Microsoft

Premier Field Engineering

Azure AD Application Proxy test site setup

Prepared for

CVS Health

September 27, 2019

Version 1.00

Prepared by

PFE Name

Jeff Gilbert

Revision and Signoff Sheet

Change Record

Date	Author	Version	Change Reference

Reviewers

Date	Name	Version Approved	Position

Table of Contents

Azure AD Application Proxy test sites

Overview

Use these instructions to publish two internally hosted applications by using Azure AD Application Proxy and perform the pre-authentication on Azure AD. One of the applications is using Windows Integrated Authentication (WIA).

Objectives

In this section, you will:

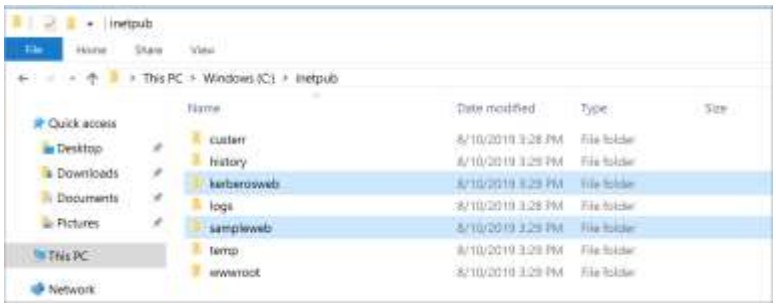
- Configure an IIS server to host two sample applications.
- Publish a sample app to make it accessible for your users outside the private network by setting the pre-authentication method to Azure AD.
- Configure a WIA application in Azure AD by giving Application Proxy Connectors permission in Active Directory Domain Services to impersonate users and send/receive tokens on their behalf (Kerberos Constrained Delegation).

Configuring the Web Server

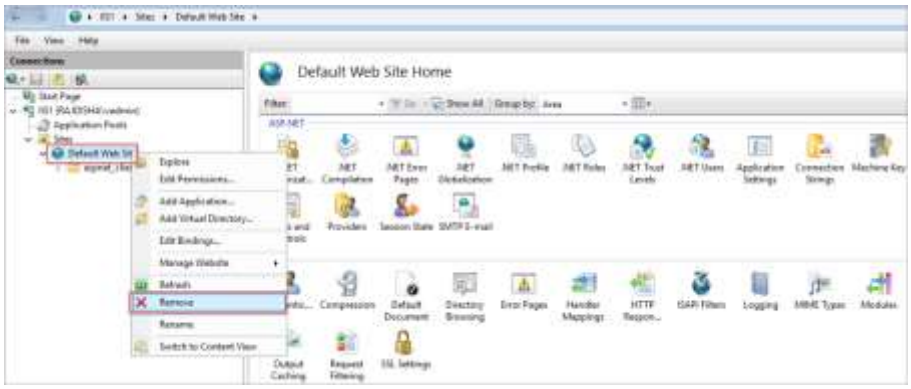
1. Sign into the IIS web server.
2. Run the following Windows PowerShell command to install all of the necessary IIS roles and services for this exercise.
   Install-WindowsFeature -Name Web-Windows-Auth,Web-Server,Web-WebServer,Web-App-Dev,Web-Net-Ext,Web-Net-Ext45,Web-Asp-Net45,NET-Framework-Core,Web-Custom-Logging,Web-Dir-Browsing,Web-Http-Errors,Web-Http-Logging,Web-Static-Content,Web-Stat-Compression,Web-Performance,Web-ISAPI-Ext,Web-ISAPI-Filter,NET-Framework-45-ASPNET,Windows-Identity-Foundation,Web-Asp-Net,Web-Mgmt-Console,Web-Mgmt-Compat,Web-Metabase

3. Disable **IE Enhanced Security configuration** from Server Manager.

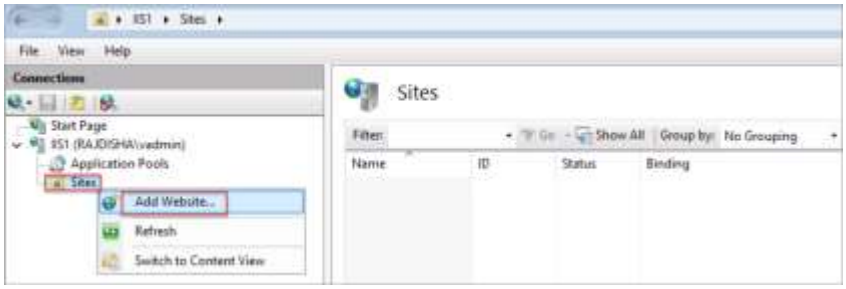
4. Save the **kerberosweb** and **sampleweb** folders from the zip files containing the web applications included with this documentation to the **C:\inetpub** folder.



5. Open the **Internet Information Services (IIS) Manager** console from the Server Manager **Tools** menu. Browse to the **Default Web Site**, right-click, and then select **Remove**. Click **Yes**.



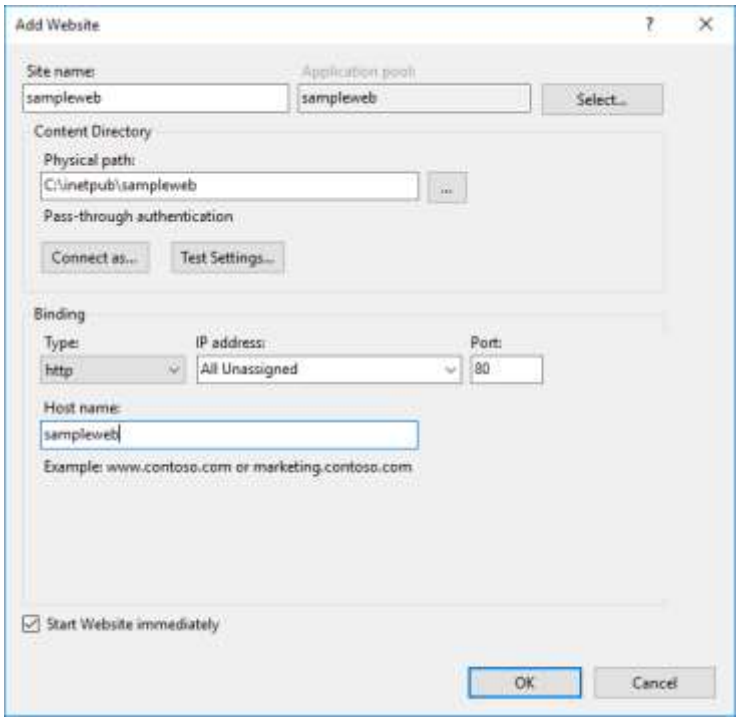
6. Browse to **Sites**, right-click, and then select **Add Website**.



