

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
- Administrative Units
- Configure Windows 10 with Azure AD domain join
- Synchronizing On-Premise AD Identities with Azure AD
- Azure AD Connect
- Azure AD User Sign-In Options
 - Password Synchronization
 - Passthrough Authentication
 - Federated SSO
- Integrating SaaS Applications with Azure AD for SSO
 - Add Users and Groups to Application
 - Revoke access to SaaS Applications
- Multi Factory Authentication
- Conditional Access Policy
- Access Reviews
- Privileged Identity Management
- Managed Identities

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 (RBAC).
- 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 (MFA).
- Extend existing on-premises Active Directory implementations to Azure AD.

Azure AD is different from AD DS: It is important to realize that using Azure AD is different from deploying an Active Directory domain controller on an Azure virtual machine and adding it to your on-premises domain.

- **Identity solution.** Azure AD is primarily an identity solution, and it is designed for Internet-based applications by using HTTP and HTTPS communications.
- **REST API Querying.** Because Azure AD is HTTP/HTTPS based, it cannot be queried through LDAP. Instead, Azure AD uses the REST API over HTTP and HTTPS.
- **Communication Protocols.** Because Azure AD is HTTP/HTTPS based, it does not use Kerberos authentication. Instead, it uses HTTP and HTTPS protocols such as SAML, WS-Federation, and OpenID Connect for authentication (and OAuth for authorization).
- **Federation Services.** Azure AD includes federation services, and many third-party services (such as Facebook).
- **Flat structure.** Azure AD users and groups are created in a flat structure, and there are no Organizational Units (OUs) or Group Policy Objects (GPOs).

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**.
- **Office 365 apps 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)
- **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)
 - Azure Active Directory Application Proxy (to publish on-premises web applications using Azure Active Directory)

- 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.

Comparison Between Editions

<https://azure.microsoft.com/en-in/pricing/details/active-directory>

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, deccansoftoutlook.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 On-Premise 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 = **DemoOrg**.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 of Existing Subscription**1. Logged-In with sandeepsoni@deccansoft.com**

2. Created a New Azure AD Tenant - decnsoft5.onmicrosoft.com

Azure Active Directory --> Add Tenant

3. Changed Directory of Existing Subscription from Old Tenant (sandeepsonideccansoft.onmicrosoft.com to New Tenant(decnsoft5.onmicrosoft.com)

Subscriptions --> Select the Subscription --> Change Directory

4. Switched to New Tenant (decnsoft5.onmicrosoft.com)

Toolbar --> Subscription + Directory

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/>

Create a New Outlook.com Account

Assign an Azure Subscription (FREE Trial / Azure Pass – Sponsorship)

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.
9. LOGOUT and LOGIN again to apply the changes.

To Assign License - Any Global Administrator can Login and do the following

5. Activated Premium P2 License

Azure Active Directory -> Licenses --> All Products --> +Add --> Premium P2

6. Created Users

Active Directory --> Users --> Add

a) user1@decnsoft5.onmicrosoft.com/Test@123 - Location=India

b) admin1@decnsoft5.onmicrosoft.com/Test@123 - Role=Global Administrator, Location=India

7. Assign License to Users

Azure Active Directory -> Licenses --> All Products --> Azure Active Directory Premium P2 --> Assign and assign license to required users.

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.

Step: 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. Guest 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

- **Office 365 groups** (recommended) are a great way for teams to collaborate by giving them a group email and a shared workspace for conversations, files, and calendar events.
- **Security groups** control access to OneDrive and SharePoint and are used for Mobile Device Management for Office 365.

Steps to create a Group

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

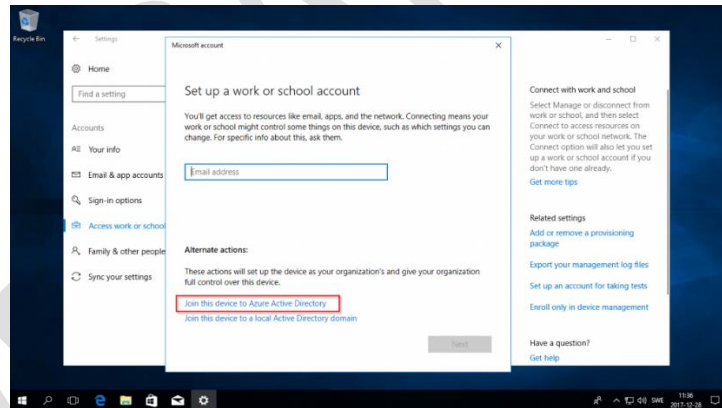
Users may join devices to Azure AD = All

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



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.

Administrative Units

- An administrative unit is an Azure AD resource that can be a container for other Azure AD resources.
- An administrative unit can contain only users and groups.

As a Global Administrator or a Privileged Role Administrator, you can use the Azure portal to:

- Create administrative units
- Add users and groups members of administrative units
- Assign IT staff to administrative unit-scoped administrator roles.

Note: Adding and removing administrative unit members dynamically based on attributes is not supported.

Available roles

Role	Description
Authentication Administrator	Has access to view, set, and reset authentication method information for any non-admin user in the assigned administrative unit only.
Groups Administrator	Can manage all aspects of groups and groups settings, such as naming and expiration policies, in the assigned administrative unit only.
Helpdesk Administrator	Can reset passwords for non-administrators and Helpdesk Administrators in the assigned administrative unit only.
License Administrator	Can assign, remove, and update license assignments within the administrative unit only.
Password Administrator	Can reset passwords for non-administrators and Password Administrators within the assigned administrative unit only.
User Administrator	Can manage all aspects of users and groups, including resetting passwords for limited admins within the assigned administrative unit only.

Example: It can be useful to restrict administrative scope by using administrative units in organizations that are made up of independent divisions of any kind. Consider the example of a large university that's made up of many autonomous schools (School of Business, School of Engineering, and so on). Each school has a team of IT admins who control access, manage users, and set policies for their school.

