheduler disabled. It does not run until you enable it manually by rerunning the installation wizard.
    2. If you have Exchange in your on-premises Active Directory, then you also have an option to enable **Exchange Hybrid deployment**. Enable this option if you plan to have Exchange mailboxes both in the cloud and on-premises at the same time.
12. After the installation has completed, sign off and sign in again before you use Synchronization Service Manager or Synchronization Rule Editor.

**Password Writeback**

With password writeback, you can configure Azure Active Directory (Azure AD) to write passwords back to your on-premises Active Directory. Password writeback removes the need to set up and manage a complicated on-premises self-service password reset (SSPR) solution, and it provides a convenient cloud-based way for your users to reset their on-premises passwords wherever they want. Password writeback is a component of Azure Active Directory Connect that can be enabled and used by current subscribers of **Premium Azure Active Directory** editions. It's recommended that you use the auto-update feature of Azure AD Connect.

The following steps assume you have already configured Azure AD Connect in your environment by using the Express or Custom settings.

1. To configure and enable password writeback, sign in to your Azure AD Connect server and start the **Azure AD Connect** configuration wizard.
2. On the **Welcome** page, select **Configure**.
3. On the **Additional tasks** page, select **Customize synchronization options**, and then select **Next**.
4. On the **Connect to Azure AD** page, enter a global administrator credential, and then select **Next**.
5. On the **Connect directories** and **Domain/OU** filtering pages, select **Next**.
6. On the **Optional features** page, select the box next to **Password writeback** and select **Next**.



Microsoft Azure Active Directory Connect

Optional features

Select enhanced functionality if required by your organization.

- ☐ Exchange hybrid deployment
- ☐ Azure AD app and attribute filtering
- ☑ Password synchronization
- ☑ Password writeback
- ☐ Group writeback (Preview)
- ☐ Device writeback
- ☐ Directory extension attribute sync

7. On the **Ready to configure** page, select **Configure** and wait for the process to finish.
8. When you see the configuration finish, select **Exit**.

**The default synchronization frequency is 30 minutes**

**To customize the Scheduler Frequency for Sync Operation:**

https://docs.microsoft.com/en-us/azure/active-directory/hybrid/how-to-connect-sync-feature-scheduler

**To disable Azure AD Sync using PowerShell**

1. Create an ***an Azure global admin account with the*** *@*.onmicrosoft.com
2. Down and Install the powershell module from

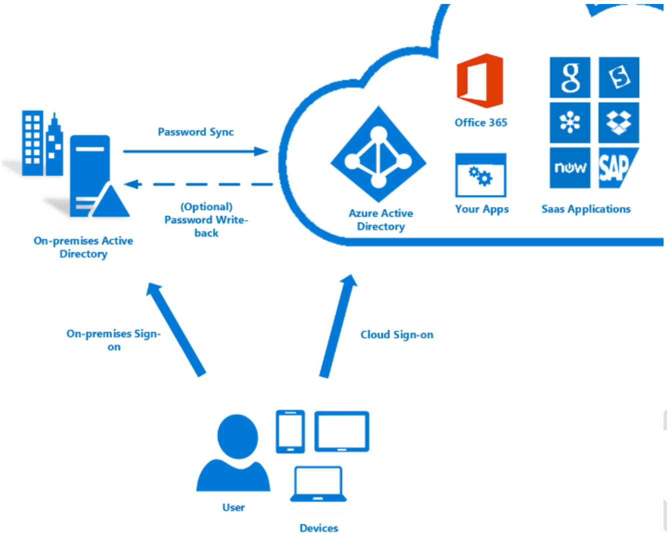
   http://connect.microsoft.com/site1164/Downloads/DownloadDetails.aspx?DownloadID=59185
3. Execute the following PowerShell commands

```
Install-Module MSOnline
Import-Module MSOnline
Install-Module AzureAD
Import-Module AzureAD
 #specify credentials for azure ad connect
$Msolcred = Get-credential
#connect to azure ad
Connect-MsolService -Credential $MsolCred
#disable AD Connect / Dir Sync
Set-MsolDirSyncEnabled –EnableDirSync $false
#confirm AD Connect / Dir Sync disabled
(Get-MSOLCompanyInformation).DirectorySynchronizationEnabled
```

## Azure AD User Sign-in Methods

**Password Hash synchronization with single sign-on** (SSO):

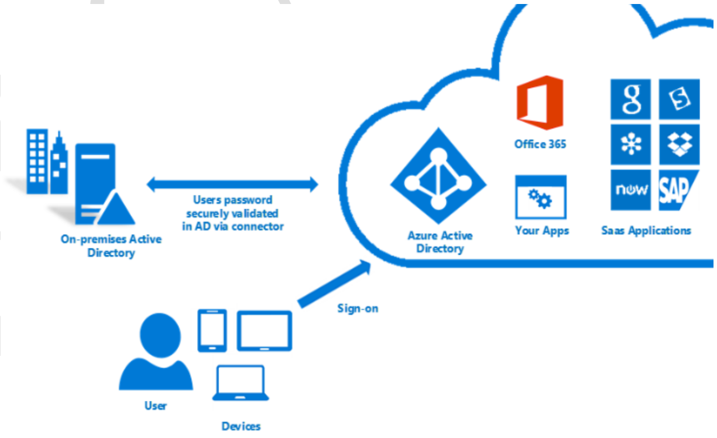
- With password synchronization, **hashes** of user passwords are synchronized from on-premises Active Directory to Azure AD.



- This ensures a user signing on to Azure uses the same password as the on-premises domain.
- When passwords are changed or reset on-premises, the new passwords are synchronized to Azure AD **immediately**.
- When you install Azure AD Connect by using the **Express Settings** option, password hash synchronization is automatically enabled.

**Pass-through authentication**:

- The user's password is validated against the on-premises Active Directory controller. The password doesn't need to be present in Azure AD in any form.

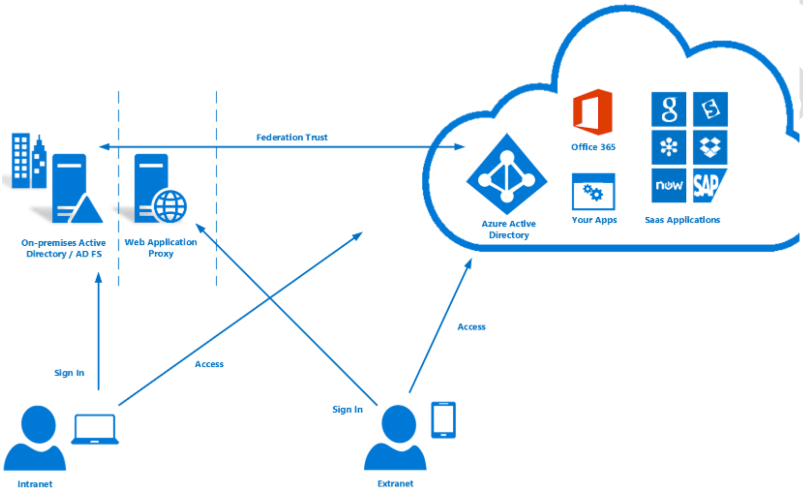

On-premises Active Directory
Users password securely validated in AD via connector
Office 365
Your Apps
Saas Applications
Azure Active Directory
Sign-on
User
Devices

- You need to choose Azure AD Connect **Custom Settings**. Password Synchronization in Optional Features tab must be unchecked.
- It uses a lightweight **on-premises agent** that listens for and responds to password validation requests.
- Agent has no management overhead. The agent automatically receives improvements and bug fixes.
- Additional agents can be installed on multiple on-premises servers to provide high availability of sign-in requests.