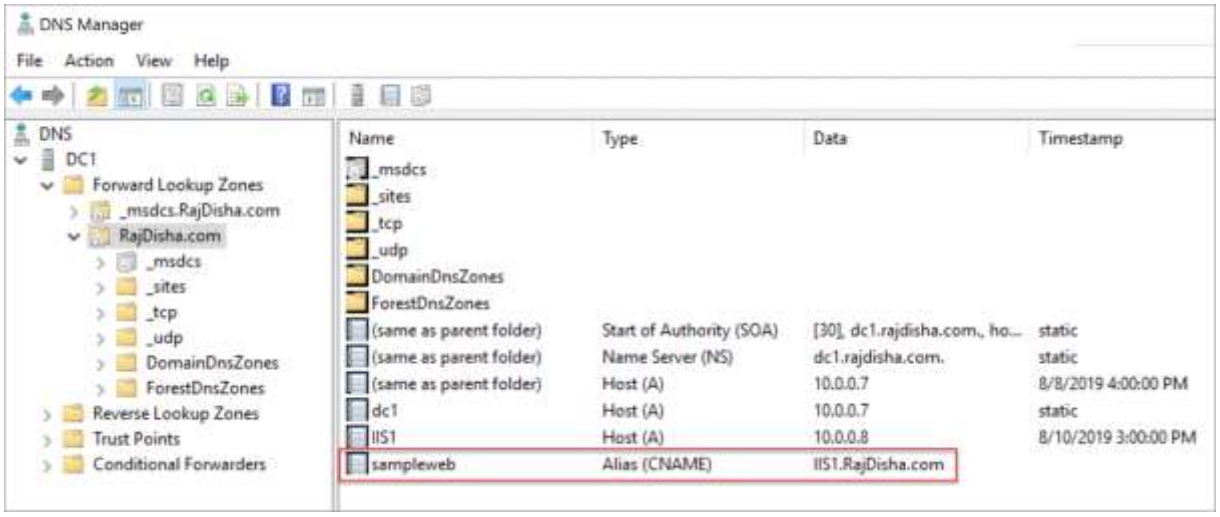
7. In the **Add website** dialog box, use the following settings for configuration:

   - Site name: **sampleweb**
   - Physical path: **C:\inetpub\sampleweb**
   - Host name: **sampleweb**

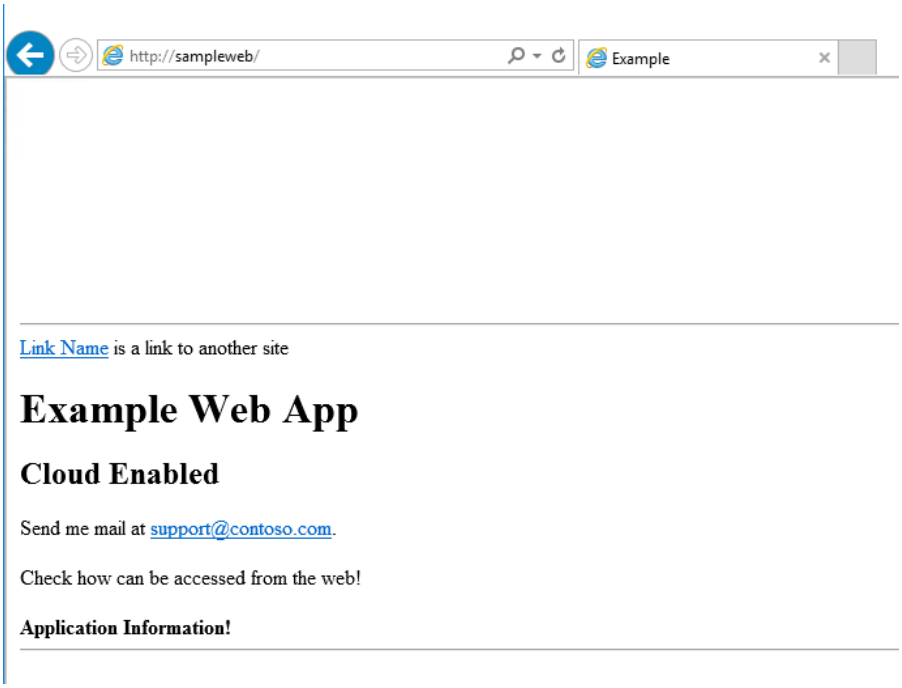
8. Click **OK**.



9. **Important:** You should create the canonical name (CNAME) record at the Domain Name System (DNS) zone of the Active Directory domain to which the IIS server is joined, and point the **sampleweb** record to the IIS Server entry.



Confirm the operation by opening a web browser and going to **http://sampleweb**.

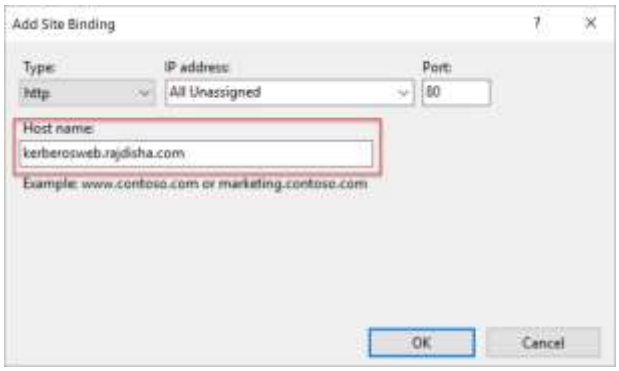


10. Repeat the previous **steps 6 to 9** for the **KerberosWeb** application.
11. After configuring the **KerberosWeb** application, go back into **Internet Information Services (IIS) Manager**, and then click **Kerberosweb**. On the right panel, double-click **Authentication**.

12. Right-click **Windows Authentication**, and then click **Enable**.

13. Right-click **kerberosweb**, and then click **Edit Bindings**.

14. Click **Add** to add the fully qualified hostname of **kerberosweb.*yourdomain.com***, and then click **OK**. (Where yourdomain.com is the fully qualified domain name (FQDN) of the Active Directory domain to which the web server is joined.)

