Posts

Azure AD Connect Lab Setup – Step by Step Guide

June 22, 2020 asingh Leave a comment

Most of the enterprise adopting cloud has a mix of on-premises and cloud based infrastructure and in such scenarios having simplified identity control becomes key factor where you can use your existing identities to control authentication and authorization across all the applications and services regardless of cloud or on-premises. Talking from Microsoft Azure's perspective, with help of Azure Active Directory and Azure AD Connect we can implement "**Hybrid Identity**" solution which will simplify authentication and authorization to all applications and services across cloud and on-premises.

AWS What's New Tweets

Tweets by @awswhatsnew

What's New @awswhatsne

Amazon Machine Image copy limits increased to 100 images per destination Region

Amazon EC2 now allows you to copy up to 100 Amazon Machine Images (AMIs) concurrently per destination region per account, an increase from the previous limit of 50 conc... aws.amazon.com/about-aws/what…

Azure AD Connect acts as bridge between your on-premises Active Directory infrastructure and Azure AD, it synchronizes user accounts, group memberships, and credential hashes from an on-premises Active Directory to Azure AD. We would be able to utilize many good feature of Azure AD like Single-Sing-On (SSO)/Federation, MFA, Hybrid Azure AD join, access control on Azure resources and Office 365 using on-premises AD identities

In this guide, I will walk you through how to configure Azure AD Connect to synchronize on-premises AD identities with Azure AD, there are different types of Azure AD Connect deployment topologies in the scenario of multiple forest and multiple Azure AD Tenants. In this lab implementation guide, Azure AD Connect deployment topology is “Single forest, single Azure AD tenant”.

There are various prerequisites to note before we can go ahead and install Azure AD Connect in our environment, please refer below list of requirements for Azure AD, on-premises AD and Azure AD Connect server:

Azure AD prerequisites:

1. An Azure AD tenant. You get one with an Azure free trial also.

2. Add and verify the domain you plan to use in Azure AD. This should be your publicly registered domain. For example, if you plan to

use contoso.com for your users then make sure this domain has been verified and you are not only using the contoso.onmicrosoft.com default domain. Every new Azure AD tenant comes with an initial domain name, <domainname>.onmicrosoft.com. You can't change or delete the initial domain name, but you can add your organization's names. Adding custom domain names helps you to create user names that are familiar to your users, such as alain@contoso.com.

In my lab, I have my a routable domain (cloudmegh.in) and my on-premises AD domain name is also same but in cause If your on-premises AD domain is non-routable domain then you can follow this Microsoft documentation to solve the non-routable domain problem by registering new UPN suffix or suffixes in AD DS to match the domain (or domains) you verified in Microsoft 365/Azure AD. After you register the new suffix, you update the user UPNs to replace the .local with the new domain name for example so that a user account looks like abc@contoso.com.
https://docs.microsoft.com/en-us/office365/enterprise/prepare-a-non-routable-domain-for-directory-synchronization

On-premises Active Directory prerequisites:
1. Use IdFix to identify errors such as duplicates and formatting problems in your directory before you synchronize to Azure AD and Office 365.
https://docs.microsoft.com/en-

Get the details: msft.it/6013pXRd1

19 Dec 2020

Microsoft Az
@Azure
Join the #WindowsVirtualDesktop digital event on January 28! Get best practices to optimize virtual desktops and apps—and ask your questions in the live chat. Register now: msft.it/6014pXBUY

Embed View on Twitter

Cloud Native Tweets

Tweets by @CloudNativeFdn

CNCF Retweeted

Julien Pivotto
@roidelapluie
Gathering uses cases / ideas /

us/office365/enterprise/install-and-run-idfix

2. It is recommended to enable the Active Directory recycle bin.

3. The AD schema version and forest functional level must be Windows Server 2003 or later. The domain controllers can run any version as long as the schema and forest level requirements are met.

4. If you plan to use the feature password writeback, then the Domain Controllers must be on Windows Server 2008 R2 or later.

5. The domain controller used by Azure AD must be writable. It is not supported to use a RODC (read-only domain controller) and Azure AD Connect does not follow any write redirects.

Azure AD Connect server prerequisites:
1. Azure AD Connect can only be installed on Windows Server Standard, Enterprise or Datacenter editions.
2. Azure AD Connect must be installed on Windows Server 2012 or later. This server must be domain joined and may be a domain controller or a member server.

SQL for Azure AD Connect:
1. Azure AD Connect requires a SQL Server database to store identity data. By default a SQL Server 2012 Express LocalDB (a light version of SQL Server Express) is installed. SQL Server Express has a 10GB size limit that enables you to

manage approximately 100,000 objects. If you need to manage a higher volume of directory objects, you need to point the installation wizard to a different installation of SQL Server.

2. Microsoft Azure SQL Database is not supported as a database.

Accounts:

1. An Azure AD Global Administrator account for the Azure AD tenant you wish to integrate with. This account must be a school or organization account and cannot be a Microsoft account.

2. If you use express settings or upgrade from DirSync, then you must have an Enterprise Administrator account for your on-premises Active Directory.

Network Connectivity:

If your local network has firewall/proxy then you need to ensure that all the required ports and endpoints mentioned in below documentations.

https://docs.microsoft.com/en-us/azure/active-directory/hybrid/reference-connect-ports

https://docs.microsoft.com/en-us/office365/enterprise/urls-and-ip-address-ranges?redirectSourcePath=%252fen-gb%252farticle%252foffice-365-urls-and-ip-address-ranges-8548a211-3fe7-47cb-abb1-355ea5aa88a2

Azure AD Connect – Key Terminologies and Components:

Azure AD Connect is a vast solution in itself so it's not feasible to cover all the deep dive architecture details in this post, below are

important AAD Connect terminologies and concepts that you need to understand when working with Azure AD Connect.

1. Azure AD Connect sync (sync engine)
2. Connector
3. Connected Data Sources or Connected Directories (CD)
4. Source anchor
5. Connector Space (CS)
6. Metaverse (MV)
7. Joined Object (or connector object)
8. Disjoined Object (or disconnector object)
9. Provisioning
10. Deprovisioning

It will be good to refer below Microsoft documentations to dive deeper into Azure AD Connect architecture and above concepts.

Azure AD Connect sync: Technical Concepts:
https://docs.microsoft.com/en-us/azure/active-directory/hybrid/plan-connect-design-concepts

Azure AD Connect sync: Understanding the architecture:
https://docs.microsoft.com/en-us/azure/active-directory/hybrid/concept-azure-ad-connect-sync-architecture

Azure AD Connect Authentication (sign-in) Options:
Below are the four different authentication (sign-in) mechanisms provided by Azure AD when you are using Azure AD Connect, based on your

Today

December 2020

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

« Jun

Categories

Uncategorized (30)

feasibility from security and compliance perspective you can choose the one appropriate. During Azure AD Connect installation wizard you will have the ability to choose one of the authentication mechanism.

1. Password Hash Synchronization (PHS):

–When we install Azure AD Connect with "Express Settings" then Password Harsh Synchronization (PHS) authentication mechanism is the default configuration.

–AAD Connect synchronizes a hash, of the hash, of an AD user's password from an on-premises AD to Azure AD.

–To synchronize user's password, Azure AD Connect sync extracts user's password hash from the on-premises Active Directory. Extra security processing is applied to the password hash before it is synchronized to the Azure Active Directory.

–PHS process runs every 2 minutes and we cannot modify the frequency of this process.