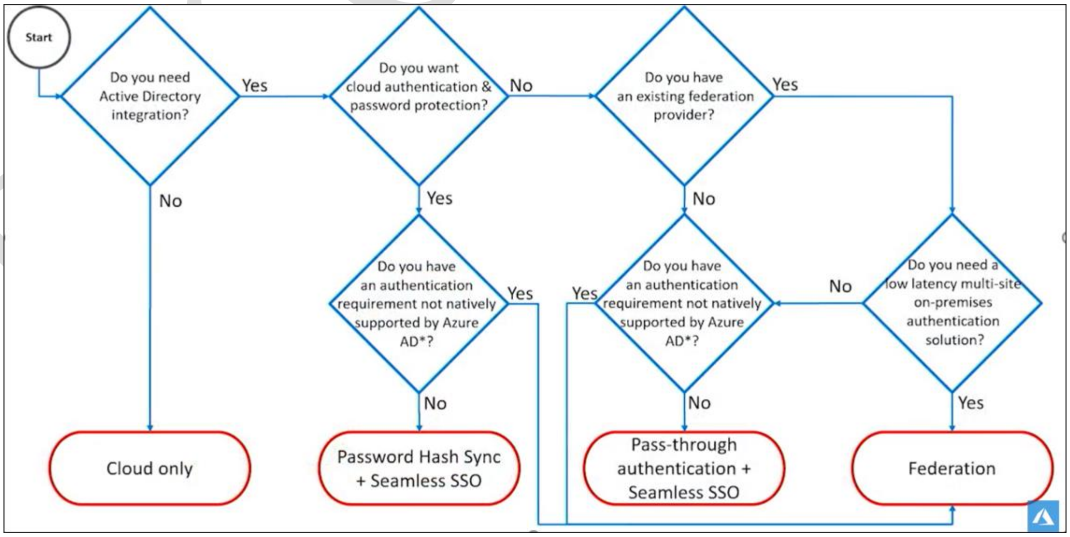
- The feature works seamlessly with **conditional access** features such as Multi-Factor Authentication (MFA) to help secure your users.
- Limitations:
  - Detection of users with leaked credentials.
  - Doesn't work for scenarios that need Azure AD Domain Services.

**Federated SSO (with Active Directory Federation Services (AD FS))**:

- 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.
- It uses Claims based Authentication

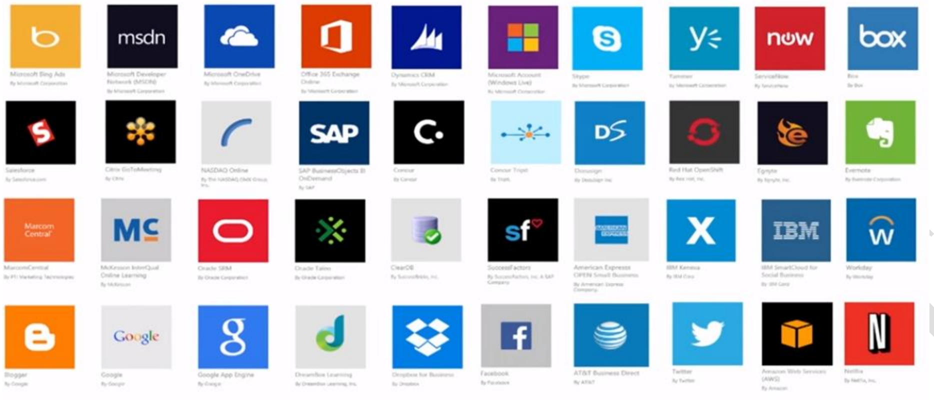


**Azure AD Authentication Decision Tree**


Start
Do you need Active Directory integration?
Yes
No
Do you want cloud authentication & password protection?
No
Yes
Do you have an existing federation provider?
Yes
No
Do you have an authentication requirement not natively supported by Azure AD*?
Yes
No
Do you have an authentication requirement not natively supported by Azure AD*?
Yes
No
Do you need a low latency multi-site on-premises authentication solution?
No
Yes
Cloud only
Password Hash Sync + Seamless SSO
Pass-through authentication + Seamless SSO
Federation

## Integrating SaaS Applications with Azure AD

Software as a service (SaaS) allows users to connect to and use cloud-based apps over the Internet. Common examples are email, calendaring, and office tools (such as Microsoft Office 365).



If you are going to deploy SaaS applications, then you will want your users to be able to use single-sign on (SSO). The Azure AD Application Gallery provides a listing of applications that are known to support a form of SSO with Azure AD.

**Azure AD gallery applications** provide automatic support for Azure AD. Therefore, the administrators do not need to provision user accounts manually for these applications.

Examples of gallery applications include Office 365, Dropbox for Business, GitHub for Business and Salesforce.

You can access the Azure AD application gallery from:

https://azuremarketplace.microsoft.com/en-us/marketplace/apps/category/azure-active-directory-apps?page=1

Here are some tips for finding apps by what capabilities they support:

- Featured applications support automatic provisioning and de-provisioning in Azure AD.
- Gallery applications support federated single sign-on using a protocol such as SAML, WS-Federation, or OpenID Connect.
- Each application in the gallery provides step-by-step instructions on how to enable single sign-on.

Automatic provisioning includes all the following:

- Automatically create new accounts in the right systems for new people when they join your team or organization.
- Automatically deactivate accounts in the right systems when people leave the team or organization.
- Ensure that the identities in your apps and systems are kept up-to-date based on changes in the directory, or your human resources system.
- Provision non-user objects, such as groups, to applications that support them.

**Integrating SaaS Applications**

1. Azure Portal → Azure Active Directory → Users and Add few users based on verified domain name or your default tenant id (XXXX.onmicrosoft.com)
2. Azure Portal → Azure Active Directory → **Enterprise Application**
3. + New Application → click All
4. Categories: Social, Under Add from the gallery, Search for **Facebook** / google / box→ Add

**Note that the steps for integrating SSO will vary from application to application and can be found in the documentation. Link for this documentation is available just above the Add button**

Adding Facebook app to Access Panel

5. Azure Portal → Azure Active Directory → Enterprise Application → All applications → **Facebook** → Settings
   1. **Single sign-on** → Mode = "Password-based Sign-on", Sign on URL = https://www.facebook.com (auto populated and is disabled)
   2. **Users and groups**: Add users/groups who can access.

   **Following action will allow the user to authenticate to the application from within the Access Panel with pre-configured identity.**

   3. Users and groups → Select User Click **Update Credentials** Button on top → Enter Email Address/Password to use → Save.
   4. **Self-service** (Optional)→
      - Allow users to request access to this application = Yes
      - To which group should assigned users be added = Facebook Guests
      - Require approval before granting access to this application? = Yes
      - Allow approvers to set user's passwords for this application? = yes
      - Who is allowed to approve access to this application? = <Primany domain users>
      - → . . . → Save
6. Azure Portal → Azure Active Directory → User Settings → Users can add gallery apps to their Access Panel = Yes (for Self service)

**Note:** Properties → **User assignment required**. (This option is only visible when the application is configured for the sign-on modes, SAML based SSO and WIA with Azure AD Authentication. This is not available for **Facebook** application and **Microsoft Account** (Windows Live)

**Testing the configuration:**

7. New Browser Window → Login to http://myapps.microsoft.com (Azure AD Access Panel) with any account already added to Azure AD and View that Facebook Application is listed.
8. Click on Facebook and Login (Only required for the first time and only if admin has not provided the u/p).
9. Close the Facebook page

10. Click again on Facebook and note that this time it will not ask for login.
11. To Add App to Access Panel (Self-Service)
    1. **1.** Access Panel → Click on UserName (top right) → Apps
        - Request access to app.
    2. **2.** The approver will receive an email to approve or reject the request.

Note:

1. **Dropbox** Gallery Application can be used with "**Federated Single Sign-On**".