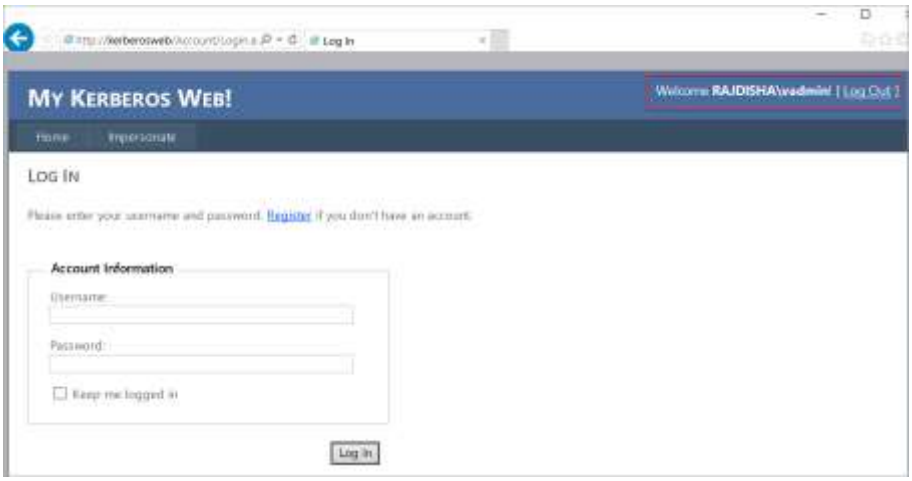


You should be able to view the two bindings as the following screenshot displays. Click **Close**.

Important: If it is not already done, You should create the CNAME record at the DNS zone of the Active Directory domain to which the IIS server is joined, and point the **kerberosweb** record to the IIS Server entry.

15. Confirm the operation by opening a web browser and going to **http://kerberosweb**.

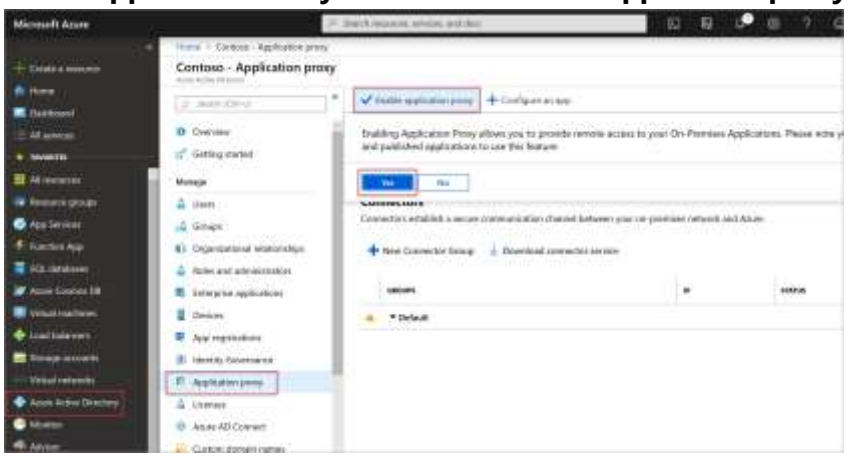
16. Click **Log In** at the top-right corner. It should then display the identity of the signed-in user.



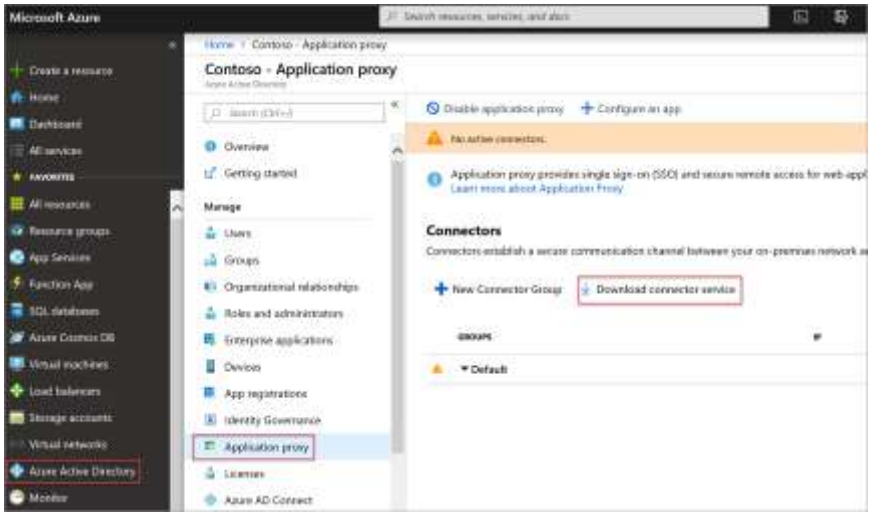
Enable the Azure AD Application Proxy Service

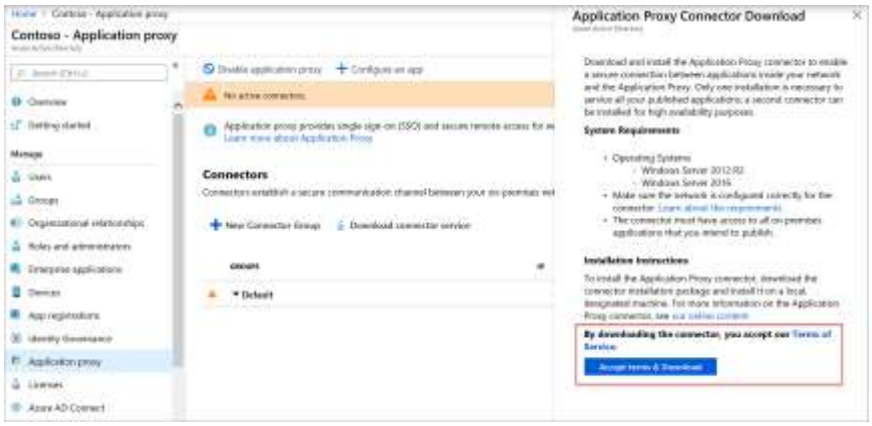
Enabling Application Proxy in Azure AD

1. Sign into the Azure Portal https://portal.azure.com as a Global Administrator. And select **Azure Active Directory** on the left-hand side.
2. Select **Application Proxy** and click on **Enable application proxy**. Click **Yes**.



3. Click **Download connector service**. This will take you to the download page. Click on **Accept terms & Download**. save the Windows Installer file (.exe) for the Application Proxy Connector at the C:\ drive.





Install and Register the Connector

1. Run **AADApplicationProxyConnectorInstaller.exe** on the **IIS Server**.
2. Read and accept license terms, select the **I agree to the license terms and conditions** check box, and then click **Install**.

3. During installation, you will be prompted to register the Connector your Azure tenant. Provide your Global Admin credentials.

Note: Ensure the admin who registers the Connector is in the same directory where you enabled the Application Proxy service.

Important: If the **IE Enhanced Security Configuration** option is set to **On** in the server where you are installing the Azure AD Connector, the registration screen might be blocked.

