ENTRUST
SECURING A WORLD IN MOTION.

INTEGRATING WITH ENTRUST CONNECT FOR MICROSOFT AZURE

Release: 1.0.0

Document issue: 1.2

Date of issue: March 2024

Help us to improve our documentation. Please click this link. and take our survey.

Member of
Microsoft Intelligent
Security Association

Microsoft Security

Table of contents

Revision, audience, and guide information

Revisions

Revision	Section	Description
1.0		First release of guide
1.1	Integrating with Microsoft Azure Key Vault	Added Step 4 & 6
1.2	Minor changes	Fixed broken URL and replaced “Azure Active Directory” with “Microsoft Entra ID”

Audience

This guide is intended for Entrust Certificate Services (ECS) users who need to integrate Azure Connect with Microsoft Key Vault.

Viewing this guide

Although this guide can be printed, it relies on hyperlinks to other sections. It is best viewed and used electronically.

Prerequisites

This guide assumes that your company already has:

- an ECS account and access to the Certificate Services REST API
- a Microsoft Azure account
- downloaded the **Entrust Connect for Microsoft Azure** binaries from the Entrust Website

Note: The Entrust Connect for Microsoft Azure App supports Microsoft Windows. Linux is not supported.

About Entrust Connect for Microsoft Azure

Azure Connect allows you to request and manage Entrust SSL Certificates in your Azure Key Vault.

When you connect the Entrust Certificate Services account to your Azure Key Vault using Azure Connect, you can store and manage your certificates directly within the Key Vault.

The Key Vault is also where the Public/Private keypair is generated, and where newly issued certificates will be installed.

What you can do from the Azure Connect user interface:

- View certificates
- Create a new SSL/TLS certificate
- Install an SSL/TLS certificate
- Reissue an SSL/TLS certificate
- Renew an SSL/TLS certificate
- Revoke an SSL/TLS certificate

The Entrust Connect for Microsoft Azure binaries are available at the following link:

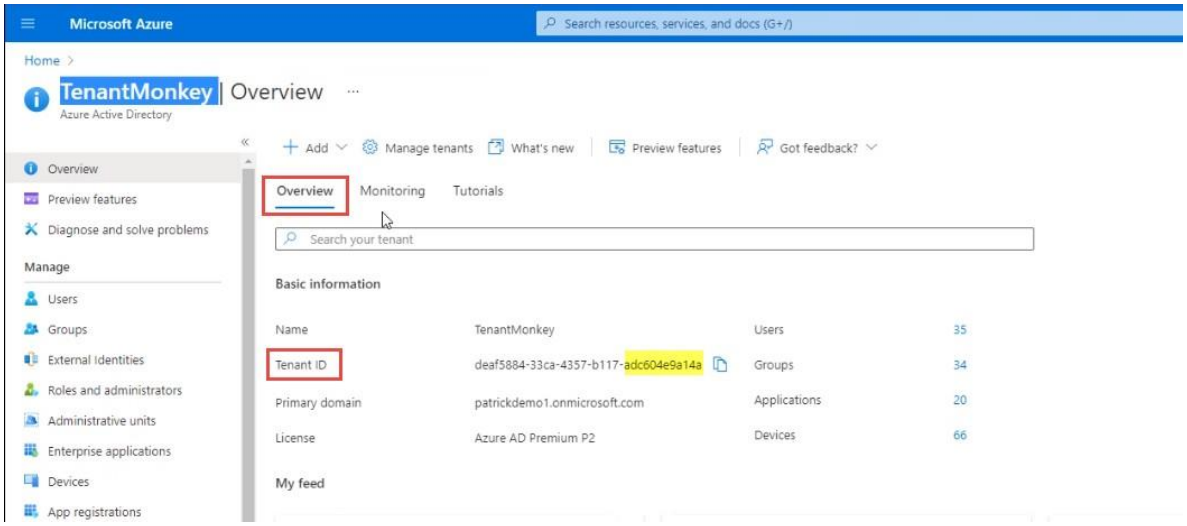
https://www.entrust.com/resources/tools/entrust-connect-microsoft-azure

Integrating with Microsoft Azure Key Vault

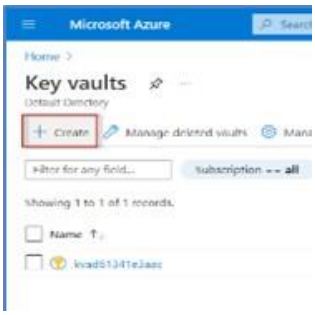
Follow these steps to set up an Azure Key Vault and integrate it with Entrust Connect for Azure.