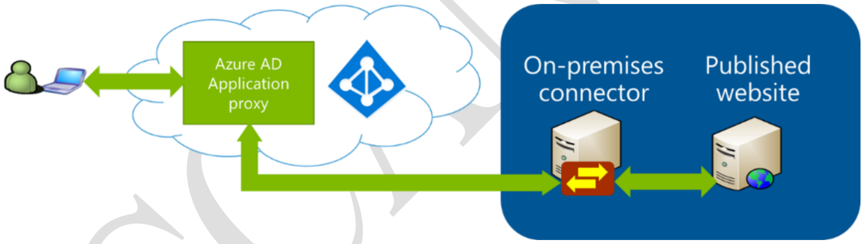
**Example2: GitHub Application**

**Tutorial: Azure Active Directory integration with GitHub**

https://docs.microsoft.com/en-us/azure/active-directory/saas-apps/github-tutorial

## Azure Active Directory Application Proxy

Users today need to be able to remotely access modern web applications hosted on-premises. They expect a single sign-on (SSO) and secure remote access experience. Azure AD Application Proxy is a feature of Azure Active Directory that provides remote access as a service, making it easy to deploy, use, and manage.



Typical apps that are published on-premises include SharePoint sites, Outlook Web Access, or any other LOB web applications your organization has. End users can access your on-premises applications the same way they access O365 and other SaaS apps integrated with Azure AD.

**Requirements for Application Proxy**

You **do not need** to change your existing network infrastructure or require VPN to implement Application Proxy for your on-premises users.

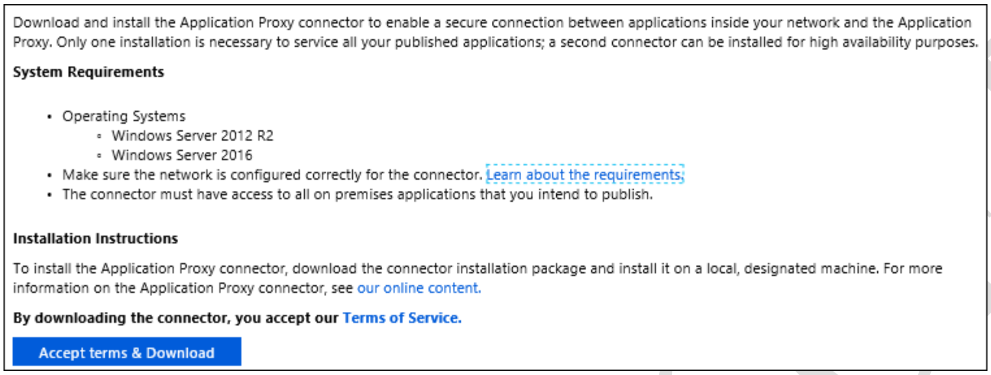
However, that are some requirements that should be noted.

- Application Proxy connector must be installed in the datacenter. One connector is required but two connectors are recommended for greater resiliency.
- Port 80 and port 443 are used for outbound connectivity. Note that no open inbound ports are required.
- An Azure subscription with Azure AD.
- One Global admin role.

- Windows Server 2012 R2 or higher on the on-premises connector.

**Configure On-premises Application:**

1. Azure AD → Enterprise Application → + New Application
2. Click on **On-premises application**
3. **Click Download Application Proxy Connector** install the same on Windows Server.



Download and install the Application Proxy connector to enable a secure connection between applications inside your network and the Application Proxy. Only one installation is necessary to service all your published applications; a second connector can be installed for high availability purposes.

**System Requirements**

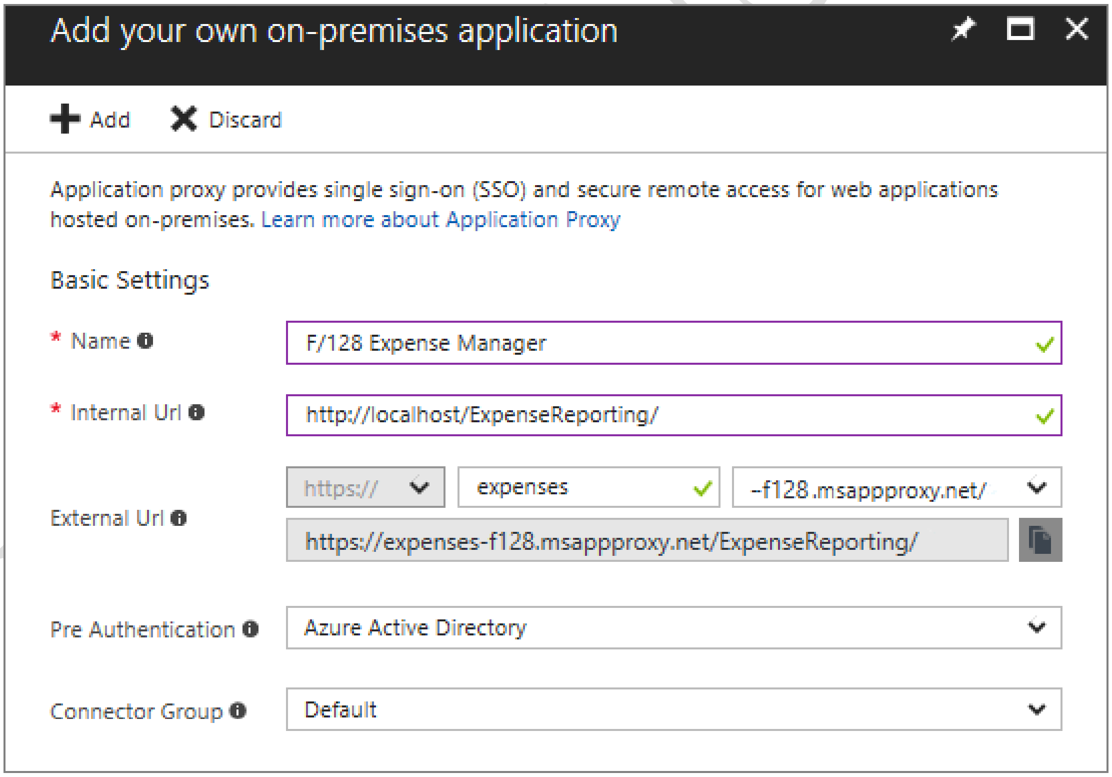
- Operating Systems
  - Windows Server 2012 R2
  - Windows Server 2016
- Make sure the network is configured correctly for the connector. Learn about the requirements.
- The connector must have access to all on premises applications that you intend to publish.

**Installation Instructions**

To install the Application Proxy connector, download the connector installation package and install it on a local, designated machine. For more information on the Application Proxy connector, see our online content.

**By downloading the connector, you accept our Terms of Service.**

4. In the **Add your own on-premises application** blade, provide the following information about your application:



Add your own on-premises application

Application proxy provides single sign-on (SSO) and secure remote access for web applications hosted on-premises. Learn more about Application Proxy

Basic Settings

* Name: F/128 Expense Manager

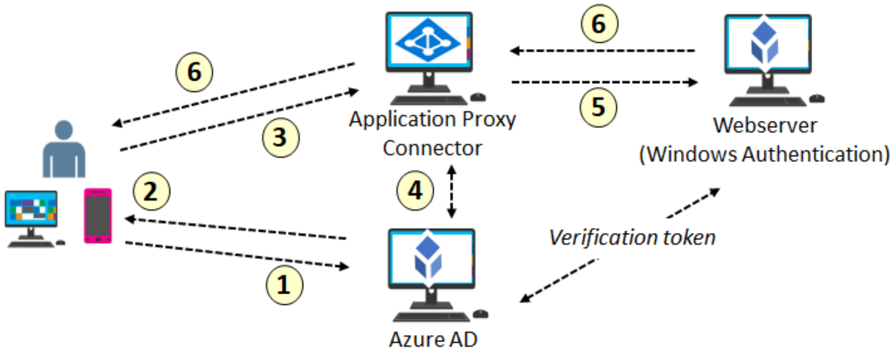
* Internal Url: http://localhost/ExpenseReporting/

External Url: https:// expenses -f128.msappproxy.net/
https://expenses-f128.msappproxy.net/ExpenseReporting/

Pre Authentication: Azure Active Directory

Connector Group: Default

**How Does Application Proxy Authentication Process Works?**


6
6
3
5
Application Proxy Connector
Webserver (Windows Authentication)
2
4
Verification token
1
Azure AD

1. The user accesses the application through the Application Proxy service and is directed to the Azure AD sign-in page to authenticate.
2. After a successful sign-in, a token is generated and sent to the 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, then directs 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 Application Proxy service and connector to the user.