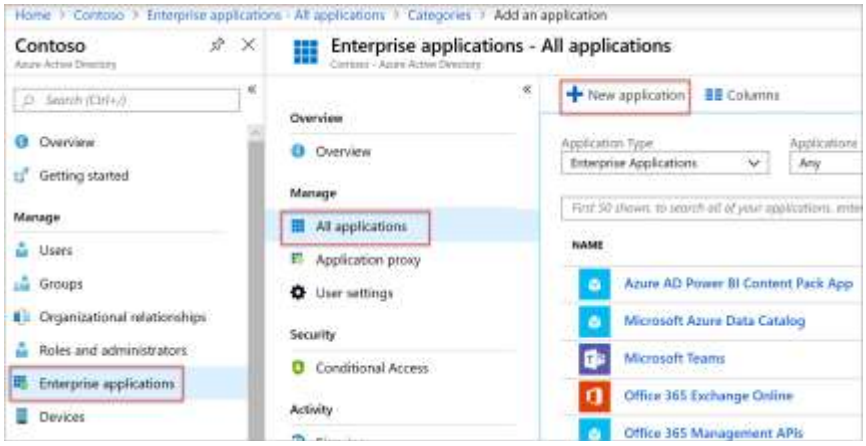
4. Ensure that you get **Setup Successful** page and click **Close**.

If environment has outbound proxy, follow instructions from the article: https://go.microsoft.com/fwlink/?linkid=869882

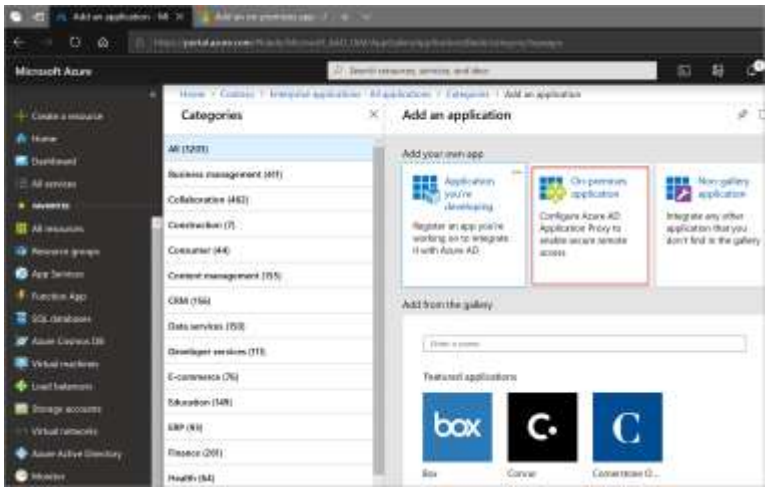
5. When the installation completes, two new services are added to your server as shown in the following screenshot. These are the Connector service, which enables connectivity, and an automated update service, which periodically checks for new versions of the Connector and updates the Connector as needed.

Publish a Sample App with the Azure Application Proxy

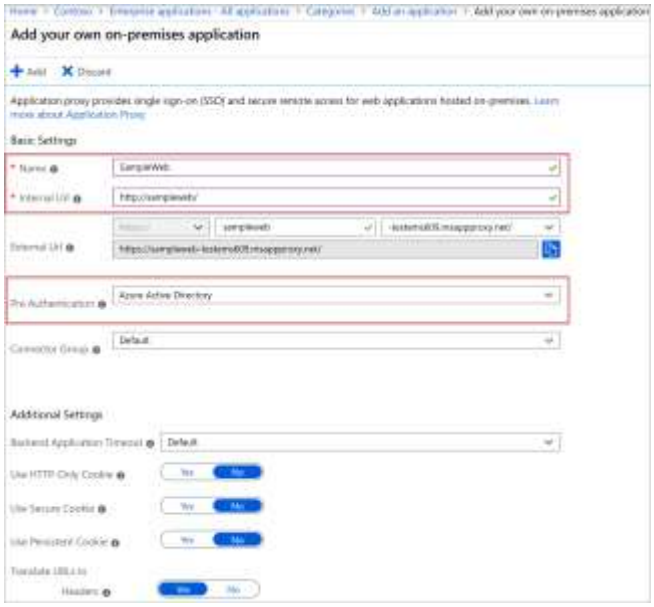
1. Sign in to the Azure Portal https:/portal.azure.com as a Global Administrator.
2. Select the directory where you enabled Application Proxy from drop down menu and go to **Azure Active Directory**.
3. Click the **Enterprise Applications**, and then click the **New Application** button at the top of the screen.



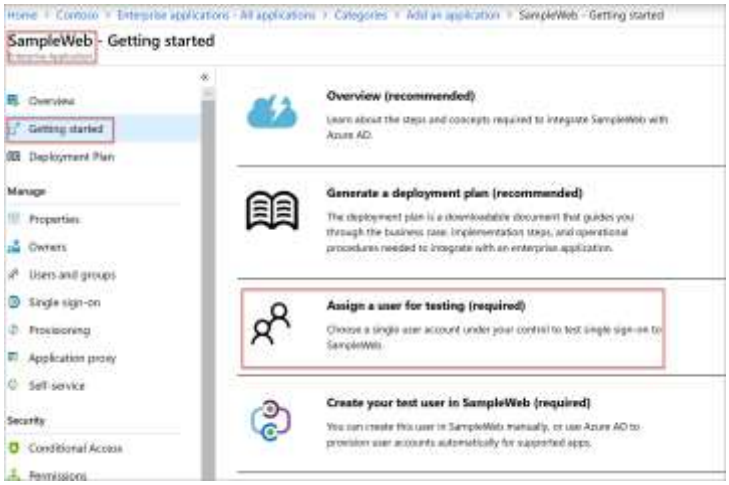
4. Select **On-premises application**



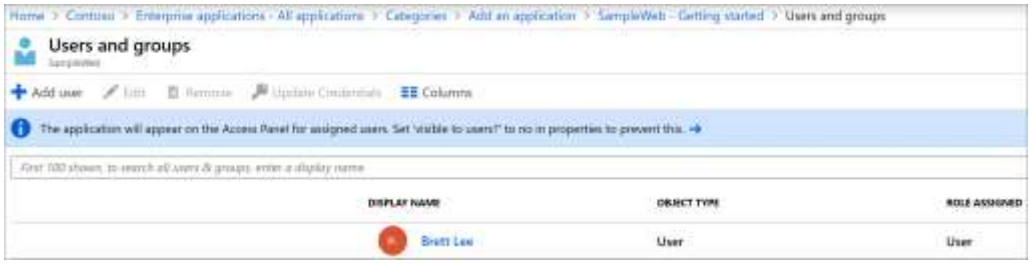
5. In the **NAME** field, type **SampleWeb** to provide a descriptive name for the application.
6. In the **Internal URL** field, specify the internal URL that the Application Proxy connector uses to access the application internally. This should be the URL of the published application that is used to access the application from inside your private network. In this case, enter **http://sampleweb/**
7. Set the **Pre Authentication** method to **Azure Active Directory**.