A central administrator could:

- Create an administrative unit for the School of Business.
- Populate the administrative unit with only the business school students and staff.
- Add the business school IT team to the role, along with its scope.

License requirements

Using administrative units requires an Azure AD Premium P1 license for each administrative unit administrator, and Azure AD Free licenses for administrative unit members.

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

Configure Self Service Group Management

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).

1. Azure AD → Groups → General →

Self Service Group Management

Owners can manage group membership requests in the Access Panel

Restrict access to Groups in the Access Panel

Security Groups

Users can create security groups in Azure portals

Owners who can assign members as group owners in Azure portals

Group that can manage security groups
No group selected

Office 365 Groups

Users can create Office 365 groups in Azure portals

Owners who can assign members as group owners in Azure portals

Group that can manage Office 365 groups
No group selected

Directory-wide Groups

Enable an "All Users" group in the directory

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

<http://www.mustbegeek.com/install-active-directory-in-windows-server-2012/>

1. Create a new VM (DomainController-vm) 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 = → 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 and Features** → 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**

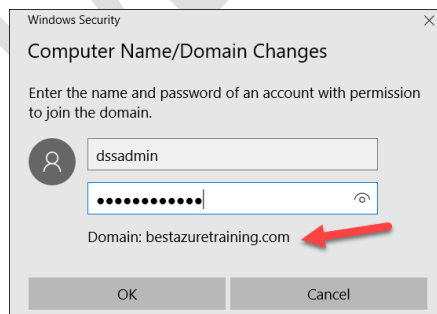
Joining a Workstation to Domain Controller

7. Create a new VM (DemoWK1-vm) 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.



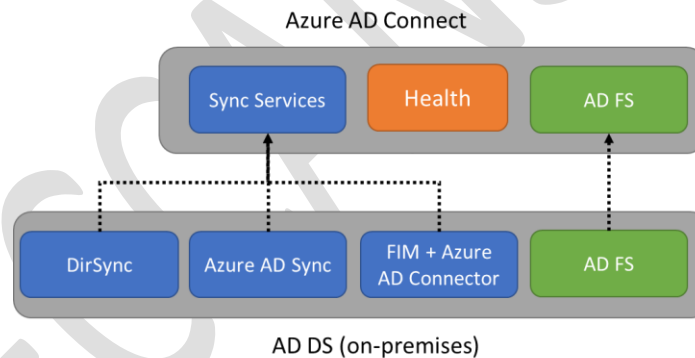
5. To Remote Desktop Login with new Identity on this VM (DemoWK1-vm)

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.

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.

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 Enhanced 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@sandeepsoni.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.

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 Azure global admin account with the ***@*.onmicrosoft.com**
example: admin@sandeepsonideccansoft.onmicrosoft.com
2. Execute the following PowerShell commands