2. Pass-through Authentication (PTA):

–Users credentials are validated by on-premises Active Directory Domain Controller via AAD Connect Authentication Agent, On-premises AD user's passwords are not stored in Azure AD in any form.

–For Pass-through Authentication to work, users need to be provisioned into Azure AD from on-premises Active Directory using Azure AD Connect. Pass-through Authentication does not apply to cloud-only users.

–Communication between Authentication Agent

and Azure AD is uses certificate-based authentication. These certificates are automatically renewed every few months by Azure AD.

–Microsoft recommends to have more than one AAD Connect Authentication Agent to provide high availability of authentication requests.

–PTA can also be used in conjunction with PHS for high availability scenarios, As per Microsoft *"Enabling Password Hash Synchronization gives you the option to failover authentication if your on-premises infrastructure is disrupted. This failover from Pass-through Authentication to Password Hash Synchronization is not automatic. You'll need to switch the sign-in method manually using Azure AD Connect. If the server running Azure AD Connect goes down, you'll require help from Microsoft Support to turn off Pass-through Authentication."*

3. Federation with ADFS:

–In ADFS federation scenario, Azure AD will be redirecting authentication request to ADFS.

4. Federation with PingFederate:

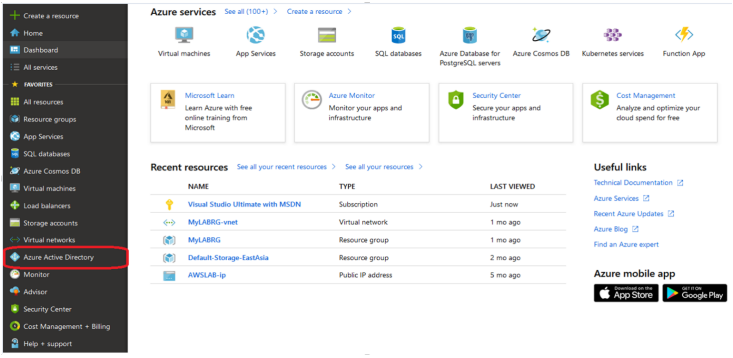
–If you are already using PingFederate in your environment then you may choose this method for authentication. AAD Connect natively supports PingFederate, please refer below official document from PingFederate regarding implementation of this.
https://support.pingidentity.com/s/article/PingFederate-Microsoft-Azure-End-to-End-Integration

Azure AD Connect Server Installation:

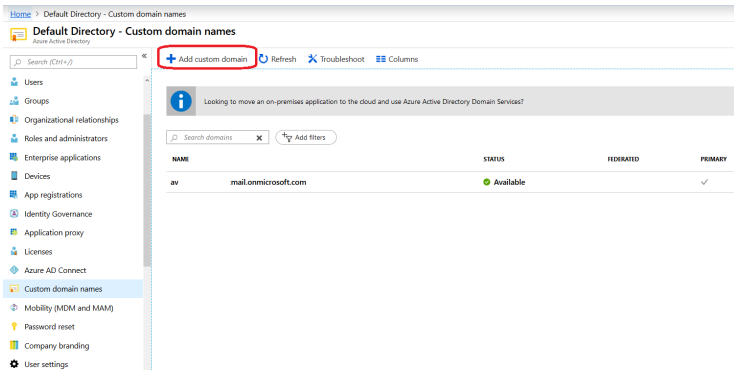
I have talked about some important concepts theoretically, Now let's go ahead and install the Azure AD Connect server in on-premises ADDS environment.

1. Add Custom Domain (Routable) to Azure AD and make it as a "Primary Domain" for Azure AD:

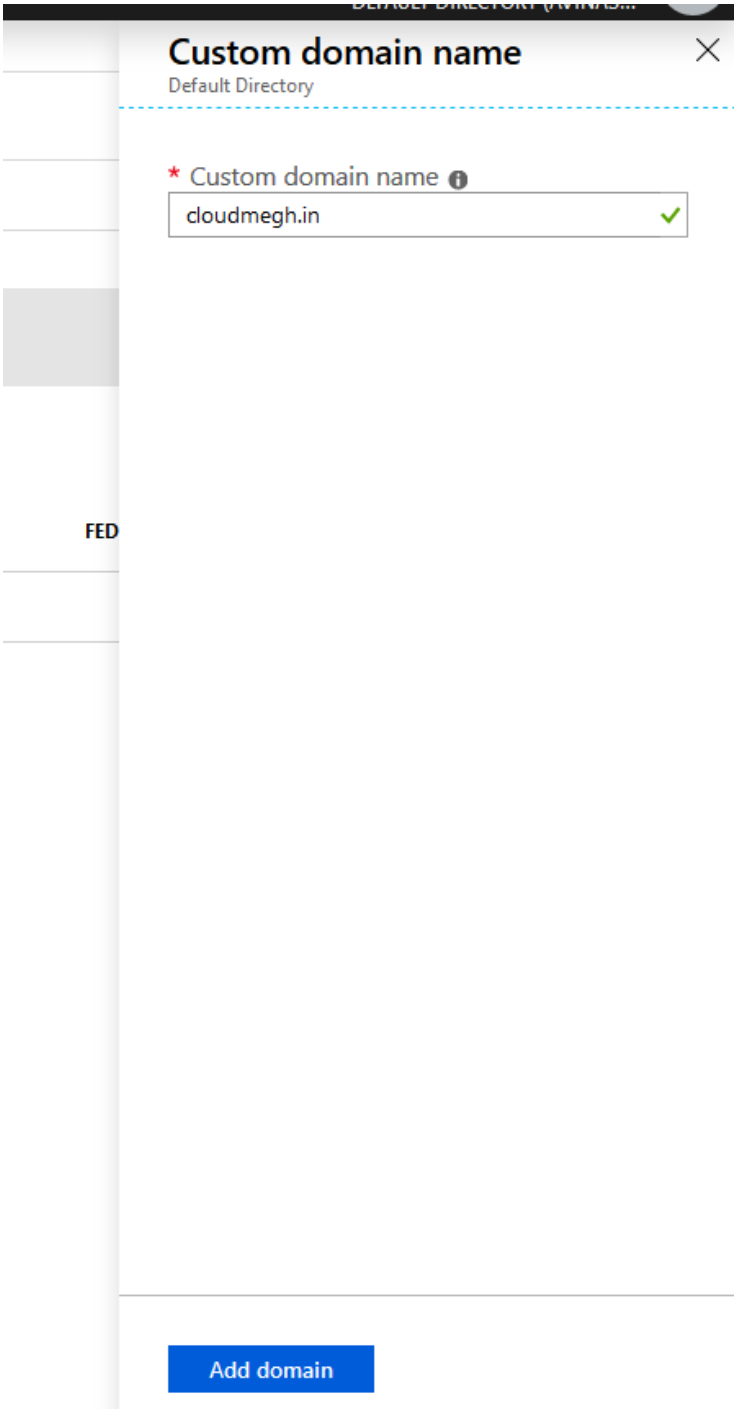
–Login to Azure portal and go to Azure Active Directory