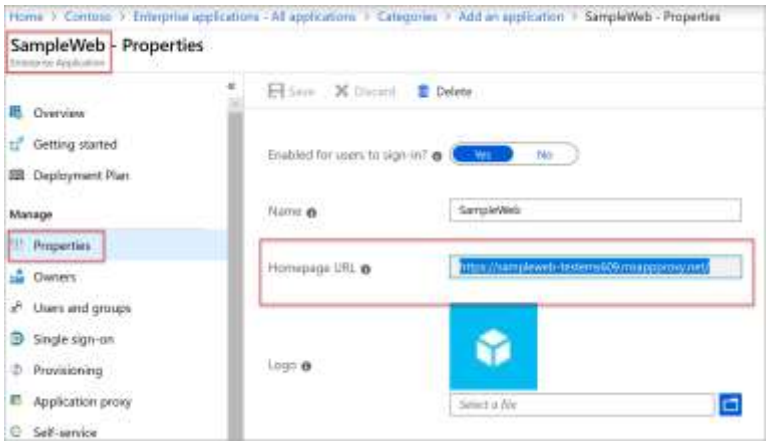
8. Click **Add** at the top of the blade.
9. For apps that are pre-authenticated, you must assign users and groups that will have access to the app. Click on the **Getting Started** blade.
10. From here you can use **Assign a user for testing (required)** to add a user.



Add user to whom you would like to assign this application.



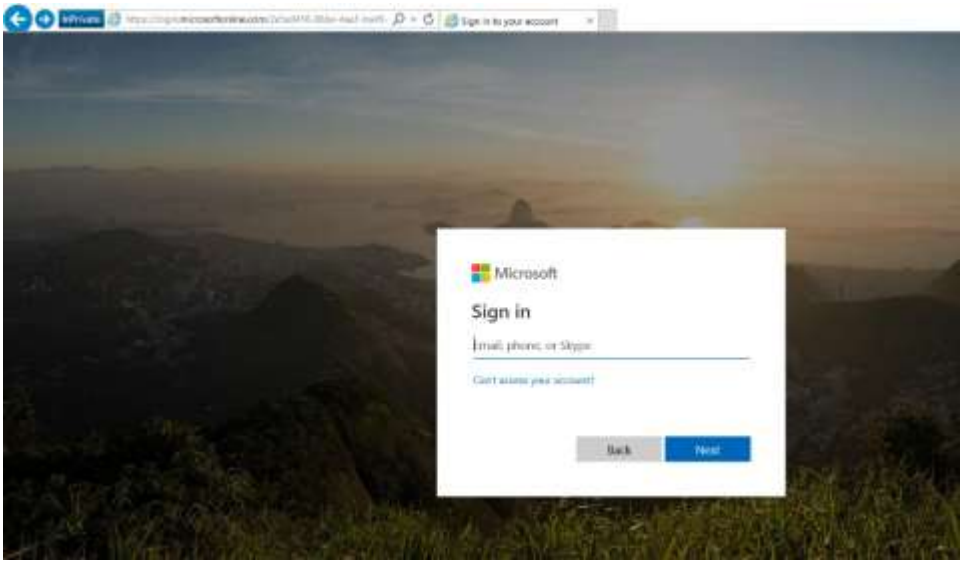
11. Go to **Properties** blade and copy **Homepage URL** of your application to clipboard.



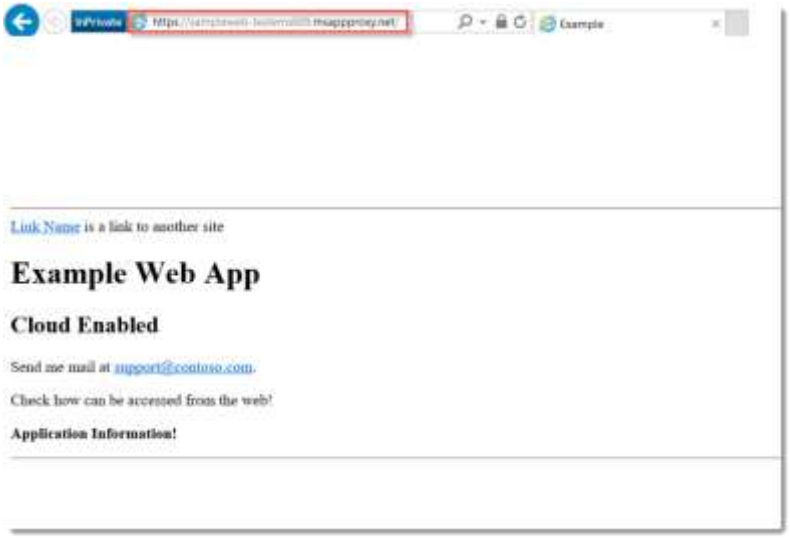
By default, the application will be published with the name of the application that you entered earlier (sampleweb), followed by a dash **–**, the name of your azure AD tenant followed by .msappproxy.net (sampleweb-*yourtenant*.msappproxy.net).

> **Note**
>
> This can be modified to use your own custom domain by uploading a certificate in PFX format to the configure section of the application properties. For more information, see https://docs.microsoft.com/en-us/azure/active-directory/active-directory-application-proxy-custom-domains

12. After the application is successfully published, open a browser, and paste the URL of the application. It will be like https://sampleweb-yourtennant.msappproxy.net.
    You can use another workstation to access application, not only the web server itself.
    If workstation was used before with some Azure AD account – it makes sense to open In Private browsing session for testing purposes.
13. Because the application uses Azure AD for authentication, and you have not authenticated to the application yet, you will automatically be redirected to the Azure AD sign-in page.



14. Use any of the user accounts that were assigned for **sampleweb** application before, and sign in.



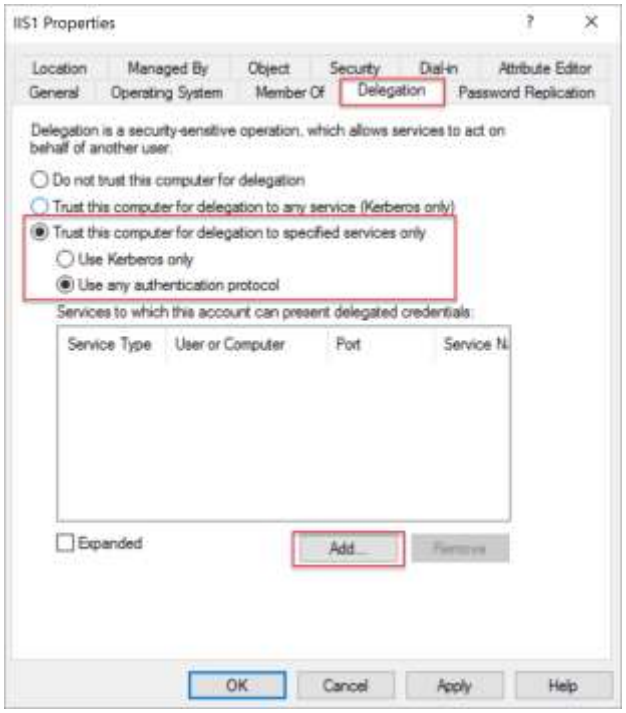
Publish a Windows Integrated Authentication App by Using the Azure Application Proxy

1. Create Service Principal Names (SPN) for your applications. Use the **SetSPN** tool to do this on a machine that has AD DS tools installed. Domain Admin rights are generally needed for this.