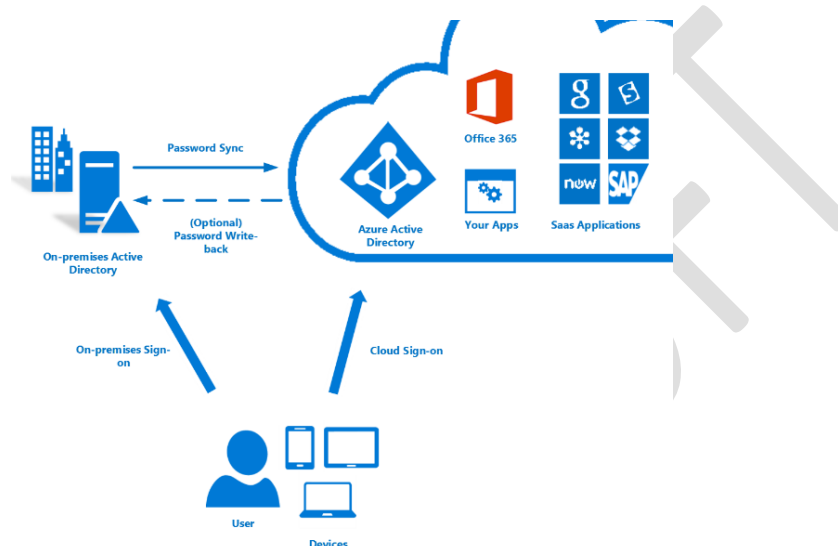
```
Install-Module MSOnline
Import-Module MSOnline
Install-Module AzureAD
Import-Module AzureAD
#specify credentials for azure ad connect
$Msolcred = Get-credential
#connect to azure ad
Connect-MsolService -Credential $MsolCred
#disable AD Connect / Dir Sync
Set-MsolDirSyncEnabled -EnableDirSync $false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

- Use this feature to sign in to Azure AD services like Office 365, Microsoft Intune, CRM Online, and Azure Active Directory Domain Services (Azure AD DS).
- In the background, the password synchronization component takes the user's password hash from on-premises Active Directory, encrypts it, and passes it as a string to Azure. Azure decrypts the encrypted hash and stores the password hash as a user attribute in Azure AD.



- This ensures a user signing on to Azure uses the same password as the on-premises domain, but doesn't require the additional infrastructure of a federated environment.
- It is important to understand that this is same sign-in, not single sign-on. The user still authenticates against two separate directory services, albeit with the same user name and password.
- 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.

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.

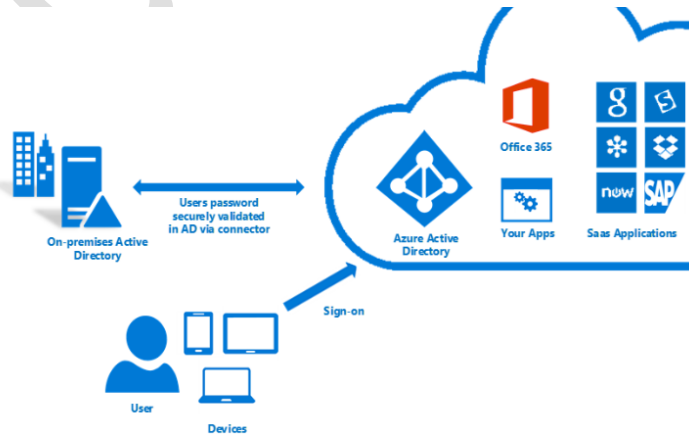
1. To configure and enable password writeback, sign in to your Azure AD Connect server and start the **Azure AD Connect** configuration wizard.
2. On the **Welcome** page, select **Configure**.
3. On the **Additional tasks** page, select **Customize synchronization options**, and then select **Next**.
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**.



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**.

Pass-Through Authentication (PTA):

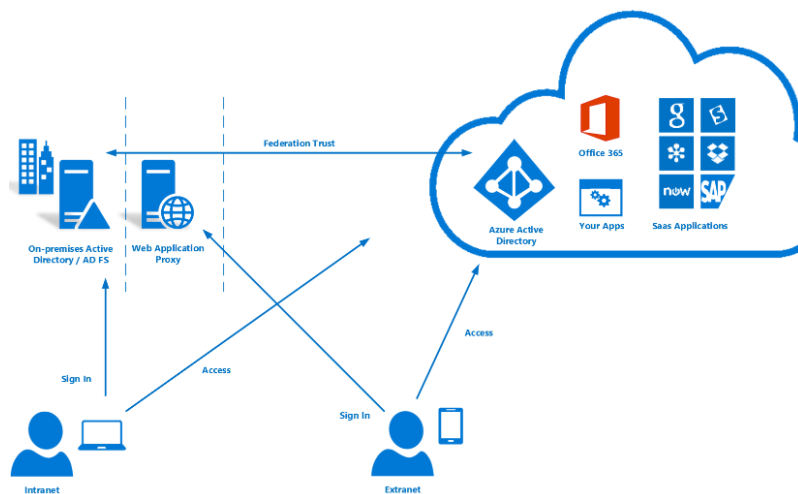
- The user's password is validated against the on-premises Active Directory controller. The password doesn't need to be present in Azure AD in any form.
- Sign-in usernames can be either the on-premises default username (userPrincipalName) or another attribute configured in Azure AD Connect (known as Alternate ID).



- 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.
- Integrated with cloud-based self-service password management, including password writeback to on-premises Active Directory and password protection by **banning commonly** used passwords.
- Pass-through authentication is not only for user sign-in but allows an organization to use other Azure AD features, such as password management, role-based access control, published applications, and conditional access policies.
- **Limitations:** Doesn't work for scenarios that need Azure AD Domain Services.

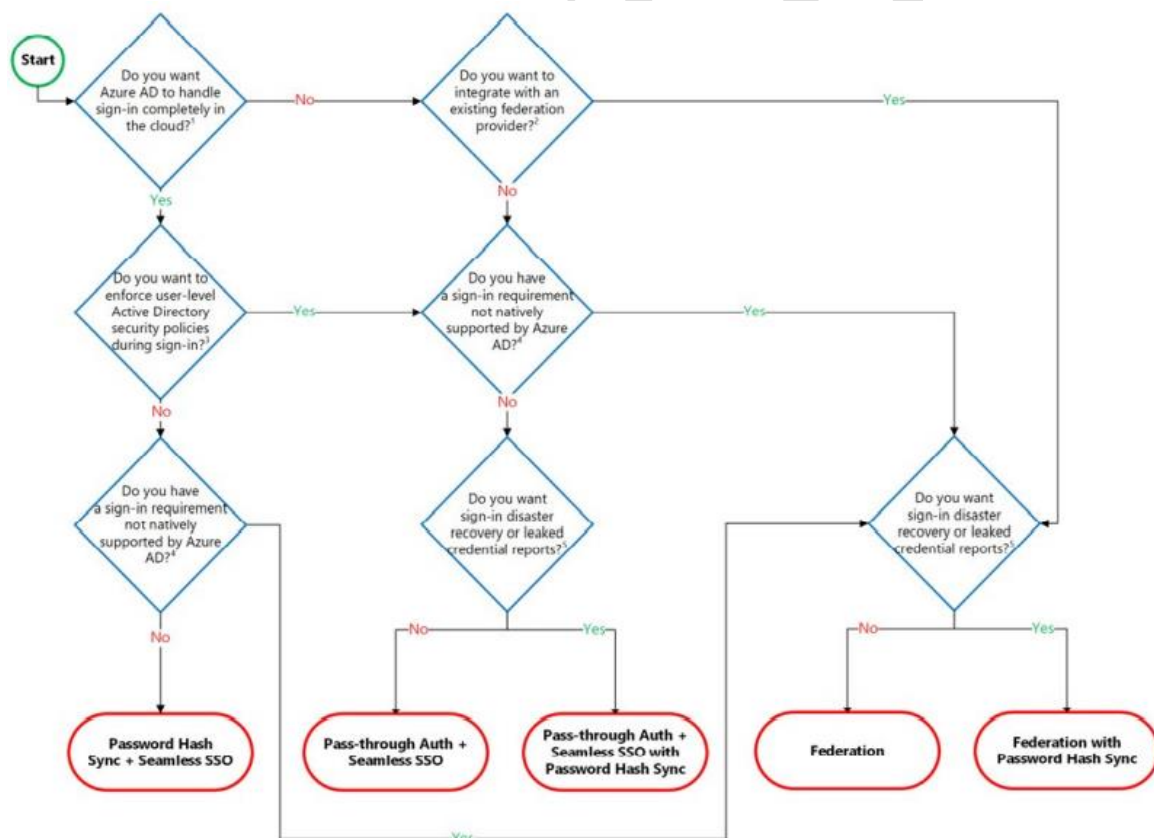
Federated SSO (with Active Directory Federation Services (AD FS)):

- Federation is a collection of domains that have established trust. A typical federation might include a number of organizations that have established trust for shared access to a set of resources.
- 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.
- This is mandatory for situations where User from Internet should be able to **access On-Premise Applications** using the Identity provided by Azure AD.
- It uses Claims based Authentication.



Azure AD Authentication Decision Tree

Summary of Sign-On Methods

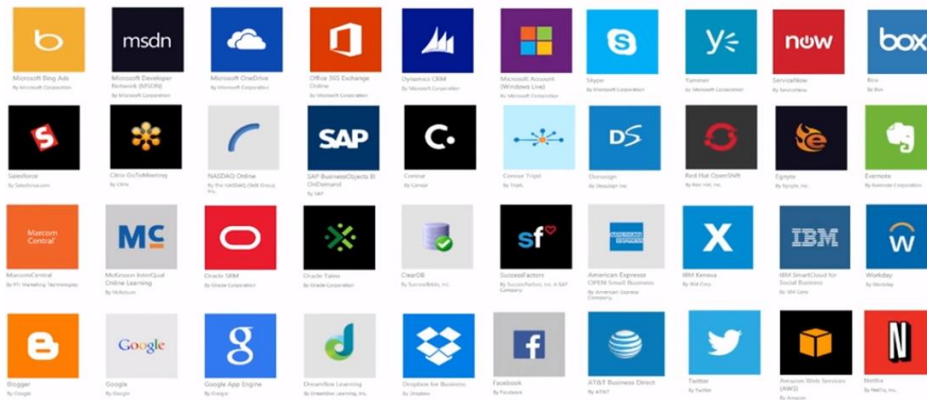


1. Do you need on-premises Active Directory integration? If the answer is No, then you would use Cloud-Only authentication.

2. If you do need on-premises Active Directory integration, then do you need to use cloud authentication, password protection, and your authentication requirements are natively supported by Azure AD? If the answer is Yes, Then you would use **Password Hash Sync + Seamless SSO**.
3. If you do need on-premises Active Directory integration, but you do not need to use cloud authentication, password protection, and your authentication requirements are natively supported by Azure AD, then you would use **Pass-through Authentication Seamless SSO**.
4. If you need on-premises Active Directory integration, have an existing federation provider and your authentication requirements are NOT natively supported by Azure AD, then you would use **Federation authentication**.

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>

- **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 Facebook SaaS Applications

1. Azure Portal → Azure Active Directory → **Enterprise Application**
2. **+ New Application** → click All
