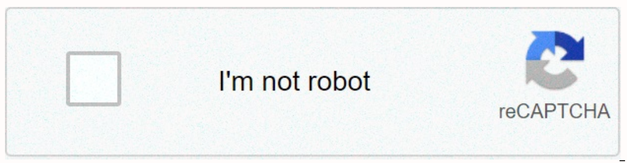

Continue

Azure ad proxy connector

Making it easier for users to access apps makes it easier for them to be produced wherever they are. Azure AD Application Proxy helps us achieve this and can help you improve site security too. Using Azure AD Application Proxy, users can access apps from anywhere outside the company firewall without the need for VPN access, and you can add MFA and security controls to apps that don't support it natively. With users working remotely, how do they access the internal line of enterprise systems such as web applications you have running? Chances are they'll have to use a VPN, but Azure AD Proxy Application could provide you with a better solution. Often we work with customers to help them move workloads to Microsoft Azure, but what if you want or need to keep something on site? Internal web applications were traditionally only accessible behind VPN because making them available outdoors was either too expensive, complex or insecure. Having apps accessible only behind the VPN means that users can be limited how, when and where they can access them so that we can get the most out of both worlds: internal applications that are published safely but without the associated costs? With remote work at a record level due to COVID-19, could reduce the pressure on your VPN solution improve performance for users who actually need to be on VPN? What if, for example, your employees could access your time tracking, HR or other workplace management systems from their smartphones or tablets, allowing them to access the data and submit it on the go? Azure AD Application Proxy allows these kinds of scenarios. What is Azure AD Application Proxy? Azure AD Premium P1 and P2 feature, it's a free solution available as part of your existing investment in Azure AD Premium. By using agents to negotiate connections from the outside to the inside, it allows you to publish applications on the web without the need to open incoming firewall ports or anything complex. Apps are created in Azure AD to support every web application you want to publish. Apps can be published as an option to specific users or limiting groups who can see the app as available and facing authentication and access with Azure AD Conditional Access, access to apps can be secured with the same policies we use for Microsoft 365: location, customer type, device type and other restrictions while adding options like multi-factoring authentication as well. Application management with Azure AD Application Proxy is a simple, effective and inexpensive option to integrate additional security into existing web applications when refactoring or redesigning the application itself is neither viable nor possible. How does Azure AD Proxy Application work? The way it works is remarkably simple. You download and install the Azure AD Application Proxy Connector service: a small headless service that works on one or more server servers your environment. It supports Windows Server Core making it extremely lightweight and can be installed on working group machines in the DMZ which means it doesn't even need to be inside your domain network. If you opt for a DMZ-based installation from Azure AD Application Proxy, this limits your application authentication options. You cannot use Kerberos-based authentication methods with DMZ-based deployments. Each group of connectors can support multiple services: you don't need to deploy a connector group for each application. Connector groups also easily metrics by simply adding capacity to existing connectors by increasing or adding additional connectors and increasing you as you bring more applications or more users start working through the service. Does it only work for Windows IIS web servers? No, Azure AD Application Proxy doesn't know or care about what the underlying platform is. Your web applications can windows IIS web servers, Apache Linux servers, or anything else. What about high availability? Azure AD Application Proxy offers high service-level availability by allowing you to deploy multiple agents in a connector group. When multiple connectors are deployed in a group of connectors, connections in the app from Azure AD are automatically balanced and distributed among them to ensure high availability and resilience allowing you to perform maintenance on an individual connector without downtime. What do we have multiple data centers? How is traffic routing managed? By creating multiple connector groups in Azure AD, you can deploy connectors in each data center location. When an app lives in a data center, you'll route it through that group; When one app lives in another data center, you set it up in the other. Using Windows PowerShell, you can easily script runbooks failover to allow you to exploit the routing of disaster recovery traffic to applications by moving an application between connector groups when the app moves to another data center. Can Azure AD Application Proxy only be used for on-site workloads? No, Azure AD Proxy Application will work wherever you have deployed web applications. If you have web servers running on-site, in third-party hosting providers (a Co-Lo), like infrastructure-as-a-Service virtual machines in Microsoft Azure or in Amazon AWS. Just deploy the connectors as you do on the spot and it still works. The migration of web applications to the AD Azure cloud application proxy is a great solution for when apps can't move, but how do you know? What makes a good candidate to switch to services like Azure App Service and host your web applications as Platform-as-a-Service instances instead of having to power and water an entire web server? The Microsoft App Service Migration Assistant (is a great tool to evaluate your and determine their suitability. Getting help with App Proxy or switching to the cloud Whether you want to publish your apps where they are today or want to evaluate your web applications to move to the cloud, Arcible can help. Our service modernization service can help you look at both approaches, while our web hosting service can help you set up new existing web applications or move them to the cloud. We can help you assess your existing workloads and find the best solution, tailored to everyone because there is no one-way approach. If you are interested in moving workloads to the cloud or want to review publishing options and allow your users to work efficiently remotely, then get in touch with us to learn more. I am not able to install Azure AD Proxy Connector Application on my Windows 2016 server that I get below problem when connecting during installation. I'm sorry, but we're having a hard time registering. AADSTS50020: 'xxxxxxxx@hotmail.com' user account of the 'live.com' identity provider does not exist in the 'Microsoft Services' tenant and cannot access the '55747057-9b5d-4b5d4-b387-a52a8bd489' (Azure AD Application Proxy Connector) in this tenant. The account must first be added as an external user in the tenant. Sign in and log in again with another Azure Active Directory user account. How to solve the above problem. This blog will show you how you can access the basic apps from anywhere using Azure AD Proxy Application. In my example, we're going to set up Windows Admin Center on a domain network. This domain will be synchronized with Azure AD with a single sign activated. This guide assumes that you have already connected Azure AD with ADDS for Single Sign OnInstalled your Enterprise App on the site that uses integrated Windows AuthenticationA admin access to both environments The guide will cover What is Azure AD App Proxy? Installing Azure AD Proxy ConnectorConfiguration of a single-panel enterprise application configuration for this applicationGing of a user to that app Access that app from the M365 portal What is Azure AD App Proxy? Application Proxy is an Azure AD feature that allows users to access web applications on-site from a remote customer. Proxy app includes both the proxy application service that runs in the cloud, and the application Proxy connector that runs on an on-site server. Azure AD, the application Proxy service and the proxy app connector work together to securely transmit the Azure AD user's login token to Web. Architecture shown below. Installing the AD Azure First proxy connector, you want to identify the Windows server in which you will use for the application Proxy server. Connect to this server and take the next steps. Connect to Azure Active Directory Admin CenterGo to the Proxy app and click Download Connector ServiceAccept and accept the termsInstall on Server. This will require you to log in with your account at M365. Once completed, you should receive a message of success. Check and confirm that your connector is active in an Enterprise app proxy configuration In this section, you'll add your app on-site. You will need the internal URL for this application and port number. In my case, I set up Windows Admin Center to use a custom port of 4443 and redirect https traffic to https. My internal URL is . Go to Azure Active Directory's Enterprise Applications on 'New App' and select On-site application details such as the name, internal URL and URL you want for the proxy. Click Add when the details are filled in. Keep the AAD as a pre-auth. Set up the unique sign for the app There are a few additional setup steps you need to take in order to activate a single seamless sign on the experience. You may receive a Kerberos delegation error or a prompt to log in when you open the app. In my case, I need to make a change to allow Kerberos delegation limited to the server object where the application (Windows Admin Center) resides. This change was made to the AD object. Go to your domain controllerLocate the computer object of the server running the proxyRight connector click and select Properties to DelegationSelect Trust this computer for delegation to specified services only. Select Use any authentication protocol. Under which this account can present delegated credentials add value to the SPN identity of the application server. This allows the application proxy connector to impersonate users in ad compared to the apps set in the list. In my example, I used a server for everything. I add the name of the server with the HTTP protocol to access the web page of my application. Once this task is complete, we will not activate the single sign from the app itself. You will need to return to Azure AD Portal. In the AD Azure portal, go to Enterprise ApplicationsSearch and select the app you created previouslyGo to the single sign on windows bladeClick built authentication Enter your SPN and logon identity. In my case, the SPN was for http and my server on prem. HTTP/2019-dc.domain.local and delegated identity were left as UPN. Click Save Assign a User to the App From the Azure Ad Portal, you'll assign a user to your app. The user I assign is an on-prem user synced with Azure AD Connect where a single panel Activated. Access Azure AD Portal And YourApp Applications and GroupsAdd users who need access to the app. M365 Access Application Once the app is assigned, the user must see it in the M365 portal. Go to and select All apps. You'll see your app here. Click on your new app and wait for it to load, without further authentication. This has been shown below. This app uses the and has no incoming NAT rules on the server. The computer used to access it is on a different network. Network.