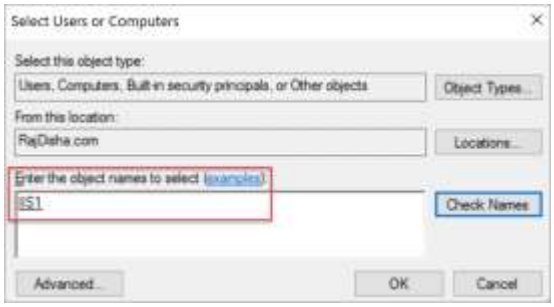
    - Type **setspn –a http/kerberosweb *IISServerName***, where IISServerName is the name of your webserver.

    - Type **setspn –a http/kerberosweb.*yourdomain.com IISServerName***, where yourdomain.com is the FQDN of the Active Directory domain which the web server is joined and *IISServerName* is the name of your web server.

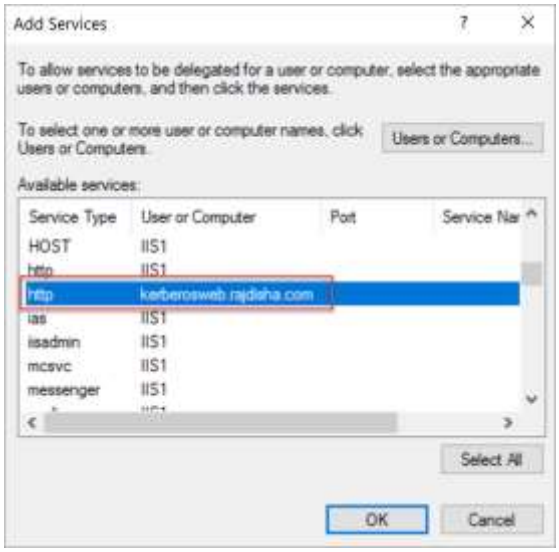
2. Validate the proper creation of the SPN by running **setspn –l *IISServerName***, where *IISServerName* is the name of your web server.
3. Open **Active Directory Users and Computers**, and select the server running the Connector (your web server). **Right-click**, select **Properties**, and then click **Delegation**. Select **Trust this computer for delegation to specified services only** and then select **Use any authentication Protocol**.
4. Click **Add**.



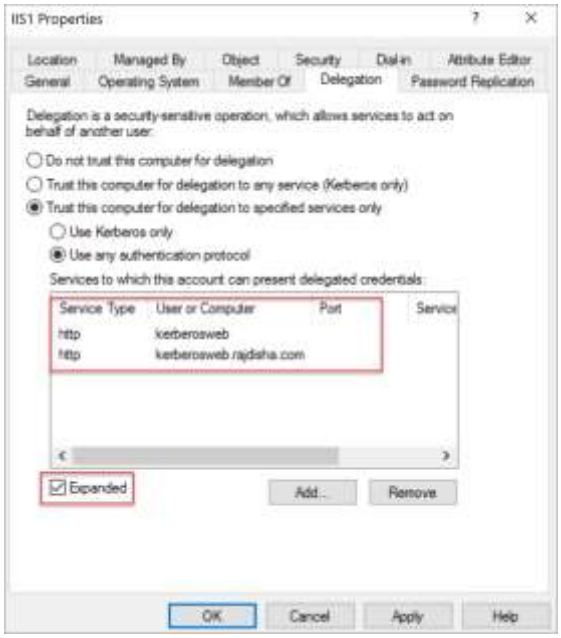
5. Within **Add Services**, click **Users or Computers**, and then enter the name of your web server and click **OK**.



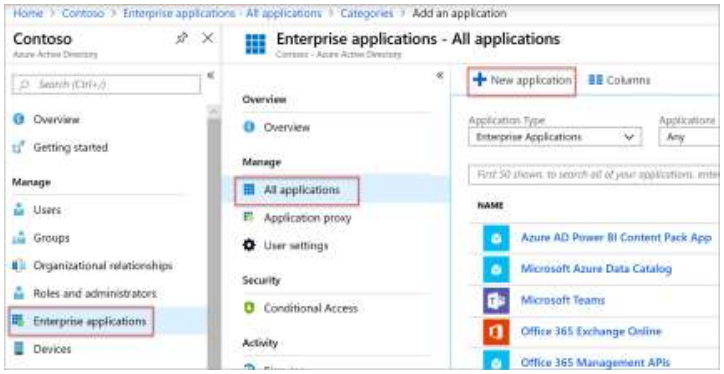
6. Scroll down and select the http kerberosweb SPN.



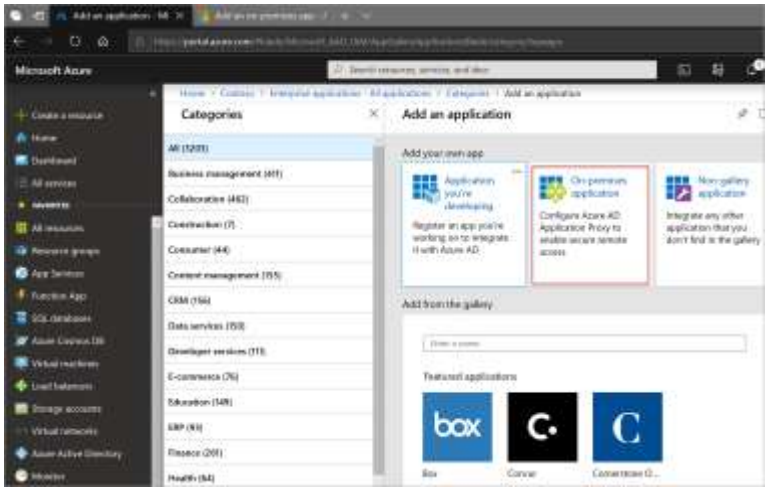
7. Click **OK**, validate the selection, and then click **OK** to close computer properties window.