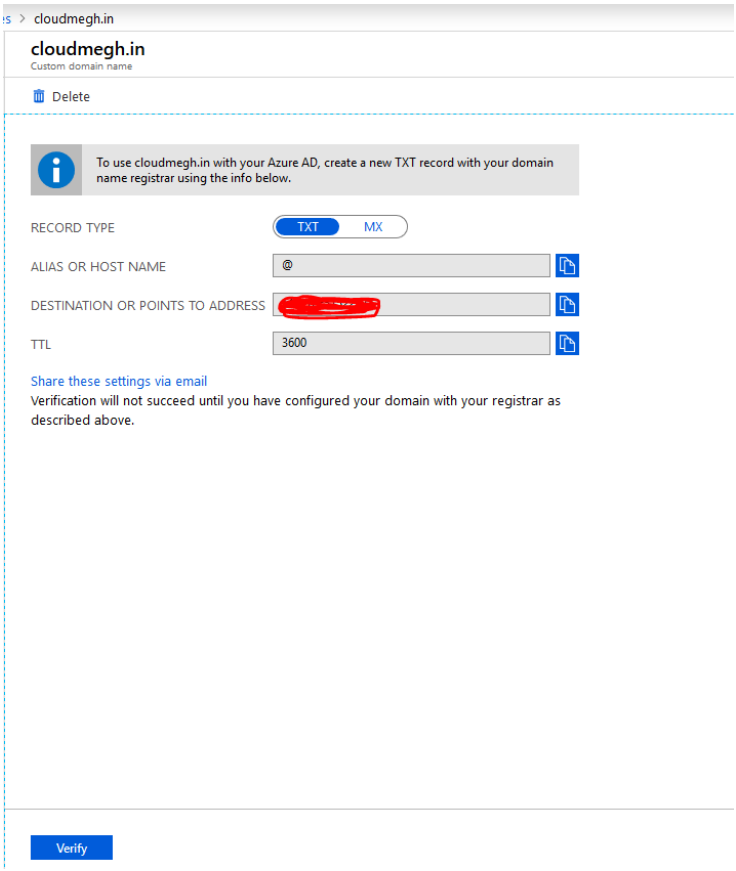
–Click on **"Add custom domain"** option.



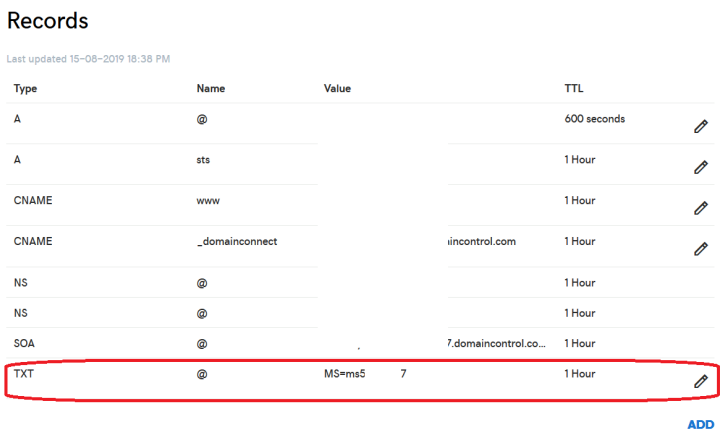
–I'm adding my routable domain "cloudmegh.in" to Azure AD, Click on "Add domain".

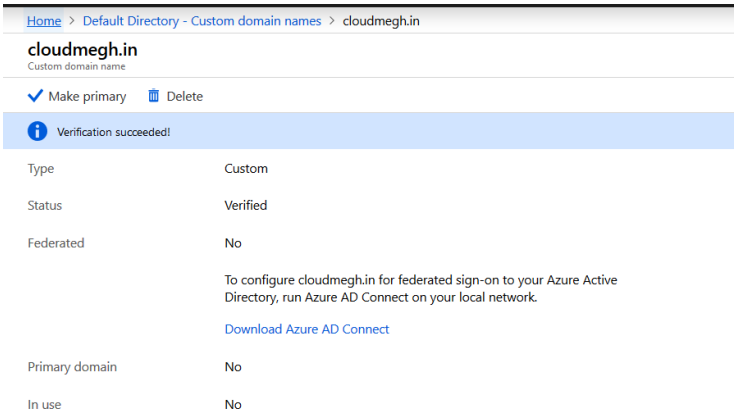


–Once you add your custom routable domain to Azure AD, it needs to be verified. To verify the domain you need to create TXT record in your domain registrar with below details.

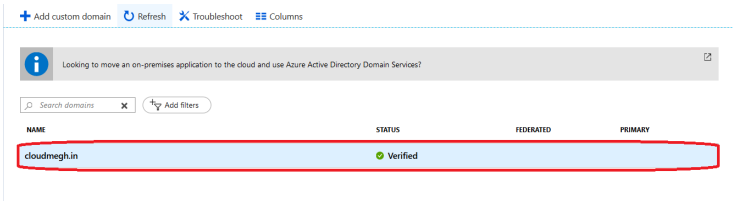


–My above domain is registered with Godaddy, so I logged into Godaddy console and add TXT record as below. Then I click on “Verify” button above in Azure AD console. In few minutes it will verify your domain information.

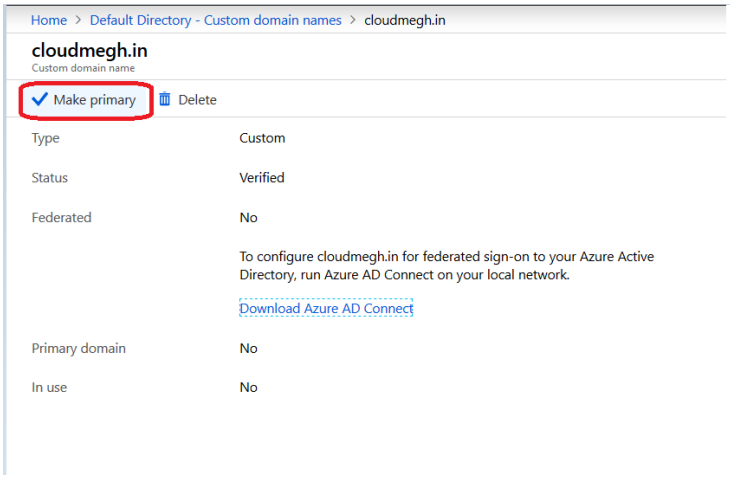


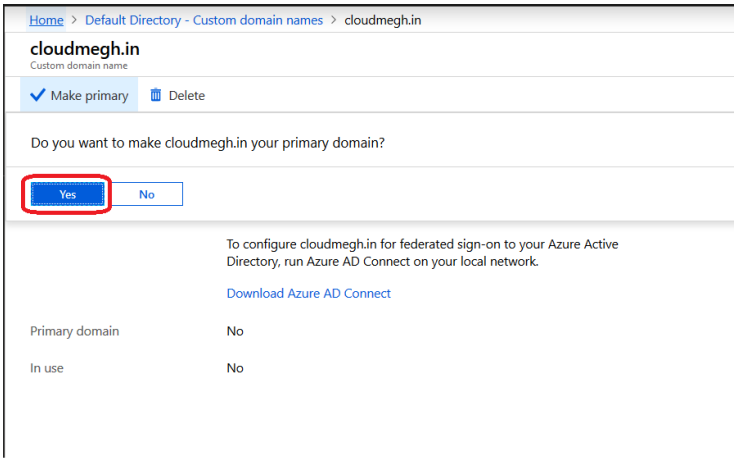


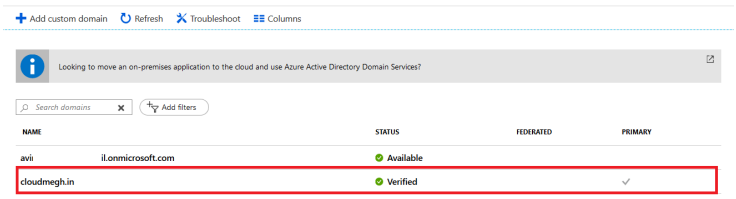
–Now my domain is showing as “Verified” in Azure AD console.