3. Categories: Social, Under Add from the gallery, Search for **Facebook** → **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

4. Azure Portal → Azure Active Directory → **Enterprise Application** → All applications → **Facebook** → Settings
5. **Facebook App** → **Single sign-on** → Mode = "Password-based Sign-on",
6. **Users and groups** → Add couple of users
7. **For atleast one user:** Select User Click **Update Credentials** Button on top → Enter Email Address/Password to use → Save.

Testing the configuration:

8. 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.
9. Click on Facebook and Login (Only required for the first time and only if admin has not provided the u/p).
10. Close the Facebook page

11. Click again on Facebook and note that this time it will not ask for login.

Integrating SAML Toolkit Application

<https://docs.microsoft.com/en-us/azure/active-directory/saas-apps/saml-toolkit-tutorial>

1. Azure Portal → Azure Active Directory → **Enterprise Application** → All applications → **SAML Toolkit Application** → Settings
2. **Users and groups**: Add users/groups who can access.
3. **SAML Toolkit Application** → **Single sign-on** → Mode = "SAML Based Sign-on",
4. Visit <https://sampletoolkit.azurewebsites.net> and follow the instructions.

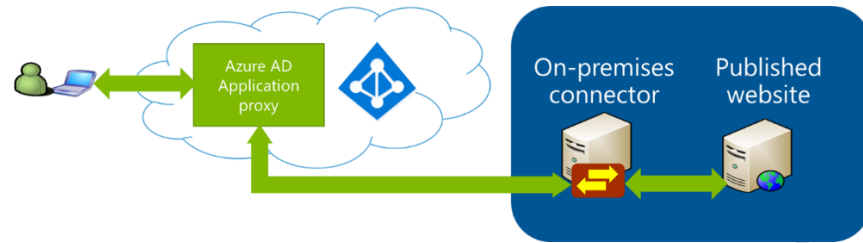
Note: Ensure that you Register on smltoolkit website using a user which exists in Azure AD.

About Self-service

- a) Allow users to request access to this application = Yes
 - b) To which group should assigned users be added = Facebook Guests
 - c) Require approval before granting access to this application? = Yes
 - d) Allow approvers to set user's passwords for this application? = yes
 - e) Who is allowed to approve access to this application? = <Primany domain users>
 - f) → ... → Save
5. Azure Portal → Azure Active Directory → User Settings → Users can add gallery apps to their Access Panel = Yes (for Self service)
 6. To Add App to Access Panel (Self-Service)
 - a) Access Panel → Click on UserName (top right) → Apps
 - Request access to app.
 - b) The approver will receive an email to approve or reject the request.

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, there 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.
- One Global admin role with verified domain name.
- Windows Server 2012 R2 or higher on the on-premises connector.

To Install Connector:

1. RDP to on premise machine on which you want to install the connector
2. Login to <https://portal.azure.com>
3. Azure AD → Application Proxy → + Download connector service → Accept terms and Download → Run the downloaded file.

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](#).

[Accept terms & Download](#)

Configure On-premises Application:

- Azure AD → Enterprise Application → + New Application → + Create your own application → Set What's the name of your app = "MyWebSite", Select Configure Application Proxy for secure remote access to an on-premises application

Name * ⓘ

Internal Url * ⓘ

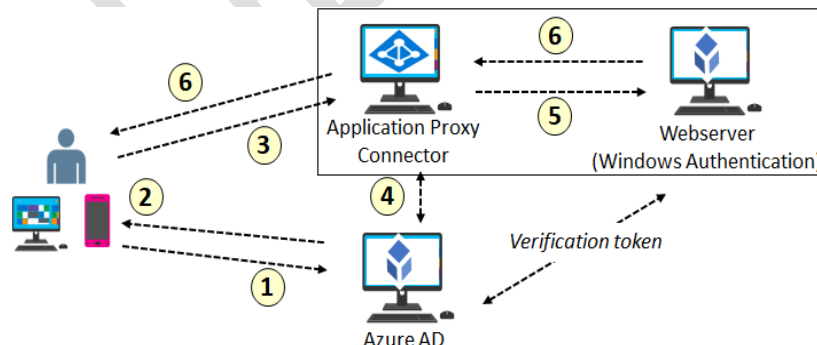
External Url ⓘ

Pre Authentication ⓘ

Connector Group ⓘ

- Azure AD → Enterprise Application → MyWebSite → Users & Groups → Add users who should be able to access the website.
- In browser Open the URL: <https://MyWebSite-mayazureorg.msappproxy.net>

How Does Application Proxy Authentication Process Works?



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.

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.

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 easily configured to use are:

- **Mobile Network: Phone Call.** A call is placed to the user's registered phone.
- **Mobile Network: Text Message to phone.** A six-digit code is sent to the user's cell phone.
- **Internet: 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.
- **Internet: 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 helps **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.

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.

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.

[Set up](#) 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".

[Next](#) [cancel](#)

MFA Licensing and Pricing

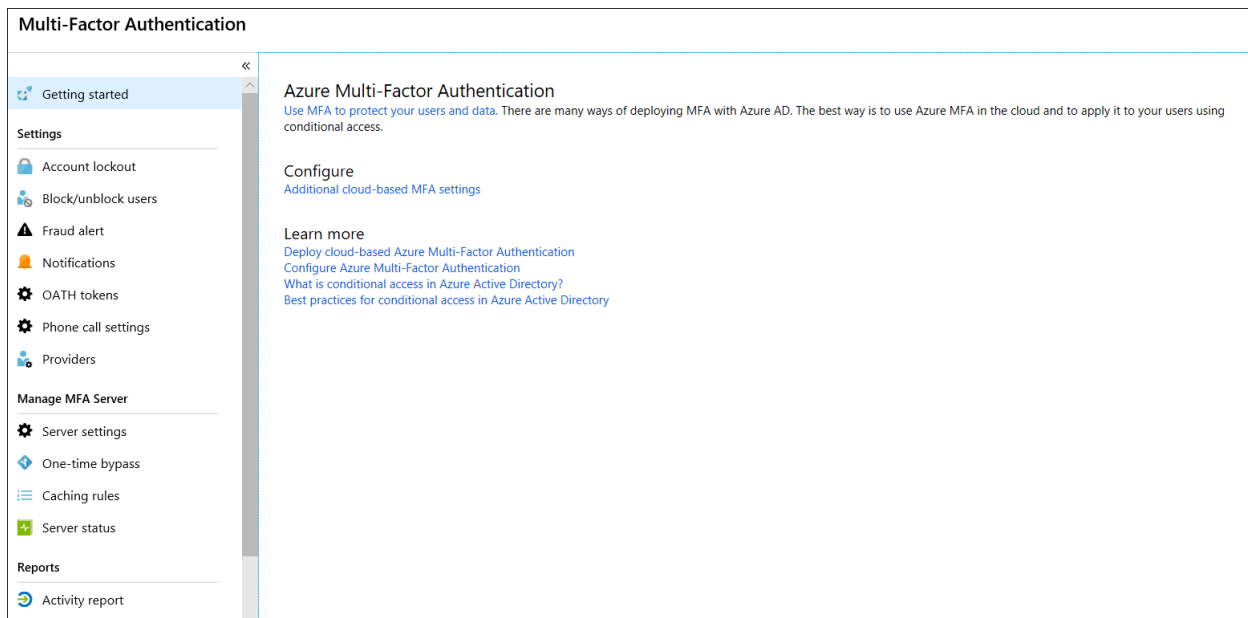
There are three pricing methods for Azure MFA.

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



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):

save

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

<input checked="" type="checkbox"/>	DISPLAY NAME ^	USER NAME	MULTI-FACTOR AUTH STATUS
<input checked="" type="checkbox"/>	admin	admin@sandeepsonideccansoft.onmicrosoft.com	Disabled