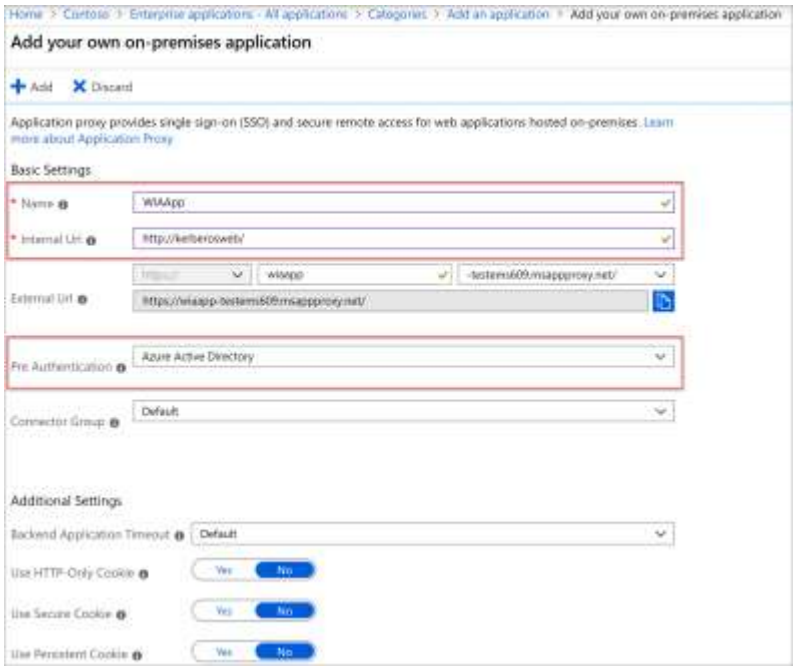
8. Restart your Web / Azure AD Application Proxy Connector server to ensure, that account will get appropriate permissions.
9. Sign in to the Azure Management Portal https://portal.azure.com as a Global Administrator.
10. Select the directory where you enabled Application Proxy from drop down menu and go to **Azure Active Directory**.
11. Click the **Enterprise Applications**, and then click the **New Application** button at the top of the screen



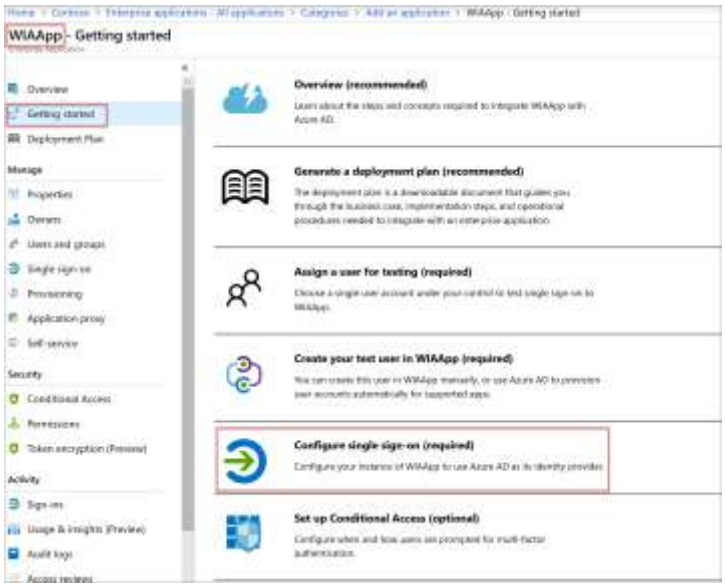
12. Select **On-Premises application**



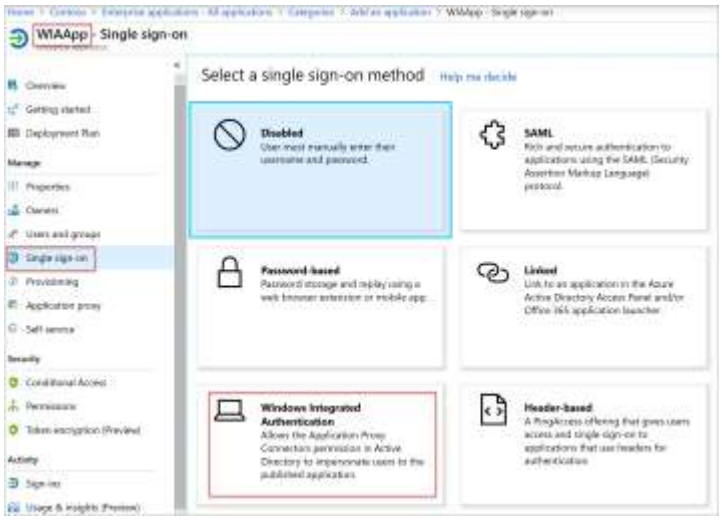
13. In the **NAME** field, type **WIAApp** to provide a descriptive name for the application.
14. In the **Internal URL** field, specify the internal URL that the Application Proxy connector uses to access the application internally. This should be the URL of the published application that is used to access the application from inside your private network. In this case, enter **http://kerberosweb/**
15. Set the **Pre Authentication** method to **Azure Active Directory** and click **Add**.



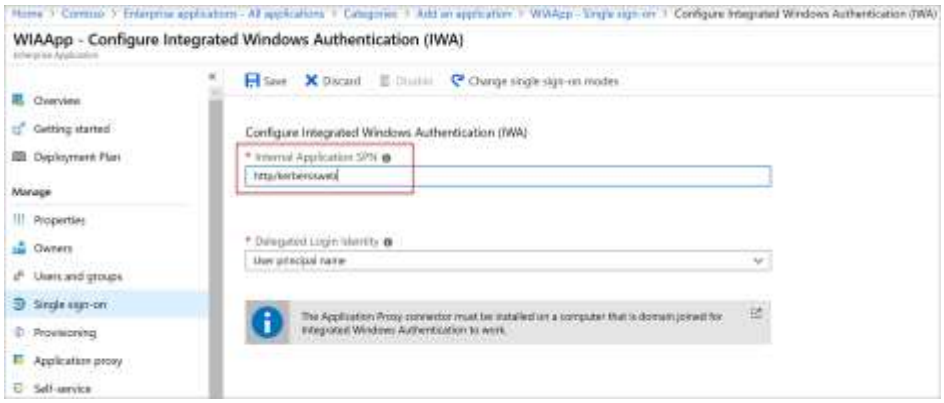
16. After your application is added you will be redirected to **WIAApp** application page. Click on **Getting Started** and then Select **Configure Single Sign On (required)** in the list.