–Now I will set my custom domain as “Primary” for the directory, The primary domain is the default domain name for a new user when you create a new user. Setting a primary domain name streamlines the process for an administrator to create new users in the portal.

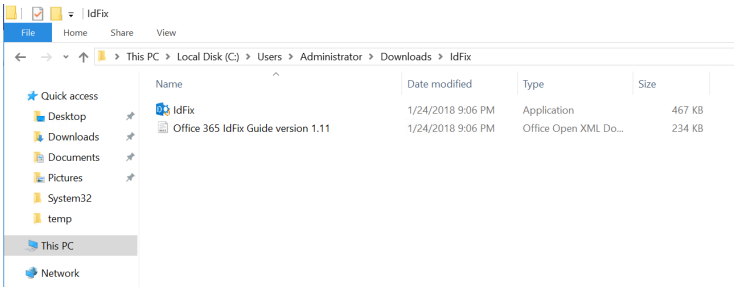


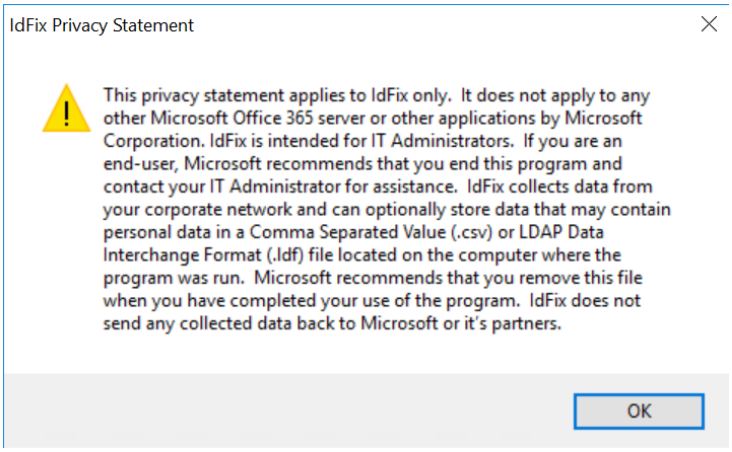


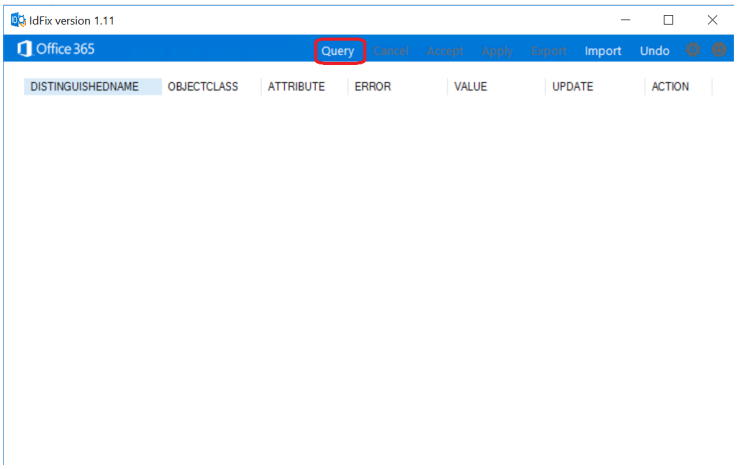


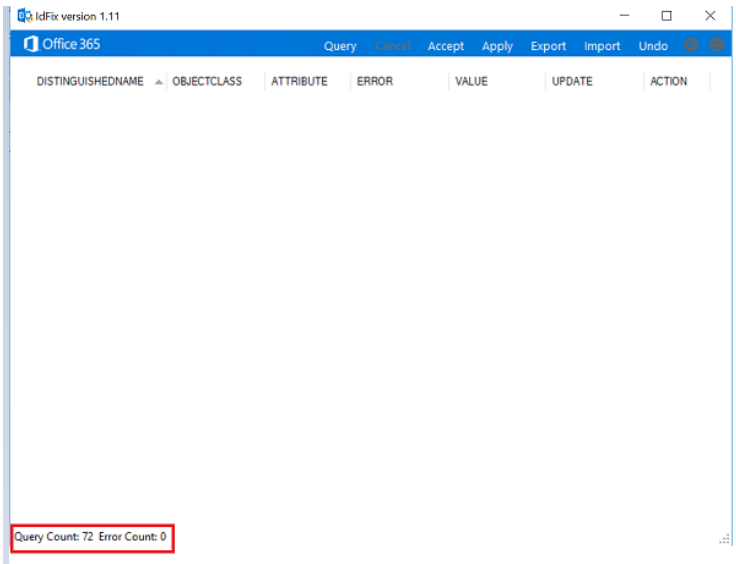
2. Download and run the Office 365 IdFix tool:

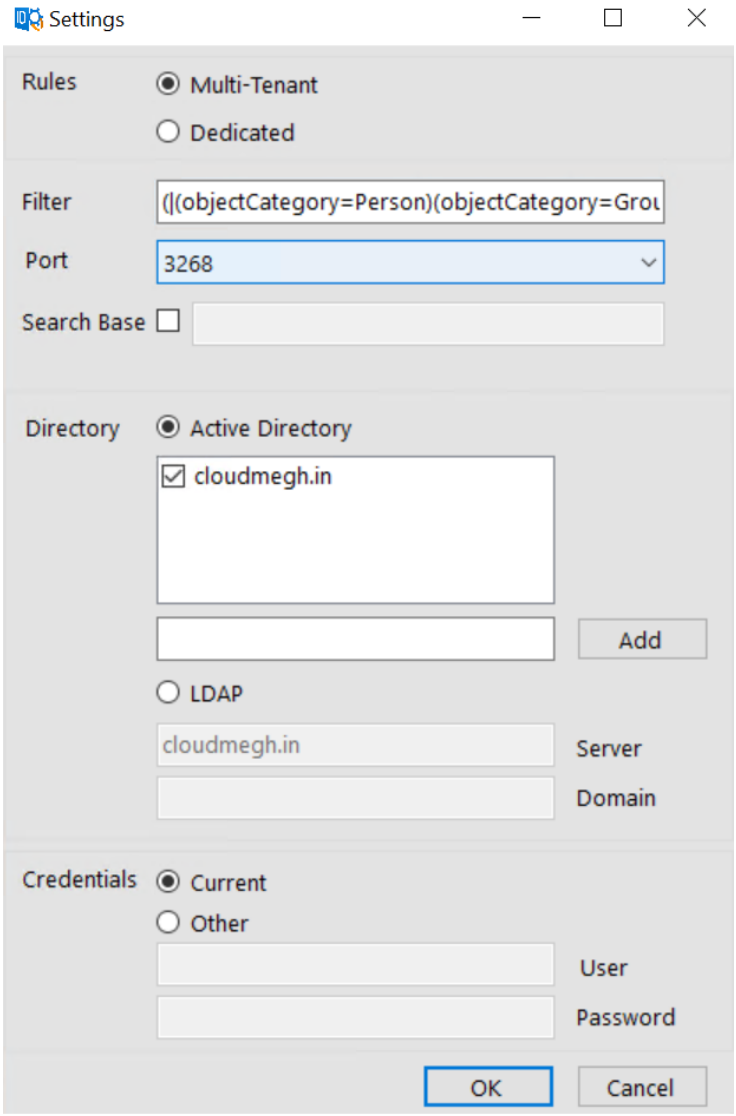
Before we synchronize our on-premises AD to Azure AD, its recommended to run IdFix tool, this tool identifies errors such as duplicates and formatting problems in your AD domain, you can run this tool on domain joined machine or Domain Controller itself.