<input type="checkbox"/>	bhattajay	bhattajay@yahoo.com	Disabled
<input type="checkbox"/>	decnsoft	decnsoft@hotmail.com	Disabled
<input checked="" type="checkbox"/>	GAdmin	gadmin@sandeepsonideccansoft.onmicrosoft.com	Disabled
<input type="checkbox"/>	phanichand.t@deccansoft.cor	phanichand.t@deccansoft.com	Disabled
<input type="checkbox"/>	rahul	rahul@deccansoft.com	Disabled
<input checked="" type="checkbox"/>	RBAC Tutorial User	test123@sandeepsonideccansoft.onmicrosoft.com	Disabled
<input type="checkbox"/>	Sandeep Soni	sandeepsoni@deccansoft.net	Disabled
<input type="checkbox"/>	Sandeep Soni	sandeepsoni@deccansoft.com	Disabled
<input type="checkbox"/>	sansoni	sansoni@gmail.com	Disabled
<input type="checkbox"/>	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.

One-time bypass

Search (Ctrl+/)

MANAGE

- Account lockout
- Block/unblock users
- Caching rules
- Fraud alert
- Notifications
- One-time bypass**
- Phone call settings

+ Add Save Discard

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

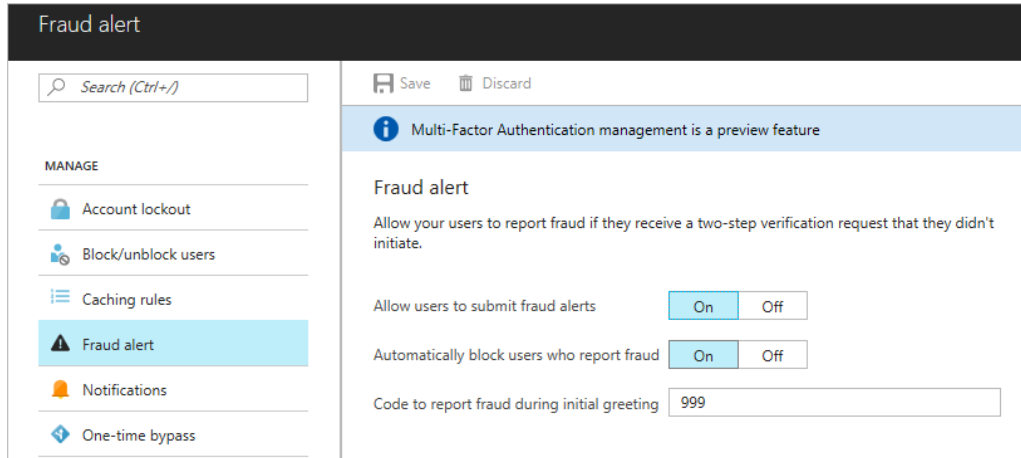
300

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.



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.

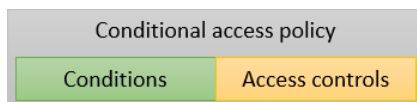
Code to report fraud during initial greeting. When users receive a phone call to perform two-step verification, they normally press # to confirm their sign-in. To report fraud, the user enters a code before pressing #. This code is 0 by default, but you can customize it.

Revoke MFA sessions

Azure Active Directory → Users → Select a User → Authentication methods → Click Remove MFA Sessions
This is let the machine where MFA is remembered to forget.

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.



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**, Name your policy **MFA Pilot**
4. 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**
5. Under **Cloud apps**, select the **Select apps** radio button
 - The cloud app for the Azure portal is **Microsoft Azure Management**
 - Click **Select**
 - Click **Done**
6. Skip the **Conditions** section
7. Under **Grant**, make sure the **Grant access** radio button is selected
 - Check the box for **Require multi-factor authentication**
 - Click **Select**
8. Skip the **Session** section
9. Set the **Enable policy** toggle to **On**

10. 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://myapps.microsoft.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.

Identity Protection is a tool that allows organizations to accomplish three key tasks:

- Automate the detection and remediation of identity-based risks.
- Investigate risks using data in the portal.
- Export risk detection data to third-party utilities for further analysis.

Identity Protection uses the learnings Microsoft has acquired from their position in organizations with Azure AD, the consumer space with Microsoft Accounts, and in gaming with Xbox to protect your users. Microsoft analyses **6.5 trillion signals** per day to identify and protect customers from threats.

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.






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.

Azure Portal → Azure Active Directory → Security → Identity Protection

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** - When cybercriminals compromise valid passwords of legitimate users, they often share those credentials. – **HIGH Risk Level**
2. **Sign-ins from anonymous IP addresses** - This risk detection type identifies users who have successfully signed in from an IP address that has been identified as an anonymous proxy IP address. – **Medium Risk Level**
3. **Impossible travel to atypical locations** - This risk detection type identifies two sign-ins originating from geographically distant locations, where at least one of the locations may also be atypical for the user, given past behavior. - **Medium Risk Level**
4. **Sign-ins from infected devices** - This risk detection type identifies sign-ins from devices infected with malware, that are known to actively communicate with a bot server. - **Medium Risk Level**

5. **Sign-in from unfamiliar locations** - This risk detection type considers past sign-in locations (IP, Latitude / Longitude and ASN) to determine new / unfamiliar locations. - Medium Risk Level
6. **Sign-ins from IP addresses with suspicious activity** - This risk detection type identifies IP addresses from which a high number of failed sign-in attempts were seen, across multiple user accounts, over a short period of time - Low Risk Level