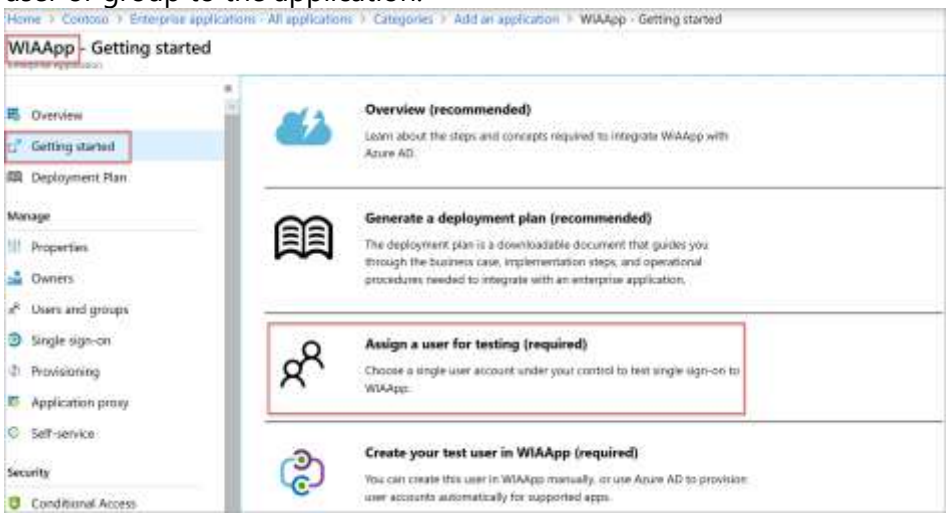
Select **Windows Integrated Authentication** from the list.

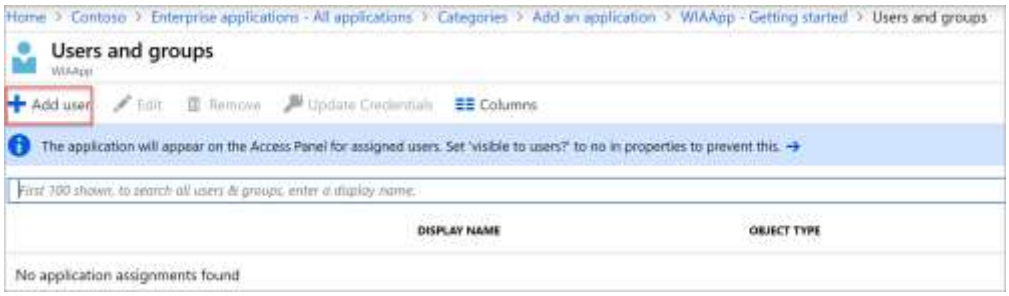


17. Enter the **Internal Application SPN** of the application server. This is the SPN of the internal application as configured in the on-premises Active Directory. This SPN is used by the Application Proxy Connector to fetch Kerberos tokens for the application by using Kerberos Constrained Delegation (KCD). Enter **http/kerberosweb**.

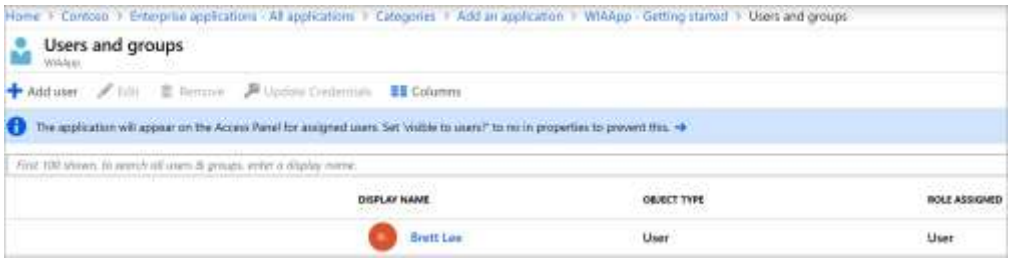


18. Click **Save** and close the blade.
19. On the **Getting Started** Blade select **Assign a User for Testing (required)** and assign a user or group to the application.

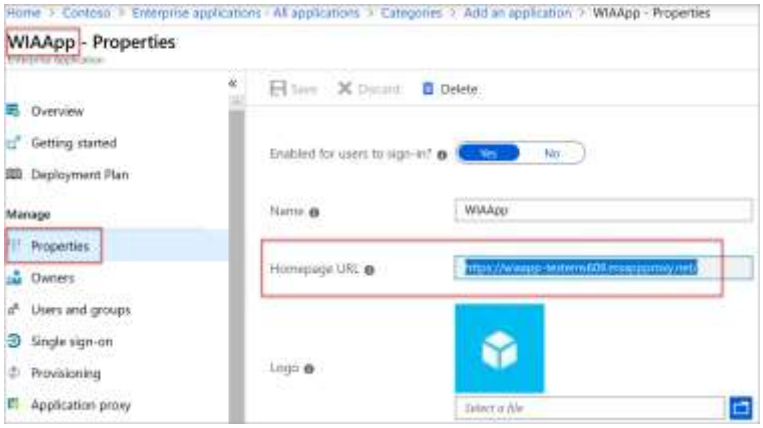




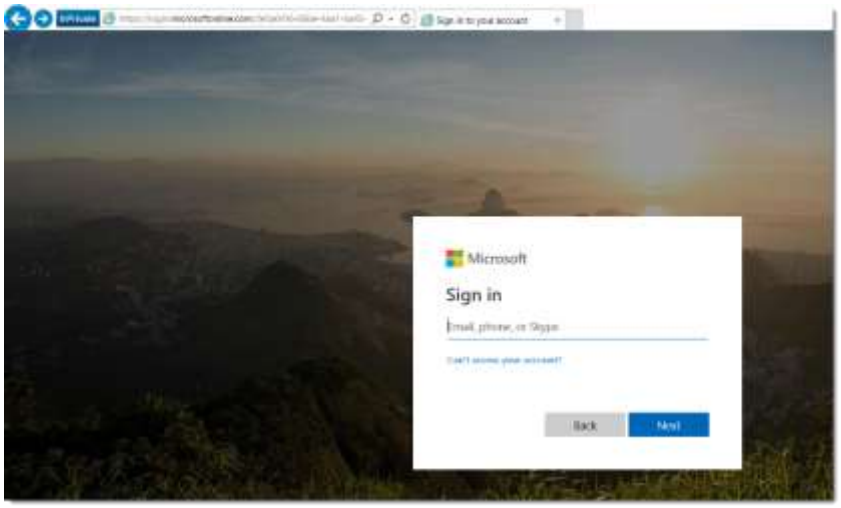
Add user to whom you would like to assign this application.



20. Go to **Properties** blade and copy **Homepage URL** of your application to clipboard.



15. After the application is successfully published, open a browser, and paste the URL of the application. It will be like https://wiaapp-yourtennant.msappproxy.net.
    You can use another workstation to access application, not only the web server itself.
    If workstation was used before with some Azure AD account – it makes sense to open In Private browsing session for testing purposes.



21. Sign in using one of the synced users which you assigned to the application. **Note**: You will need to assign the user an EMS or Azure AD premium license.


wiaapp-testems609.msappproxy.net
My Kerberos Web!
Welcome RAJDISHA\Bilal [Log Out]
Home
Impersonate
Welcome to Kerberos Web!
Object
Value
Windows Identity
IS APPPOOL\KerberosWeb
IsAuthenticated
True
ImpersonationLevel
None
Thread Identity
RAJDISHA\Bilal
Authorization Data
AuthenticationType
Negotiate
Token
1608