IdFix Privacy Statement
This privacy statement applies to IdFix only. It does not apply to any other Microsoft Office 365 server or other applications by Microsoft Corporation. IdFix is intended for IT Administrators. If you are an end-user, Microsoft recommends that you end this program and contact your IT Administrator for assistance. IdFix collects data from your corporate network and can optionally store data that may contain personal data in a Comma Separated Value (.csv) or LDAP Data Interchange Format (.ldf) file located on the computer where the program was run. Microsoft recommends that you remove this file when you have completed your use of the program. IdFix does not send any collected data back to Microsoft or it's partners.
OK


IdFix version 1.11
Office 365
Query
Cancel
Accept
Apply
Export
Import
Undo
DISTINGUISHEDNAME
OBJECTCLASS
ATTRIBUTE
ERROR
VALUE
UPDATE
ACTION


IdFix version 1.11
Office 365
Query
Cancel
Accept
Apply
Export
Import
Undo
DISTINGUISHEDNAME
OBJECTCLASS
ATTRIBUTE
ERROR
VALUE
UPDATE
ACTION
Query Count: 72 Error Count: 0



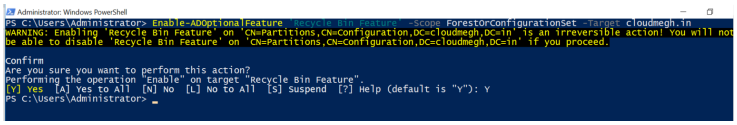
3. Enable AD Recycle Bin:

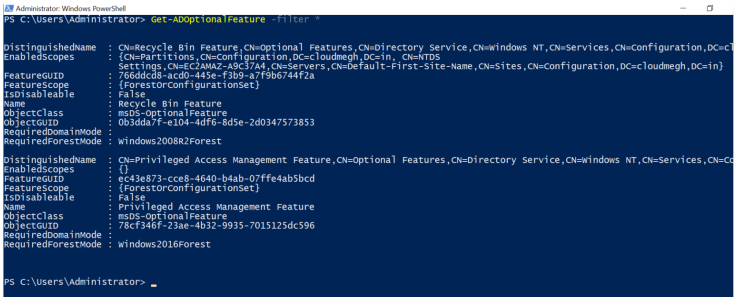
Microsoft recommends enabling AD Recycle Bin feature in your on-premises AD environment, If you accidentally deleted an on-premises AD user object, the corresponding Azure AD user object will be deleted in the next sync cycle. By default, Azure AD keeps the deleted Azure AD user object in soft-deleted state for 30 days.

I ran below PowerShell cmdlet to enable AD Recycle Bin feature for my AD domain.

Enable-ADOptionalFeature 'Recycle Bin Feature' -Scope ForestOrConfigurationSet -

Target cloudmegh.in



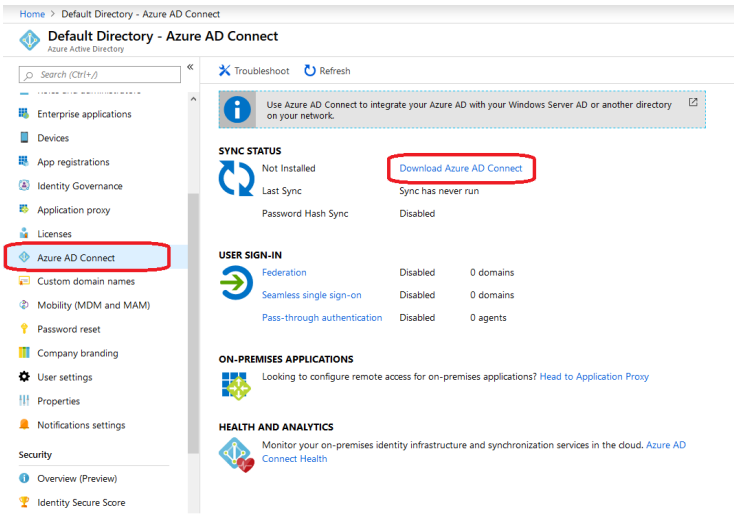


4. Download and install Azure AD Connect server:

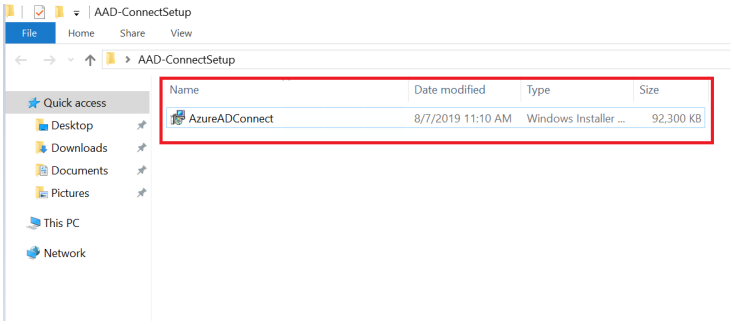
–Now I will go back to Azure portal and navigate to Azure Active Directory console, Go to Azure AD Connect and click on “Download Azure AD Connect” option which will redirect me to below download link.

Azure AD Connect download:

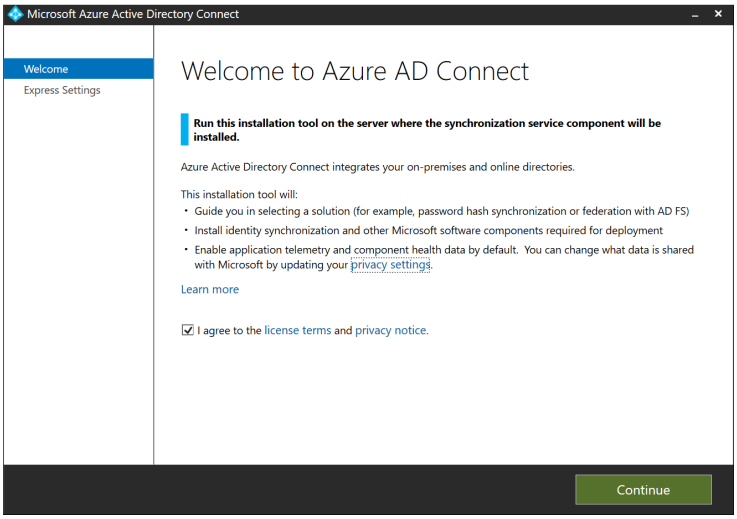
https://www.microsoft.com/en-us/download/details.aspx?id=47594



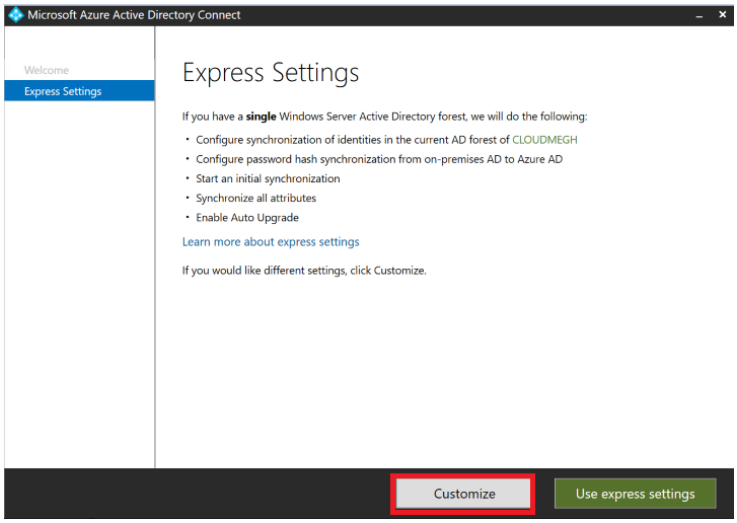
–I have downloaded AAD Connect setup on my on-premises Windows Server where I will be installing this. This machine is running Windows Server 2016 OS and joined to my AD domain "cloudmegh.in".