Identity Protection policies

Azure Active Directory Identity Protection includes three default policies that administrators can choose to enable.

All the policies allow for excluding users such as your emergency access or break-glass administrator accounts.

1. Multi-factor authentication registration policy
2. User risk mediation policy
3. Sign-in risk mediation policy

<p>Policy name Multi-factor authentication registration policy</p> <p>Assignments</p> <p>Users ⓘ All users ></p> <p>Controls</p> <p>Access ⓘ Require Azure MFA registration ></p> <p>i MFA Registration Policy only affects cloud-based Azure MFA. If you have MFA Server it will not be affected.</p> <p>Enforce Policy <input checked="" type="checkbox"/> On <input type="checkbox"/> Off</p>	<p>Policy name User risk remediation policy</p> <p>Assignments</p> <p>Users ⓘ All users ></p> <p>Conditions ⓘ User risk ></p> <p>Controls</p> <p>Access ⓘ Require password change ></p> <p>Review</p> <p>Estimated impact ⓘ Number of users impacted ></p> <p>Enforce Policy <input checked="" type="checkbox"/> On <input type="checkbox"/> Off</p>	<p>Policy name Sign-in risk remediation policy</p> <p>Assignments</p> <p>Users ⓘ All users ></p> <p>Conditions ⓘ Sign-in risk ></p> <p>Controls</p> <p>Access ⓘ Require multi-factor authentication ></p> <p>Review</p> <p>Estimated impact ⓘ Number of sign-ins impacted ></p> <p>Enforce Policy <input checked="" type="checkbox"/> On <input type="checkbox"/> Off</p>
---	--	---

Azure MFA registration policy

Identity Protection can help organizations roll out Azure Multi-Factor Authentication (MFA) using a Conditional Access policy requiring registration at sign-in. Enabling this policy is a great way to ensure new users in your organization have registered for MFA on their first day. Multi-factor authentication is one of the self-remediation

methods for risk events within Identity Protection. Self-remediation allows your users to act on their own to reduce helpdesk call volume.

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
☒ On ☐ Off

Save

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:

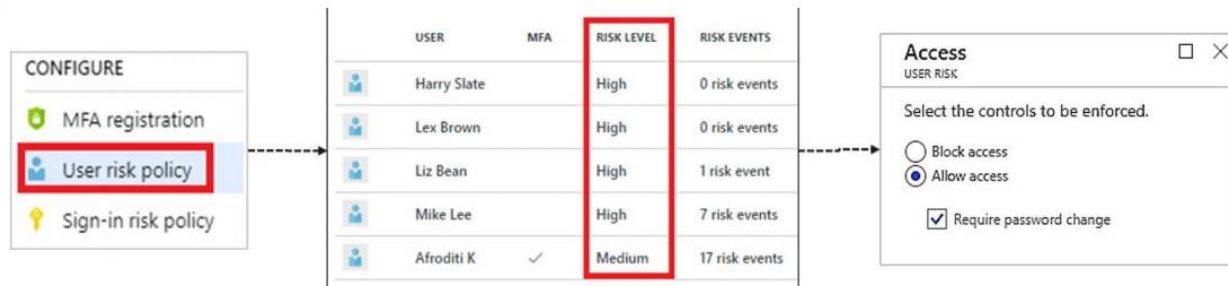
1. Select users → Select Test User → Select → Done
2. Conditions → Sign-in risk level = Medium and above → Select → Done.
3. Controls → Select Allow access → check Require multi-factor authentication

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. Enforce 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.

User risk policy

Identity Protection can calculate what it believes is normal for a user's behavior and use that to base decisions for their risk. User risk is a calculation of probability that an identity has been compromised. Administrators can decide based on this risk score signal to enforce organizational requirements. Administrators can choose to block access, allow access, or allow access but require a password change using Azure AD self-service password reset.



The above image shows the configuration of User Risk Policy applied

- To user sign-ins
- Automatically respond based on a specific user's risk level
- Provide the condition (risk level) and action (block or allow)
- Use a high threshold during policy roll out
- Use a low threshold for greater security

With the information provided by the risky users report, administrators can find:

- Which users are at risk, have had risk remediated, or have had risk dismissed?
- Details about detections
- History of all risky sign-ins

- Risk history Administrators can then choose to act on these events.

Administrators can choose to:

- Reset the user password
- Confirm user compromise
- Dismiss user risk
- Block user from signing in
- Investigate further using Azure ATP

Access Reviews

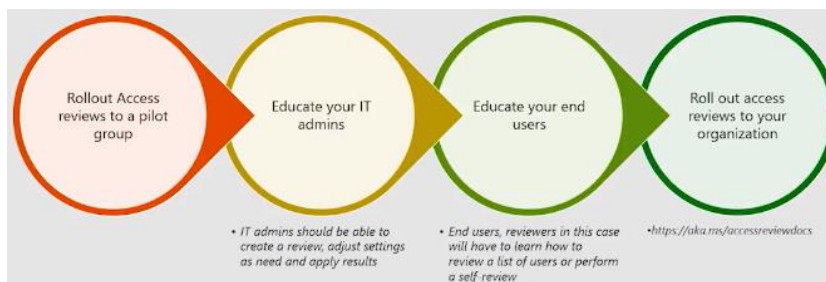
Azure Active Directory (Azure AD) access reviews enable organizations to efficiently manage **group memberships, access to enterprise applications, and role assignments.**

User's access can be reviewed on a regular basis to make sure only the right people have continued access.

Why are access reviews important

Azure AD enables you to collaborate internally within your organization and with users from external organizations, such as partners. Users can join groups, invite guests, connect to cloud apps, and work remotely from their work or personal devices. The convenience of leveraging the power of self-service has led to a need for better access management capabilities.

- As new employees join, how do you ensure they have the right access to be productive?