**Demo**: https://youtu.be/6ZFed6DBhV0

## Azure Multi Factor Authentication (MFA)

Azure MFA helps safeguard access to data and applications while maintaining simplicity for users. It provides additional security by requiring a **second form of authentication** and delivers strong authentication through a range of easy to use authentication methods.

**There are two ways to enable MFA:**

- **The first option is to enable each user for MFA**. When users are enabled individually, they perform two-step verification each time they sign in. There are a few exceptions, such as when they sign in from trusted IP addresses or when the remembered devices feature is turned on.
- The second option is to set up a **conditional access policy** that requires two-step verification under certain conditions. This method uses the Azure AD Identity Protection risk policy to require two-step verification based only on the sign-in risk for all cloud applications.

**The Azure MFA authentication Process**

Sign in request will first be sent to Azure Active Directory for initial validation. If the correct credentials were entered and, validated, the request is then forwarded to Azure MFA authentication server. The Azure MFA server will then send an additional verification challenge to the user.

The methods that can be easy configured to use are:

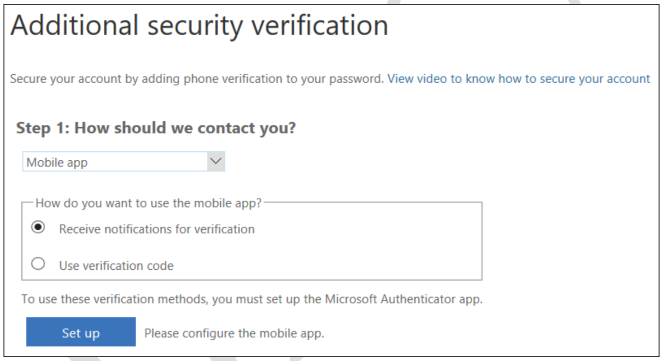
- **Phone Call.** A call is placed to the users register phone.
- **Text Message to phone**. A six-digit code is sent to the user's cell phone.
- **Mobile App Notification**. A verification request is sent to a user's smart phone asking them to complete the verification by selecting **Verify** in the mobile app.
- **Mobile app verification code**. A six-digit code is sent to the user **Microsoft Authenticator mobile app**. This code is then entered on the sign in page.

**Microsoft Authenticator App**

The Microsoft Authenticator app help **prevent unauthorized access** to accounts and to stop fraudulent transactions by giving you an additional level of security for your work or school account (for example, alain@contoso.com) or your personal Microsoft account (for example, alain@outlook.com). You can use it either as a **second verification method** or as a replacement for your password when using phone sign-in.

**When using the app for two-step verification, it can work in one of two ways:**

- **Notification**. The app sends a notification to your device. Make sure the notification is correct, and then select Verify. If you don't recognize the notification, select Deny.
- **Verification code**. After you type your username and password, you can open the app and copy the verification code provided on the Accounts screen on to the sign-in screen. The verification code acts as a second form of authentication.



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?

Mobile app

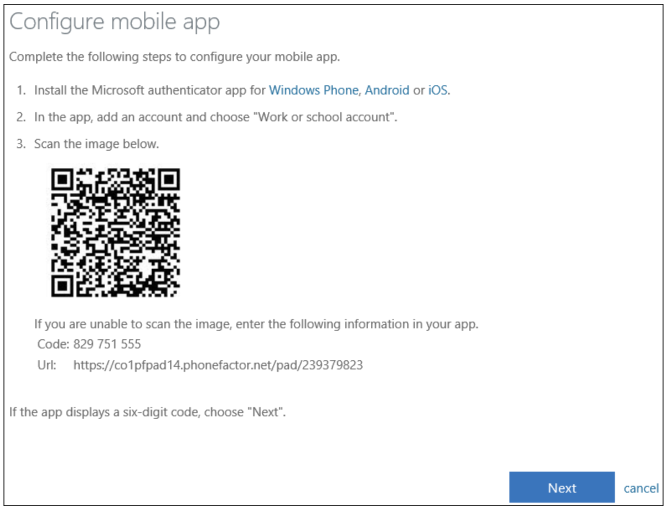
How do you want to use the mobile app?

● Receive notifications for verification

○ Use verification code

To use these verification methods, you must set up the Microsoft Authenticator app.

Please configure the mobile app.



Configure mobile app

Complete the following steps to configure your mobile app.

1. Install the Microsoft authenticator app for Windows Phone, Android or iOS.
2. In the app, add an account and choose "Work or school account".
3. Scan the image below.

If you are unable to scan the image, enter the following information in your app.
Code: 829 751 555
Url: https://co1pfpad14.phonefactor.net/pad/239379823

If the app displays a six-digit code, choose "Next".

**MFA Licensing and Pricing**

There are three pricing methods for Azure MFA.

Consumption based billing. Azure MFA is available as a stand-alone service with per-user and per-authentication billing options.

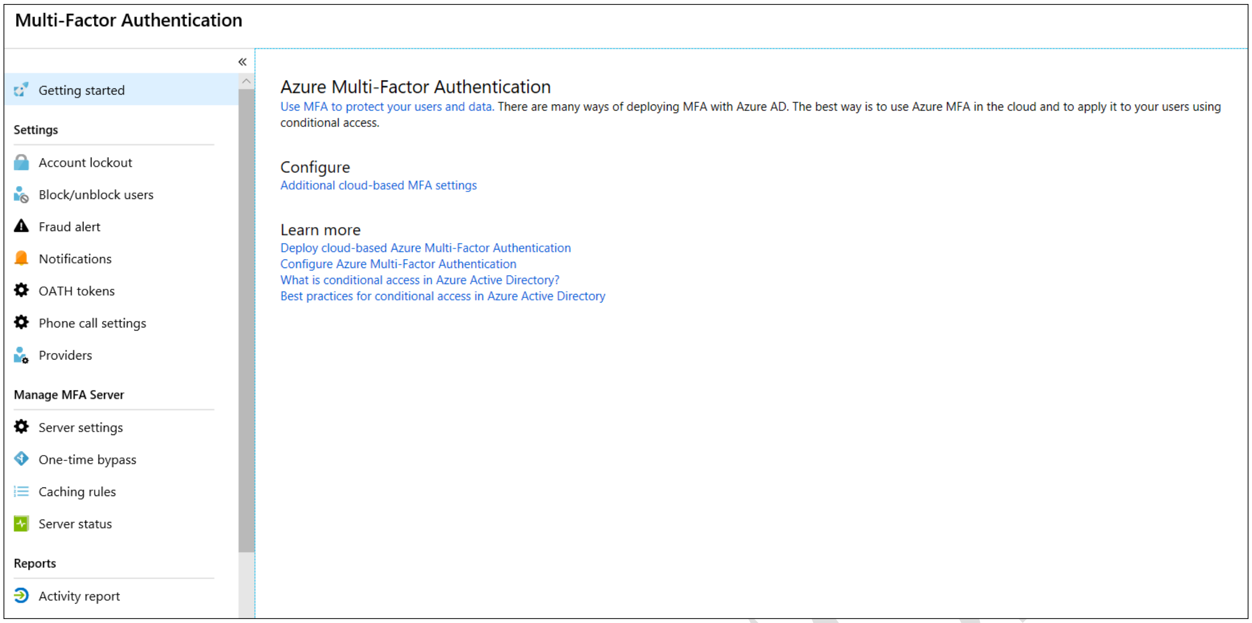
- **Per user**. You can pay per user. Each user has unlimited authentications. Use this model if you know how many users you have and can accurately estimate your costs.
- **Per authentication**. You can pay for a bundle (10) of authentications. Use this model when you are unsure how many users will participate in MFA authentication. MFA licenses included in other products. MFA is included in Azure AD Premium, Enterprise Mobility Suite, and Enterprise Cloud Suite.
- **Direct and Volume licensing**. MFA is available through a Microsoft Enterprise Agreement, the Open Volume License Program, the Cloud Solution Providers program, and Direct, as an annual user based model.