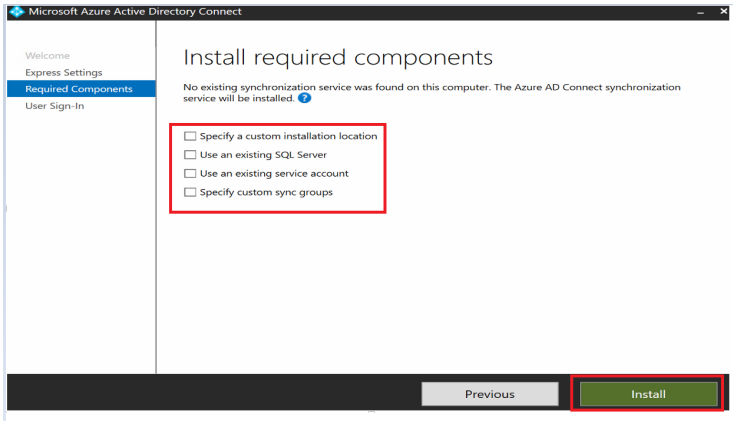
–I ran AAD Connect setup and installation wizard started, click on license terms agreement and then click on **"Continue"** button to proceed further.



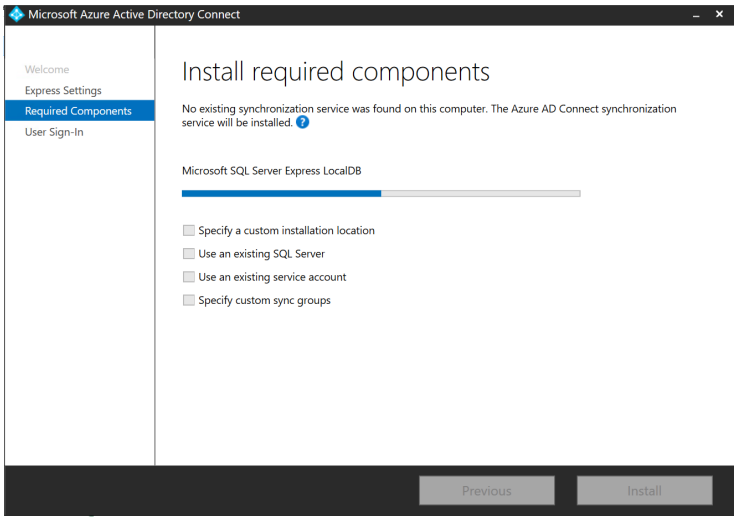
–If you want to install AAD Connect with "**Express Settings**" it will install and configure everything with predefined set of configuration as shown in below snapshot. I'm going with "**Customize**" installation option in my lab.



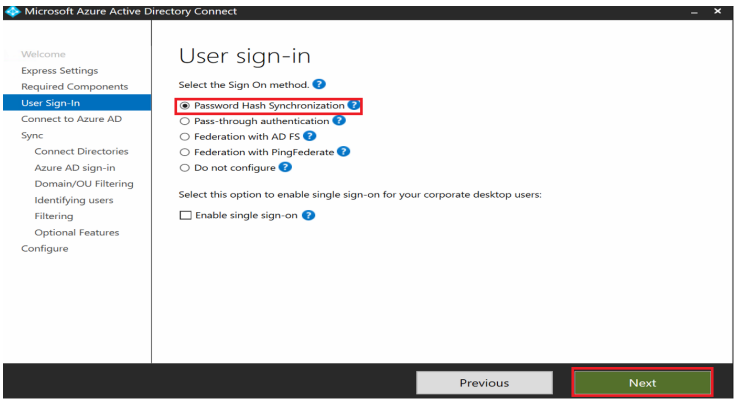
–I'm leaving below option to default (unchecked), you may specify the options as appropriate in your environment. I clicked on "**Install**" button to proceed with AAD Connect server installation.



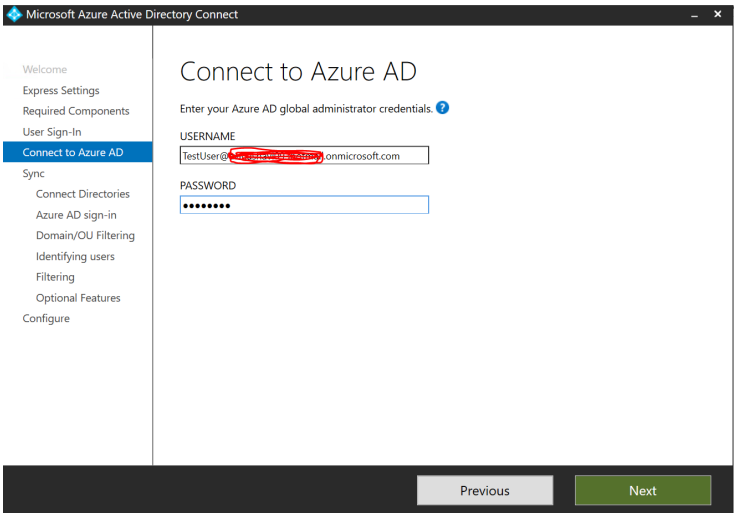
–It will install and configure the required components now, Azure AD Connect requires a SQL Server database to store identity data. By default a SQL Server 2012 Express LocalDB (a light version of SQL Server Express) is installed.

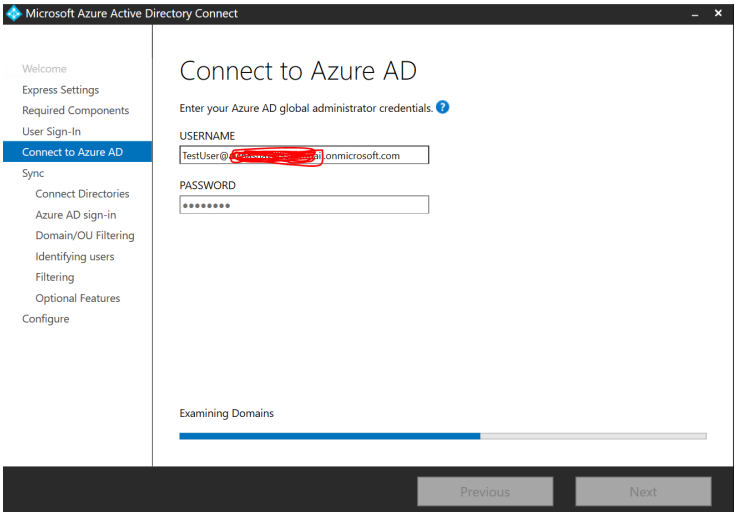


–Once it completes required components installation/configuration, it will move to below option where we need to choose authentication method for users. I'm choosing **"Password Hash Synchronization"** authentication mechanism. Click on "**Next**" to proceed further.