- As people move teams or leave the company, how do you ensure their old access is removed, especially when it involves guests?
- Excessive access rights can lead to audit findings and compromises as they indicate a lack of control over access.
- You have to proactively engage with resource owners to ensure they regularly review who has access to their resources.



Why to use Access Reviews?

- Too many users in privileged roles.
- When automation is infeasible.
- When a group is used for a new purpose.
- Business critical data access.
- Ask group owners to confirm they still need guests in their groups.
- Have reviews recur periodically.

Prerequisites

- Azure AD Premium P2
- Global administrator or User administrator

Create one or more access reviews

1. Azure Active Directory → **Identity Governance** → Create an access review
2. <https://docs.microsoft.com/en-us/azure/active-directory/governance/create-access-review>

Configure an access review

1. In the Portal, search for and select Identity Governance.
2. Under Access Reviews select Access Reviews.
3. Click New Access Review.
4. We will create an access review to ensure the validate the AZ500Admin group membership.
5. Complete the required information and discuss each setting. Configuration settings are added as you make your selections. For example, if you select a weekly access review, you will be prompted for the duration.
 - Review name: AZ500Review
 - Start date: current date
 - Frequency: One-time
 - Users to review: Members of a group
 - Scope: Everyone
 - Select a group: AZ500Admins
 - Reviewers: Selected user
 - Select reviewers: *add yourself as a reviewer

- Review the Upon completion settings, specifically the action if a reviewer doesn't respond.
 - Review Advanced settings.
6. Start the access review.
 7. On the Access review page ensure the new access review is listed.
 8. The Status will change from Not started to Initializing.

Conduct an access review

Review access to groups and applications in Azure AD access reviews

You can start the Access Review process from the **notification email** or **by going directly to the site**.

If you don't have the email, Sign in to the My Apps portal at <https://myapps.microsoft.com> → Click on User Icon → Access Reviews tile lists all pending reviews

<https://docs.microsoft.com/en-us/azure/active-directory/governance/perform-access-review>

1. When the access review is scheduled you will receive an email. This is the email associated with your reviewer account.
2. View the email and discuss the review instructions. Note when the review period will end.
3. In the email, click **Start review**.

Reminder: Please review users' access to the Managers in the Deccansoft5

User1, your organization requested that you approve or deny continued access for one or more users to the **Managers** group in the **Review Managers** review. The review period will end on **November 10, 2020**.

Start review >

Access packages
Request history
Approvals
Access reviews

← Access reviews

Review Managers

Please review user members of 'Managers' [See details](#)

✓ Approve ✗ Deny ? Don't know ↺ Reset decisions ⚙ Accept recommendations

Name ↑	Recommendation	Decision	Reviewed by
<input type="radio"/> Admin admin@deccansoft5.onmicrosoft.com	Deny Last signed in more than 30 days ago		Details
<input type="radio"/> User2 user2@deccansoft5.onmicrosoft.com	Deny Last signed in more than 30 days ago		Details

4. On the Access reviews page, click the AZ500Review.
5. Notice you are reviewing the AZ500Admin group members. There are two members.

6. Use the Details link to view information about the user.
7. Select **Approve** for one user and **Deny** for the other. Be sure to provide a Reason.
8. Submit your reviews.

Review the access review results

1. Click the AZ500Review
2. From the Overview blade review the results.
3. There should be one member approved and one member denied.
4. Click Results for more detailed information about the reviewer and their reasons.
5. From the Overview blade, click Stop and confirm you want to stop the review.
6. The Review status should now be Complete

Privileged Identity Management (PIM)

It is a service that enables you to manage, control, and monitor access to important resources in your organization.

What does it do?

- Provide **just-in-time** privileged access to Azure AD and Azure resources
- Assign **time-bound** access to resources using start and end dates
- Require **approval** to activate privileged roles
- Enforce **multi-factor authentication** to activate any role