**To Configure MFA:**

Azure Active Directory → Users and groups → All users → **Multi-Factor Authentication**

OR

Azure Portal → Azure Active Directory → **MFA** → **Configure** → **Additional Cloud based MFA Settings**



Multi-Factor Authentication

Azure Multi-Factor Authentication

Use MFA to protect your users and data. There are many ways of deploying MFA with Azure AD. The best way is to use Azure MFA in the cloud and to apply it to your users using conditional access.

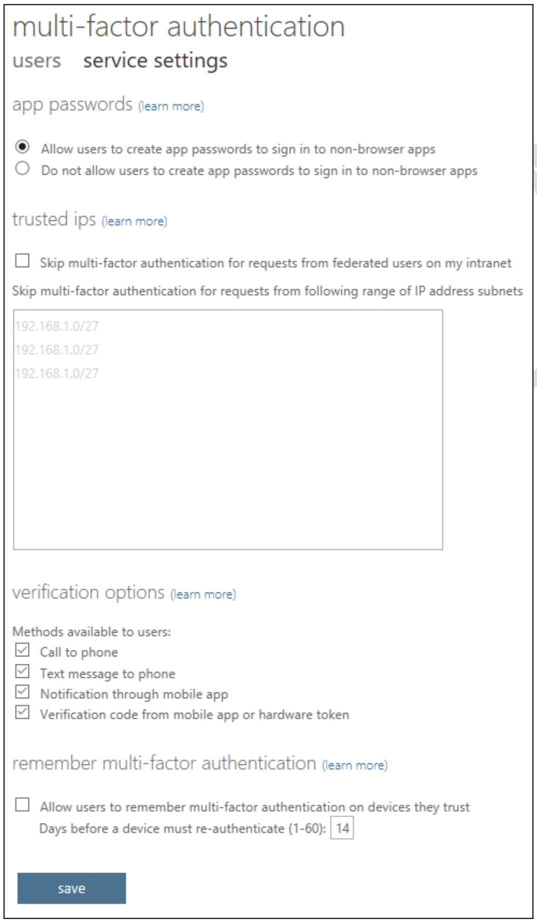
Configure

Additional cloud-based MFA settings

Learn more

Deploy cloud-based Azure Multi-Factor Authentication
Configure Azure Multi-Factor Authentication
What is conditional access in Azure Active Directory?
Best practices for conditional access in Azure Active Directory

## Service Settings



multi-factor authentication

users service settings

app passwords (learn more)

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

trusted ips (learn more)

□ Skip multi-factor authentication for requests from federated users on my intranet

Skip multi-factor authentication for requests from following range of IP address subnets

192.168.1.0/27
192.168.1.0/27
192.168.1.0/27

verification options (learn more)

Methods available to users:

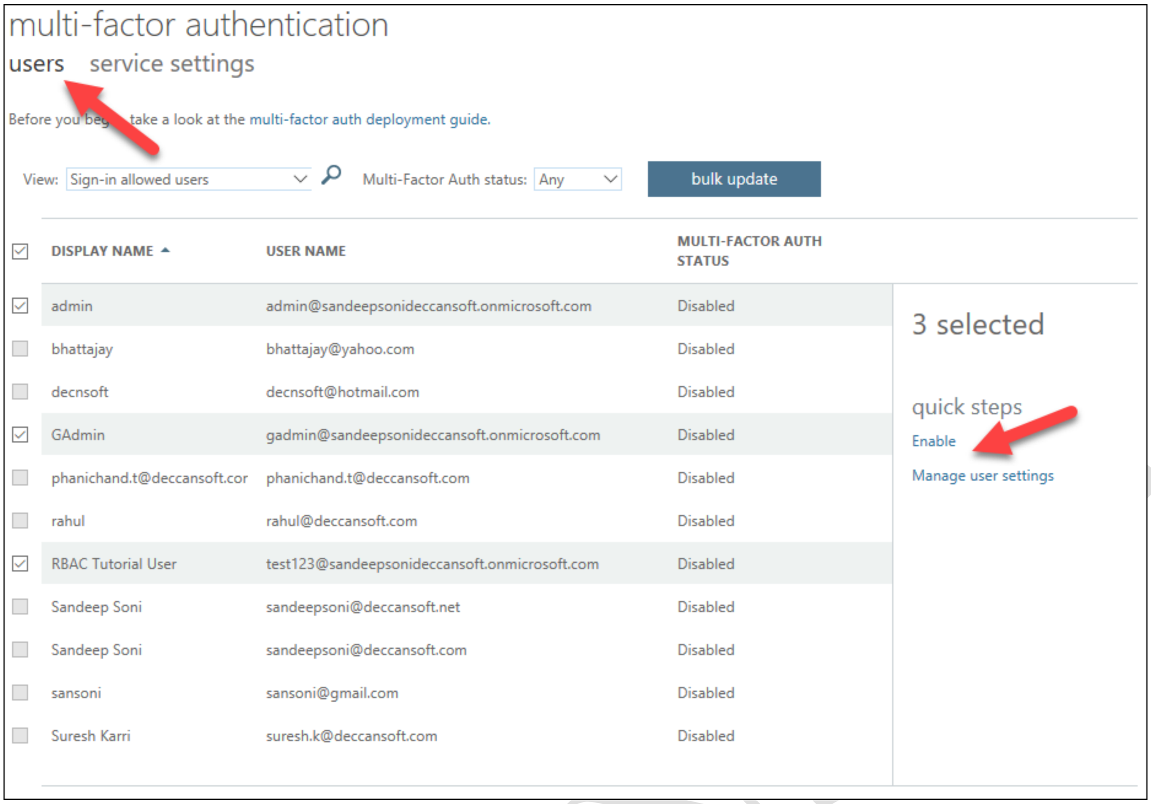
☑ Call to phone
☑ Text message to phone
☑ Notification through mobile app
☑ Verification code from mobile app or hardware token

remember multi-factor authentication (learn more)

□ Allow users to remember multi-factor authentication on devices they trust

Days before a device must re-authenticate (1-60): 14

## User Settings



multi-factor authentication

users service settings

Before you begin, take a look at the multi-factor auth deployment guide.

View: Sign-in allowed users Multi-Factor Auth status: Any

bulk update

| ☑ | DISPLAY NAME ▲ | USER NAME | MULTI-FACTOR AUTH STATUS |
|---|---|---|---|
| ☑ | admin | admin@sandeepsonideccansoft.onmicrosoft.com | Disabled |
| □ | bhattajay | bhattajay@yahoo.com | Disabled |
| □ | decnsoft | decnsoft@hotmail.com | Disabled |
| ☑ | GAdmin | gadmin@sandeepsonideccansoft.onmicrosoft.com | Disabled |
| □ | phanichand.t@deccansoft.cor | phanichand.t@deccansoft.com | Disabled |
| □ | rahul | rahul@deccansoft.com | Disabled |
| ☑ | RBAC Tutorial User | test123@sandeepsonideccansoft.onmicrosoft.com | Disabled |
| □ | Sandeep Soni | sandeepsoni@deccansoft.net | Disabled |
| □ | Sandeep Soni | sandeepsoni@deccansoft.com | Disabled |
| □ | sansoni | sansoni@gmail.com | Disabled |
| □ | Suresh Karri | suresh.k@deccansoft.com | Disabled |

3 selected

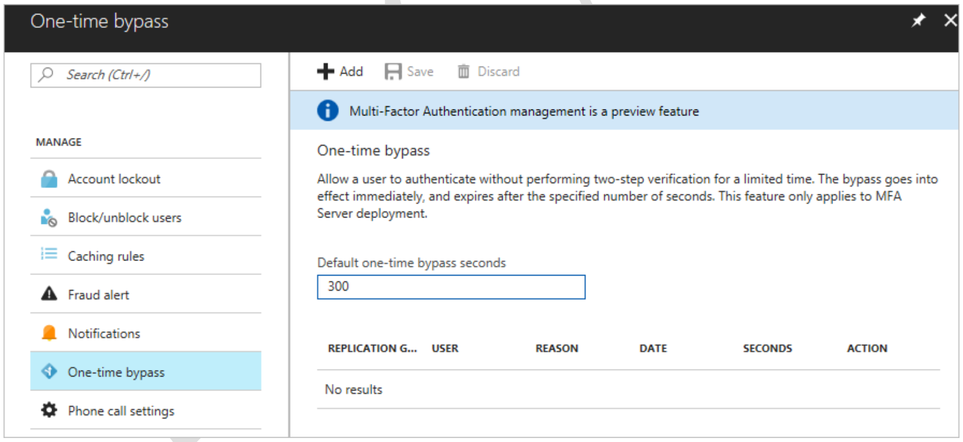
quick steps

Enable

Manage user settings

### One-time Bypass:

The one-time bypass feature allows a user to authenticate a single time without performing two-step verification. The bypass is temporary and expires after a specified number of seconds.