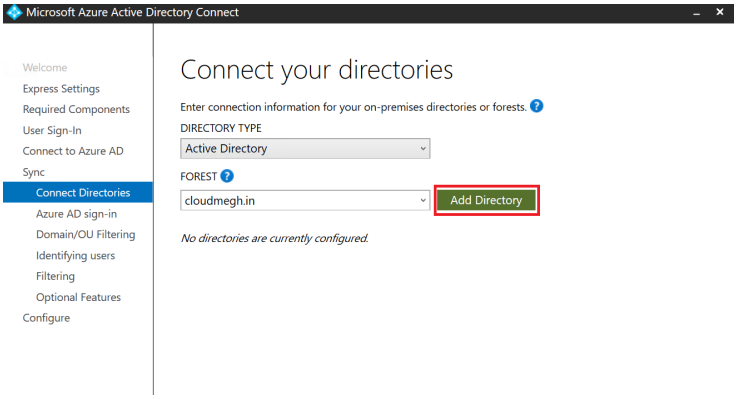


–Now we need to provide user credential that will be used to connect to Azure AD tenant, this user should have **Global Administrator** permission on your Azure AD tenant.





–Now select AD Forest by clicking on **“Add Directory”** which will be the connected data source for AAD Connect sync.

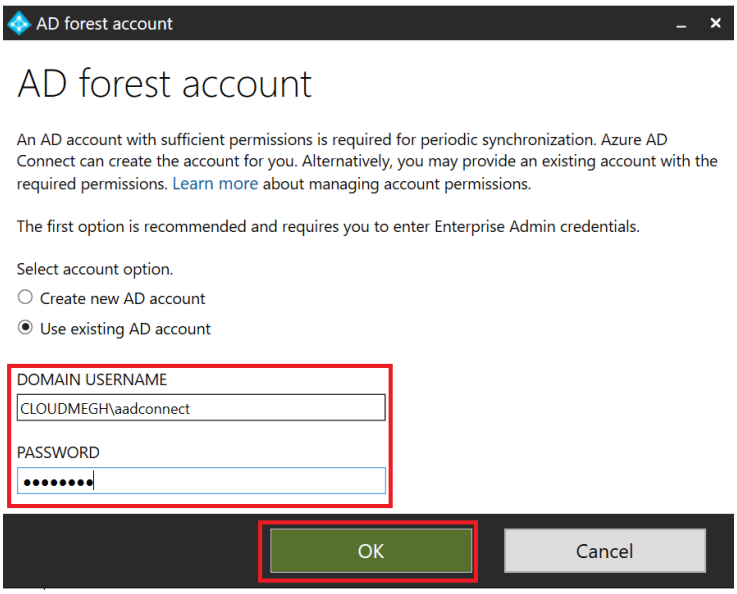


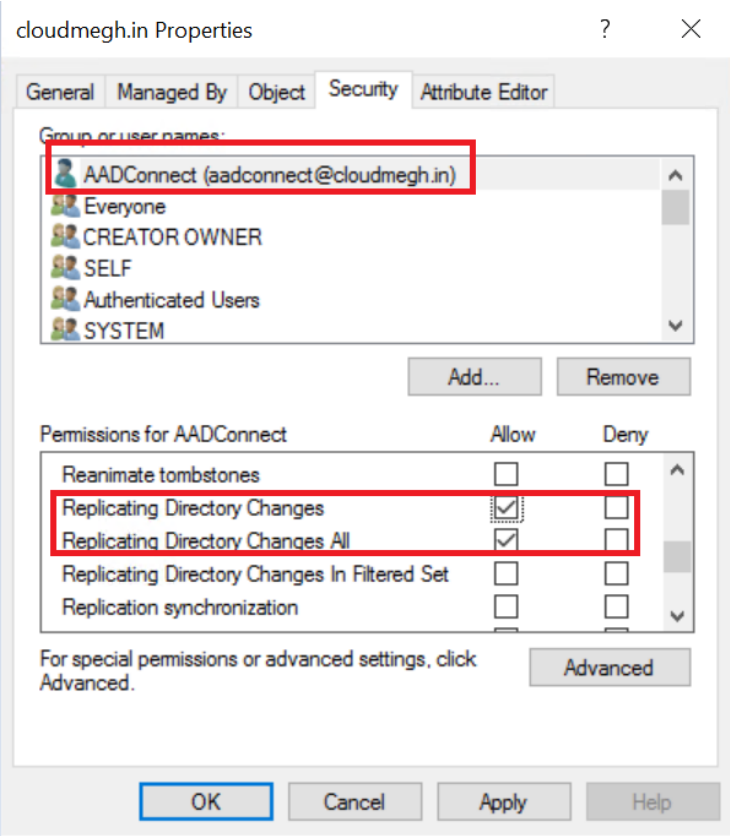
–Once you click on “Add Directory” it will ask you to provide user credentials that should have

minimum below permission for “Password hash sync” to function correctly. Depending upon what features you are using and what installation method permission requirement will change.