- Use **justification** to understand why users are activate
- Get **notifications** when privileged roles are activated
- Conduct **access reviews** to ensure users still need roles
- Download **audit history** for internal or external audit

To use Privileged Identity Management, your directory must have one of the following paid or trial licenses:

- Azure AD Premium P2
- OR
- Enterprise Mobility + Security (EMS) E5

High-level overview of how Privileged Identity Management works

1. Privileged Identity Management is set up so that users are eligible for privileged roles.

2. When an eligible user needs to use their privileged role, they activate the role in Privileged Identity Management.
3. Depending on the Privileged Identity Management settings configured for the role, the user must complete certain steps (such as performing multi-factor authentication, getting approval, or specifying a reason.)
4. Once the user successfully activates their role, they will get the role for a pre-configured time period.
5. Administrators can view a history of all Privileged Identity Management activities in the audit log. They can also further secure their Azure AD organizations and meet compliance using Privileged Identity Management features like access reviews and alerts.



Prepare PIM for Azure AD roles

Once you have enabled Privileged Identity Management for your directory, you can prepare Privileged Identity Management to manage Azure AD roles.

Here are the tasks we recommend for you to prepare for Azure AD roles, in order:

1. Configure Azure AD role settings.

1. Sign in to the [Azure portal](#) with a user who is in the [Privileged role administrator](#) role
2. Azure AD Privileged Identity Management → **Azure AD Roles** → **Settings**
3. Select the role Application Administrator → Edit → Change settings as per requirement (Enable Azure MFA) → Update

2. Give eligible assignments.

1. Azure AD Privileged Identity Management → Azure AD Roles → **Roles** → **+ Add assignments**
2. Under Membership Tab: Select role = Application Administrator, Select members (couple of users from Azure AD)
3. Under Steetings Tab: Assignment Type = **Eligible** → Click on Assign

3. Allow eligible users to activate their Azure AD role just-in-time.

1. Azure AD Privileged Identity Management → Azure AD Roles → **My Roles**
2. Select Role → Activate → Under Eligible assignment tab → Select Role → **Activate**
3. Set Duration and Reason → Activate
4. Verify: Switch to Active assignments

Prepare PIM for Azure roles

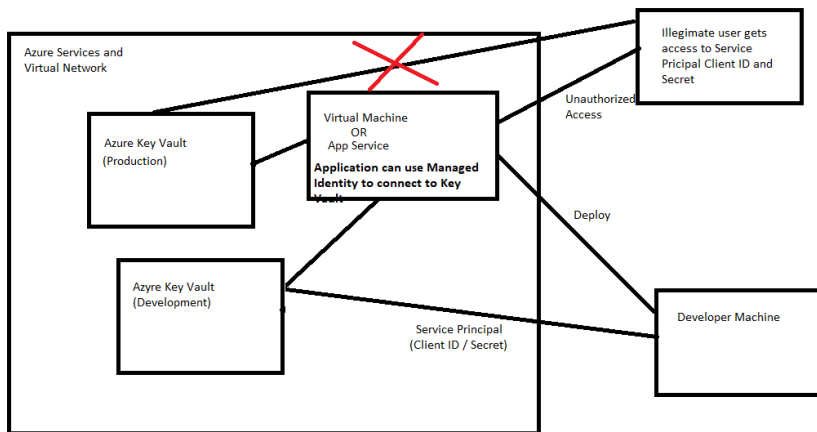
Once you have enabled Privileged Identity Management for your directory, you can prepare Privileged Identity Management to manage Azure roles for Azure resource access on a subscription.

Here are the tasks we recommend for you to prepare for Azure roles, in order:

1. [Discover Azure resources](#)
2. [Configure Azure role settings.](#)
3. [Give eligible assignments.](#)
4. [Allow eligible users to activate their Azure roles just-in-time.](#)

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:

1. 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.

2. 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
If Windows VM is used.

```
$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
```

*Note: IP of request should be same as provided in URL. The mention is IP is NOT IP Address of VM.
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.KeyVault;
using Microsoft.Azure.Services.AppAuthentication;
using System;
using System.Configuration;
using System.Threading.Tasks;

class Program
{
    static async Task Main(string[] args)
    {
        var azureServiceTokenProvider = new AzureServiceTokenProvider();
        var kv = new KeyVaultClient(new
KeyVaultClient.AuthenticationCallback(azureServiceTokenProvider.KeyVaultTokenCallback));
        var sec = await kv.GetSecretAsync(ConfigurationManager.AppSettings["S1"]);
        Console.WriteLine(sec.Value);
    }
}
```

//Service to Service Authentication

Following code in App Service can access Storage Account without explicitly having reference to Storage Account Key

Before this enable Managed Identity on App Service.

```
var azureServiceTokenProvider = new AzureServiceTokenProvider();
string accessToken = await azureServiceTokenProvider.GetAccessTokenAsync("https://storage.azure.com/");
```

```
StorageCredentials creds = new StorageCredentials(accessToken)
CloudStorageAccount account = new CloudStorageAccount(creds, useHttps: true);
CloudBlobClient client = account.CreateCloudBlobClient();
```

DECCANSOFT