Multi-Factor Authentication management is a preview feature

One-time bypass

Allow a user to authenticate without performing two-step verification for a limited time. The bypass goes into effect immediately, and expires after the specified number of seconds. This feature only applies to MFA Server deployment.

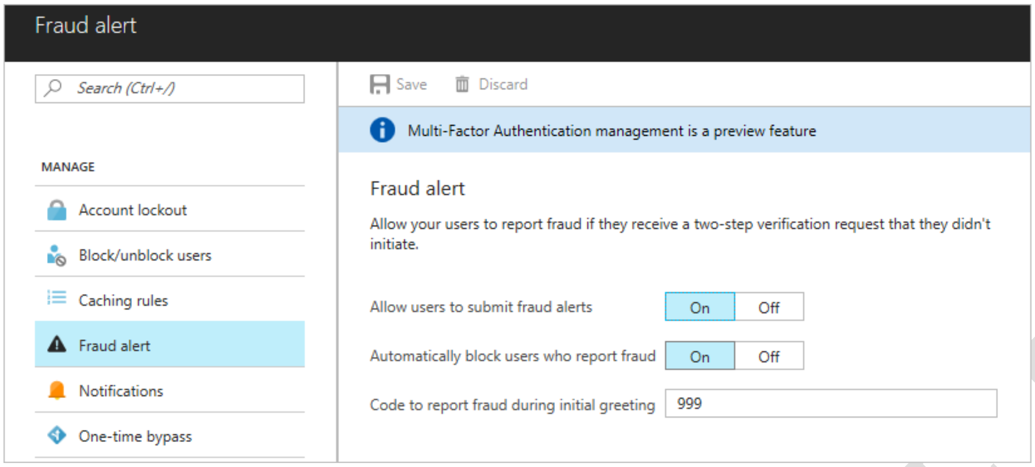
Default one-time bypass seconds: 300

| REPLICATION G... | USER | REASON | DATE | SECONDS | ACTION |
|---|---|---|---|---|---|
| No results | | | | | |

✓□ In situations where the mobile app or phone is not receiving a notification or phone call, you can allow a one-time bypass, so the user can access the desired resource.

### Fraud Alerts

Configure the fraud alert feature so that your users can report fraudulent attempts to access their resources. Users can report fraud attempts by using the mobile app or through their phone.



Fraud alert

Search (Ctrl+/)

MANAGE

- Account lockout
- Block/unblock users
- Caching rules
- Fraud alert
- Notifications
- One-time bypass

Save Discard

Multi-Factor Authentication management is a preview feature

Fraud alert

Allow your users to report fraud if they receive a two-step verification request that they didn't initiate.

Allow users to submit fraud alerts [✓ On] Off

Automatically block users who report fraud [✓ On] Off

Code to report fraud during initial greeting 999

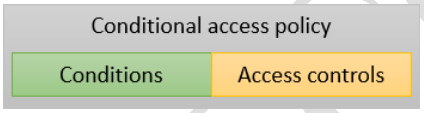
**Block user when fraud is reported**. If a user reports fraud, their account is blocked for 90 days or until an administrator unblocks their account. An administrator can review sign-ins by using the sign-in report and take appropriate action to prevent future fraud. An administrator can then unblock the user's account.

**Code to report fraud during initial greeting**. When users receive a phone call to perform two-step verification, they normally press # to confirm their sign-in. To report fraud, the user enters a code before pressing #. This code is 0 by default, but you can customize it.

## Conditional Access Policy

Conditional access is a capability of Azure AD (with an Azure AD Premium license) that enables you to enforce controls on the access to apps in your environment based on **specific conditions** from a central location.

With Azure AD conditional access, you can factor how a resource is being accessed into an access control decision. By using **conditional access policies**, you can apply the right access controls under the required conditions.



Conditional access policy

Conditions | Access controls

In the context of conditional access:

- "**When this happens**" is called conditions.
- "**Then do this**" is called access controls.

With access controls, you can either **Block Access** altogether or **Grant Access** with **additional requirements** by selecting the desired controls. You can have several options:

- Require MFA from Azure AD or an on-premises MFA (combined with AD FS).
- Grant access to only trusted devices.
- Require a domain-joined device.
- Require mobile devices to use Intune app protection policies.

**Walkthrough: Configuring Condition Access Policy**

**Pre-requisite:**

1. Create a non-administrator test user with a password you know for testing.

**Enable Azure Multi-Factor Authentication**

1. Sign in to the Azure portal using a Global Administrator account.
2. Browse to **Azure Active Directory**, **Conditional access**
3. Select **New policy**
4. Name your policy **MFA Pilot**
5. Under **users and groups**, select the **Select users and groups** radio button
   - Select your pilot group created as part of the prerequisites section of this article
   - Click **Done**
6. Under **Cloud apps**, select the **Select apps** radio button
   - The cloud app for the Azure portal is **Microsoft Azure Management**
   - Click **Select**
   - Click **Done**
7. Skip the **Conditions** section
8. Under **Grant**, make sure the **Grant access** radio button is selected
   - Check the box for **Require multi-factor authentication**
   - Click **Select**
9. Skip the **Session** section
10. Set the **Enable policy** toggle to **On**
11. Click **Create**

**Test Azure Multi-Factor Authentication**

To prove that your conditional access policy works, you test logging in to a resource that should not require MFA and then to the Azure portal that requires MFA.

1. Open a new browser window in InPrivate or incognito mode and browse to https://account.activedirectory.windowsazure.com.
   - Log in with the test user created as part of the prerequisites section of this article and note that it should not ask you to complete MFA.
   - Close the browser window.
2. Open a new browser window in InPrivate or incognito mode and browse to https://portal.azure.com.
   - Log in with the test user created as part of the prerequisites section of this article and note that you should now be required to register for and use Azure Multi-Factor Authentication.
   - Close the browser window.

Detailed Steps: https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/app-based-mfa#create-your-conditional-access-policy

## Azure AD Identity Protection

Azure Active Directory Identity Protection is a feature of the **Azure AD Premium P2 edition** that enables you to **detect and prevent against Identity attacks**.

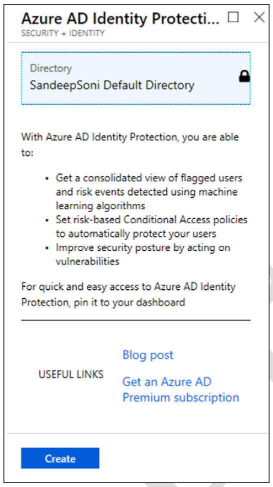
Every time Microsoft gets a sign in request, they look at the IP Address, the location, the user agent, the user's sign in pattern in the past and based on that they determine if the user is **good or bad** and there are policies which can kick in automatically to act against that.