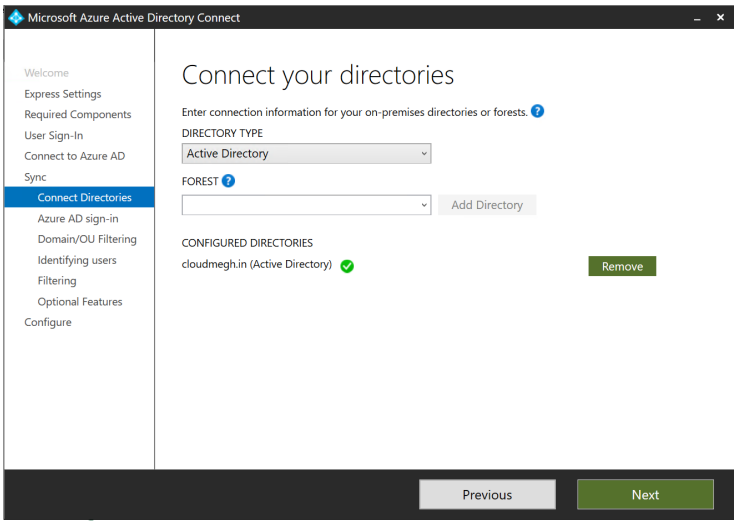
Replicate Directory Changes

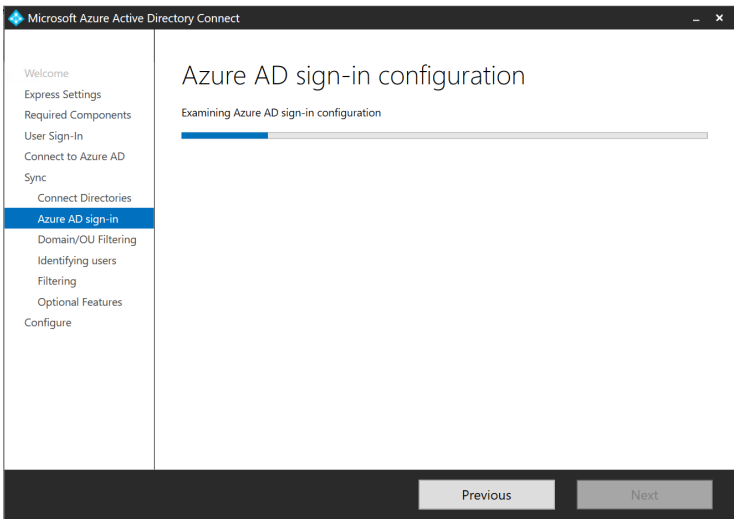
Replicate Directory Changes All





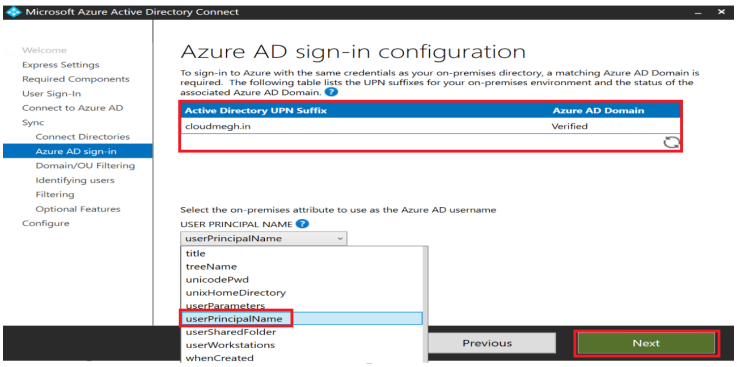
–We have now added ADDS Forest **“cloudmegh.in”** successfully, click on “Next” to proceed further.





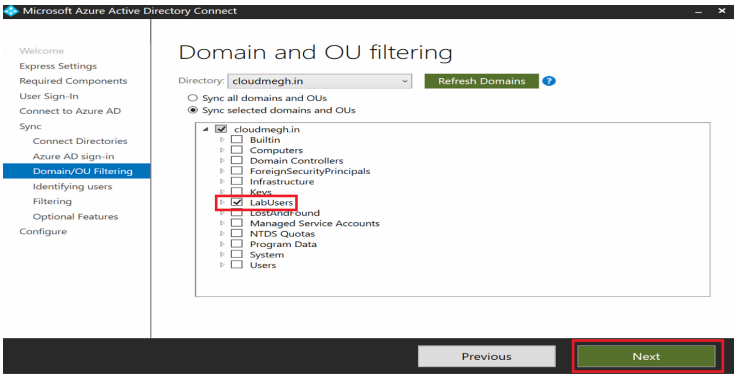
–This below page allows us to review the UPN domains present in on-premises AD DS and which have been verified in Azure AD. This allows us to configure the attribute to use for the userPrincipalName.

Note: *The attribute* ***userPrincipalName*** *is the attribute users use when they sign in to Azure AD and Office 365. The domains used, also known as the UPN-suffix, should be verified in Azure AD before the users are synchronized. Microsoft recommends to keep the default attribute* ***userPrincipalName****. If this attribute is non-routable and cannot be verified, then it is possible to select another attribute.*

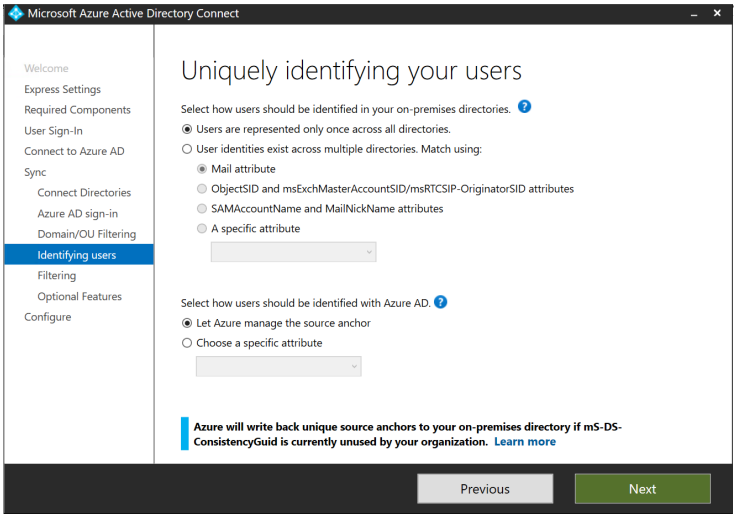


–By default all domains and OUs are synchronized. If there are some domains or OUs you do not want to synchronize to Azure AD, you can unselect these domains and OUs.

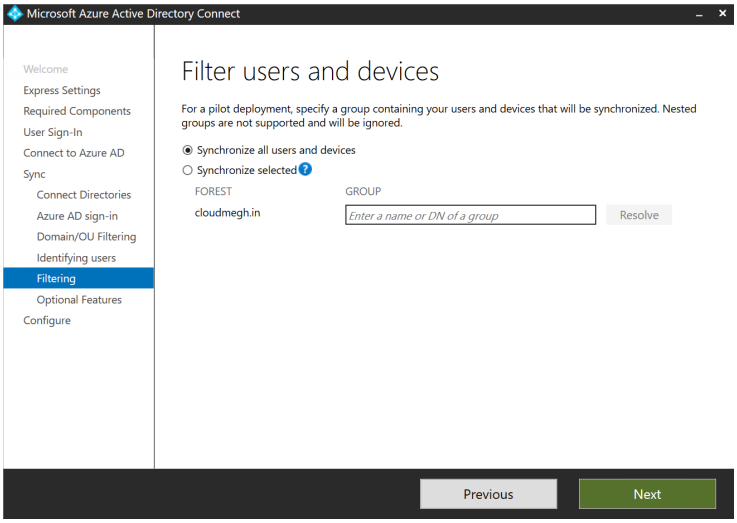
I have selected specific OU "**LabUsers**" from my AD domain, AD users under this specific OU will only be synchronized to Azure AD, click on "Next"



–I'm leaving below configuration to default ones and proceeding further.



–I'm not configuring any sync filtering at the moment.



–Now we have some **“Optional features”** of Azure AD Connect that we can enable based on scenario and requirement, I’m enable **“Password Writeback”** feature which is important in hybrid environment where Azure AD is connected to an on-premises ADDS environment, this scenario can cause passwords to be different between the two directories if users changes the password using Azure AD portal. By enabling password writeback, password changes that originate in Azure AD is written back to your on-premises directory.

–Click on “**Install**” button.

–It will take few minutes to complete the configuration.

–Finally the installation and configuration of AAD Connect has completed successfully and synchronization process has also been initiated.

–After completion of Azure AD Connect installation and configuration, if you want to see the existing configure or would like to modify something, you need to open “Azure AD Connect” console and review it.

–If you click on **“View current configuration”**, it will show you your existing your existing synchronization settings and features enable in your AAD Connect.

–If you open **“Synchronization Service Manager”** you can see what’s happening with AAD Connect sync engine from synchronization perspective and if there are any errors.

–You can also stop the sync run manually if needed from here.

Verify user sync from Azure AD console:

–Since synchronization process was already initiated above, let's go back to Azure AD console

and see if objects were synchronized successfully.

–Yes, I can see users from source "**Windows Server AD"** in my Azure AD tenant which confirms sync process was able to synchronize the users successfully.

–We can sync status and last sync details as well from here to validate when did the last sync happen.

I will continue to explore other capabilities in Azure AD in hybrid identity scenario and will cover it in next post.

← Azure Kubernetes Service (AKS) – Part 2

Leave a Reply

Your email address will not be published.

Required fields are marked *

Comment

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

Post Comment