### Benefits

1. Get a **consolidated view** to examine suspicious user activities detected using **Identity Protection machine learning algorithms** with signals like brute force attacks, leaked credentials, and sign-ins from unfamiliar locations.
2. **Improve the security posture** of your organization by acting on a customized list of configuration vulnerabilities that could lead to an elevated risk of account compromise in your organization.
3. **Set risk-based Conditional Access policies** to automatically protect your users.

### Enabling Azure Active Directory Identity Protection

Azure Portal → All Services → **Azure AD Identity Protection** → Settings Blade → **Onboard** → **Create**



Azure AD Identity Protecti...
SECURITY + IDENTITY

Directory
SandeepSoni Default Directory

With Azure AD Identity Protection, you are able to:

- Get a consolidated view of flagged users and risk events detected using machine learning algorithms
- Set risk-based Conditional Access policies to automatically protect your users
- Improve security posture by acting on vulnerabilities

For quick and easy access to Azure AD Identity Protection, pin it to your dashboard

USEFUL LINKS
Blog post
Get an Azure AD Premium subscription

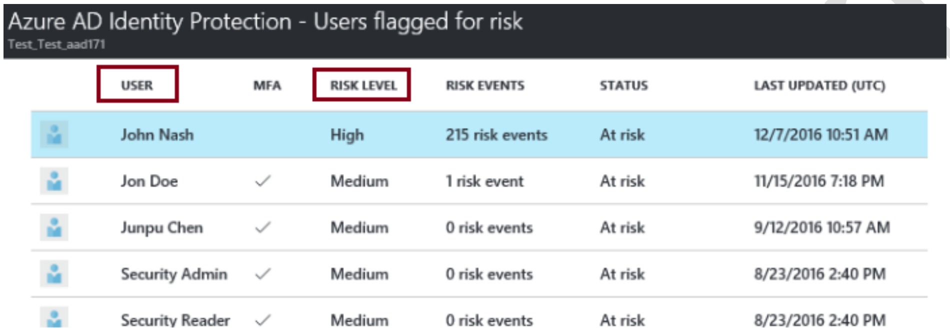
### What can Azure AD Identity Protection do?

1. **Discover Users Flagged for Risk:** Detect users flagged for risk and investigate risk events for the user.
2. **Discover Risk Events:** Detect and investigate risk events like users with leaked credentials, sign-ins from anonymous ip address, etc.
3. **Discover Vulnerabilities:** Detect weaknesses in your environment that you can fix to improve your security posture.

4. **Mitigate Risk Events:** Enable policy to require multi-factor authentication or block sign-in based on sign-in risk.
5. **Remediate Users:** Manually password reset for a user or enable policy for password reset or blocking sign-in based on user risk.

### Discover Users Flagged for Risk

With the security reports in Azure Active Directory (Azure AD) you can gain insights into the probability of compromised user accounts in your environment. Azure AD detects suspicious actions that are related to your user accounts. For each detected action, a record called risk event is created.



Azure AD Identity Protection - Users flagged for risk

Test_Test_aad171

| USER | MFA | RISK LEVEL | RISK EVENTS | STATUS | LAST UPDATED (UTC) |
|---|---|---|---|---|---|
| John Nash | | High | 215 risk events | At risk | 12/7/2016 10:51 AM |
| Jon Doe | ✓ | Medium | 1 risk event | At risk | 11/15/2016 7:18 PM |
| Junpu Chen | ✓ | Medium | 0 risk events | At risk | 9/12/2016 10:57 AM |
| Security Admin | ✓ | Medium | 0 risk events | At risk | 8/23/2016 2:40 PM |
| Security Reader | ✓ | Medium | 0 risk events | At risk | 8/23/2016 2:40 PM |

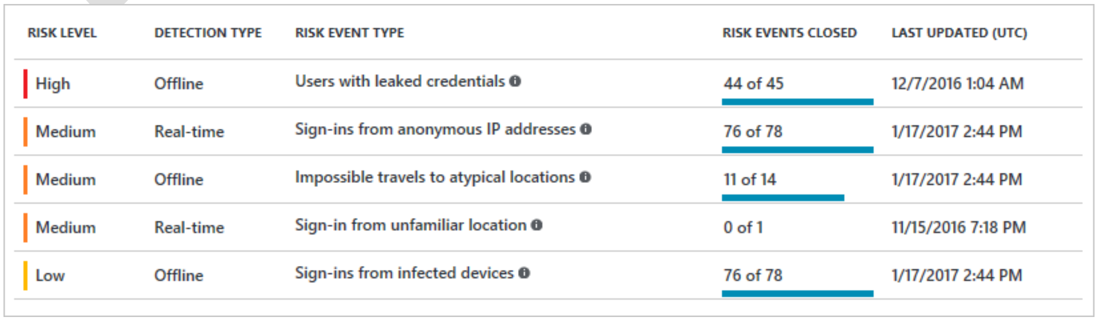
Risk events are used to calculate:

- **Users flagged for risk.** A risky user is an indicator for a user account that might have been compromised.
- **Risky sign-ins**. A risky sign-in is an indicator for a sign-in attempt that might have been performed by someone who is not the legitimate owner of a user account. A sign-in risk level is an indication (High, Medium, or Low) of the likelihood that a sign-in attempt was made by someone other than the legitimate owner of the user account.

✓□ Azure AD Identity Protection sends two types of automated notification emails to help you manage user risk and risk events: users at risk detected email, and a weekly digest email.

### Discover Risk Events / Risks Detected

Most security breaches take place when attackers gain access to an environment by stealing a user's identity. Discovering compromised identities is no easy task. Azure Active Directory uses adaptive machine learning algorithms and heuristics to detect **suspicious actions** that are related to your user accounts. Each detected suspicious action is stored in a record called risk event.



| RISK LEVEL | DETECTION TYPE | RISK EVENT TYPE | RISK EVENTS CLOSED | LAST UPDATED (UTC) |
|---|---|---|---|---|
| High | Offline | Users with leaked credentials | 44 of 45 | 12/7/2016 1:04 AM |
| Medium | Real-time | Sign-ins from anonymous IP addresses | 76 of 78 | 1/17/2017 2:44 PM |
| Medium | Offline | Impossible travels to atypical locations | 11 of 14 | 1/17/2017 2:44 PM |
| Medium | Real-time | Sign-in from unfamiliar location | 0 of 1 | 11/15/2016 7:18 PM |
| Low | Offline | Sign-ins from infected devices | 76 of 78 | 1/17/2017 2:44 PM |

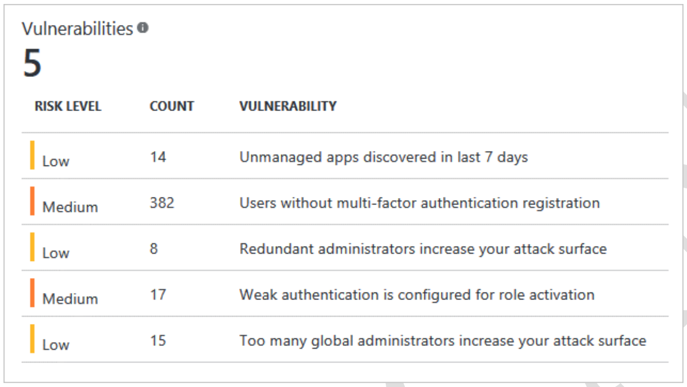
Currently, Azure Active Directory detects six types of risk events:

1. Users with leaked credentials
2. Sign-ins from anonymous IP addresses
3. Impossible travel to atypical locations
4. Sign-ins from infected devices
5. Sign-ins from IP addresses with suspicious activity
6. Sign-ins from unfamiliar locations

**Vulnerabilities Detected**

Vulnerabilities are weaknesses in your environment that can be exploited by an attacker. We recommend that you address these vulnerabilities to improve the security posture of your organization and prevent attackers from exploiting them.

**Azure AD Identity Protection - Vulnerabilities**



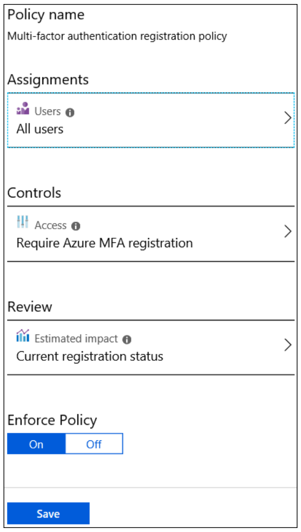
Vulnerabilities
5

| RISK LEVEL | COUNT | VULNERABILITY |
|---|---|---|
| Low | 14 | Unmanaged apps discovered in last 7 days |
| Medium | 382 | Users without multi-factor authentication registration |
| Low | 8 | Redundant administrators increase your attack surface |
| Medium | 17 | Weak authentication is configured for role activation |
| Low | 15 | Too many global administrators increase your attack surface |

**Configure MFA registration policy**

Azure multi-factor authentication is a method of verifying who you are that requires the use of more than just a username and password. It provides a second layer of security to user sign-ins and transactions.

**Steps:** Azure AD Identity Protection → **MFA registration →** Set the values as specified below



Policy name
Multi-factor authentication registration policy

Assignments

Users
All users

Controls

Access
Require Azure MFA registration

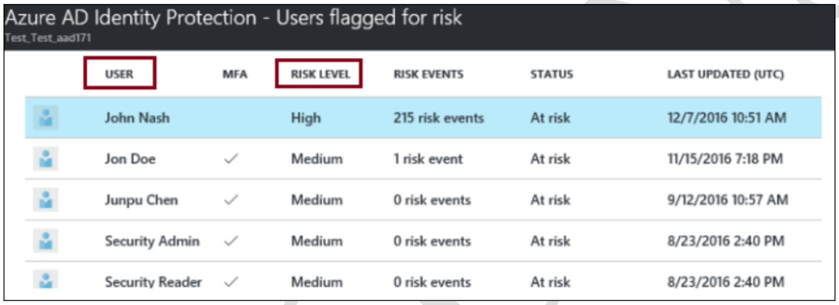
Review

Estimated impact
Current registration status

Enforce Policy

### Configure the sign-in risk policy

Azure AD analyzes each sign-in of a user. The objective of the analysis is to detect suspicious actions that come along with the sign-in. For example, is the sign-in done using an anonymous IP address, or is the sign-in initiated from an unfamiliar location.



Azure AD Identity Protection - Users flagged for risk
Test_Test_aad171

| USER | MFA | RISK LEVEL | RISK EVENTS | STATUS | LAST UPDATED (UTC) |
|---|---|---|---|---|---|
| John Nash | | High | 215 risk events | At risk | 12/7/2016 10:51 AM |
| Jon Doe | ✓ | Medium | 1 risk event | At risk | 11/15/2016 7:18 PM |
| Junpu Chen | ✓ | Medium | 0 risk events | At risk | 9/12/2016 10:57 AM |
| Security Admin | ✓ | Medium | 0 risk events | At risk | 8/23/2016 2:40 PM |
| Security Reader | ✓ | Medium | 0 risk events | At risk | 8/23/2016 2:40 PM |

### Walkthrough: Block access when a session risk is detected with Azure Active Directory Identity Protection

1. Install **Tor Browser** on your machine - The Tor Browser is designed to help you preserve your privacy online. Identity Protection detects a sign-in from a Tor Browser as **sign-ins from anonymous IP addresses**, which has a medium risk level.
2. Create a New User in Azure AD**,** username = "TestUser@domainname.com"
3. Azure AD Identity Protection → **Sign-in risk policy** → in the Assignments section:
    1. Select users → Select Test User → Select → Done
    2. Conditions → Sign-in risk level = Medium and above → Select → Done.
    3. Controls → Select Allow access → check Require multi-factor authentication

   **Note:** For security reasons, **Require multi-factor authentication** setting works only for users that have already been registered for MFA. Identity protection **blocks** users with an MFA requirement if they are **not registered** for MFA yet.
4. Enfoce Policy = On → Save

5. To test your policy, try to sign-in to your Azure portal as **Test User** using the Tor Browser. Your sign-in attempt should be blocked by your conditional access policy.

**Configure the user risk policy**

With the user risk, Azure AD detects the probability that a user account has been compromised. As an administrator, you can configure a user risk conditional access policy, to automatically respond to a specific user risk level.

## Managed Identities

- A common challenge when building cloud applications is how to manage the credentials in your code for authenticating to cloud services. Keeping the credentials secure is an important task. Ideally, the credentials never appear on developer workstations and aren't checked into source control. Azure Key Vault provides a way to securely store credentials, secrets, and other keys, but your code has to authenticate to Key Vault to retrieve them.
- The managed identities for Azure resources feature in Azure Active Directory (Azure AD) solves this problem. The feature provides Azure services with an automatically managed identity in Azure AD. You can use the identity to authenticate to any service that supports Azure AD authentication, including Key Vault, without any credentials in your code.
- The managed identities for Azure resources feature is free with Azure AD for Azure subscriptions. There's no additional cost.

**There are two types of managed identities:**

- A **system-assigned managed identity** is enabled directly on an Azure service instance. When the identity is enabled, Azure creates an identity for the instance in the Azure AD tenant that's trusted by the subscription of the instance. After the identity is created, the credentials are provisioned onto the instance. The lifecycle of a system-assigned identity is directly tied to the Azure service instance that it's enabled on. If the instance is deleted, Azure automatically cleans up the credentials and the identity in Azure AD.
- A **user-assigned managed identity** is created as a standalone Azure resource. Through a create process, Azure creates an identity in the Azure AD tenant that's trusted by the subscription in use. After the identity is created, the identity can be assigned to one or more Azure service instances. The lifecycle of a user-assigned identity is managed separately from the lifecycle of the Azure service instances to which it's assigned.

**Example:**

This tutorial shows you how to use a system-assigned managed identity for a Windows virtual machine (VM) to access Azure Key Vault.

1. Create a VM with Identity Management enabled
2. Create a Key Value and add a Secret
3. Key Vault → Select **Access policies** and click **Add new**.
4. In Configure from template, select **Secret Management**.
5. Choose **Select Principal**, and in the search field enter the name of the VM you created earlier. Select the VM in the result list and click **Select**.
6. RDP to VM
7. Invoke the web request on the tenant to get the token for the local host in the specific port for the Windows VM.

```
$response = Invoke-WebRequest -Uri 'http://169.254.169.254/metadata/identity/oauth2/token?api-
version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net' -Method GET -Headers @{Metadata="true"}
$content = $response.Content | ConvertFrom-Json
$KeyVaultToken = $content.access_token
```

If Linux VM is used.

```
curl 'http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-
01&resource=https%3A%2F%2Fvault.azure.net' -H Metadata:true
curl https://<YOUR-KEY-VAULT-URL>/secrets/<secret-name>?api-version=2016-10-01 -H "Authorization:
Bearer <ACCESS TOKEN>"
```

8. Finally, use PowerShell's Invoke-WebRequest command to retrieve the secret you created earlier in the Key Vault, passing the access token in the Authorization header.

```
(Invoke-WebRequest -Uri https://<your-key-vault-URL>/secrets/<secret-name>?api-version=2016-10-01 -
Method GET -Headers @{Authorization="Bearer $KeyVaultToken"}).content
```

**In C#**

Add references to the Microsoft.Azure.Services.AppAuthentication and any other necessary NuGet packages to your application. The below example also uses Microsoft.Azure.KeyVault.

```
using Microsoft.Azure.Services.AppAuthentication;
using Microsoft.Azure.KeyVault;
// ...
var azureServiceTokenProvider = new AzureServiceTokenProvider();
string accessToken = await azureServiceTokenProvider.GetAccessTokenAsync("https://vault.azure.net");
// OR
var kv = new KeyVaultClient(new
KeyVaultClient.AuthenticationCallback(azureServiceTokenProvider.KeyVaultTokenCallback));
```