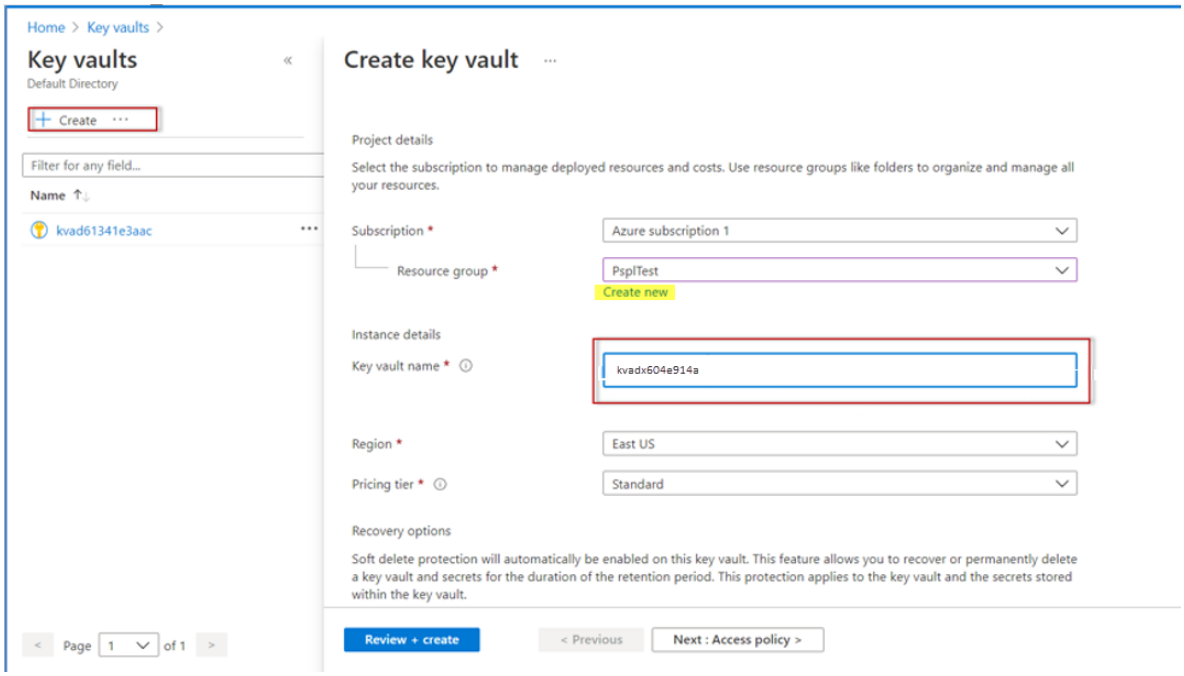
Step 1: Create a new Azure Key Vault

1. Log in to the Azure portal at https://portal.azure.com.
2. In the top search panel, search for `Microsoft Entra ID`.
3. In Microsoft Entra ID, on the **Overview** tab, locate the **Tenant ID**.



4. Copy and save the last 12 digits of the **Tenant ID**. You will need it in an upcoming step.
5. In the top search panel, search for `Key vaults`.
6. On the Key vaults page, click **Create**.





7. On the **Create key vault** screen, enter the following information:
   a. **Subscription**: Select a subscription.
   b. **Resource group**: Click **Create new** and enter a new Resource group name.
   c. **Key Vault name**: Create the name using `kv` plus the Tenant ID you copied earlier; for example, **`kv`**`adx604e914a. Do no include hyphen or space.`
   d. **Region**: Select your region.
   e. **Pricing Tier**: Select the appropriate pricing tier.
8. Click **Review and create** and complete creation of the key vault.

Step 2: Generate ECS REST API Credentials

Follow the steps below to generate REST API credentials from the Entrust Certificate Services (ECS) account.

1. Login to ECS account
2. Click **Administration**
3. Click **Advanced Settings**.
4. On the Advanced Settings page, click **API**
5. Click **Generate Credentials.** Note: You will need to have at least 1 active SSL certificate in your account. You will need to export this certificate into **.PFX** format for next steps.

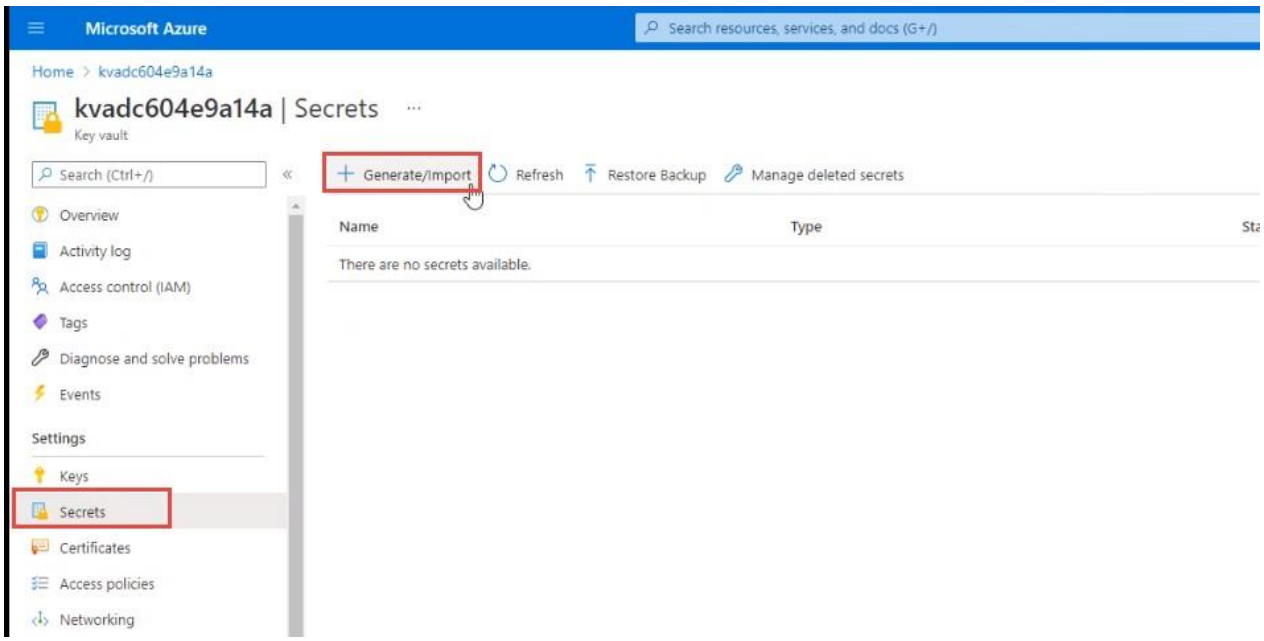
6. Store the newly generated credentials in a safe spot. Note: The API Key role must be set to **Super**

Step 3: Store the Entrust Certificate Services API user name and password

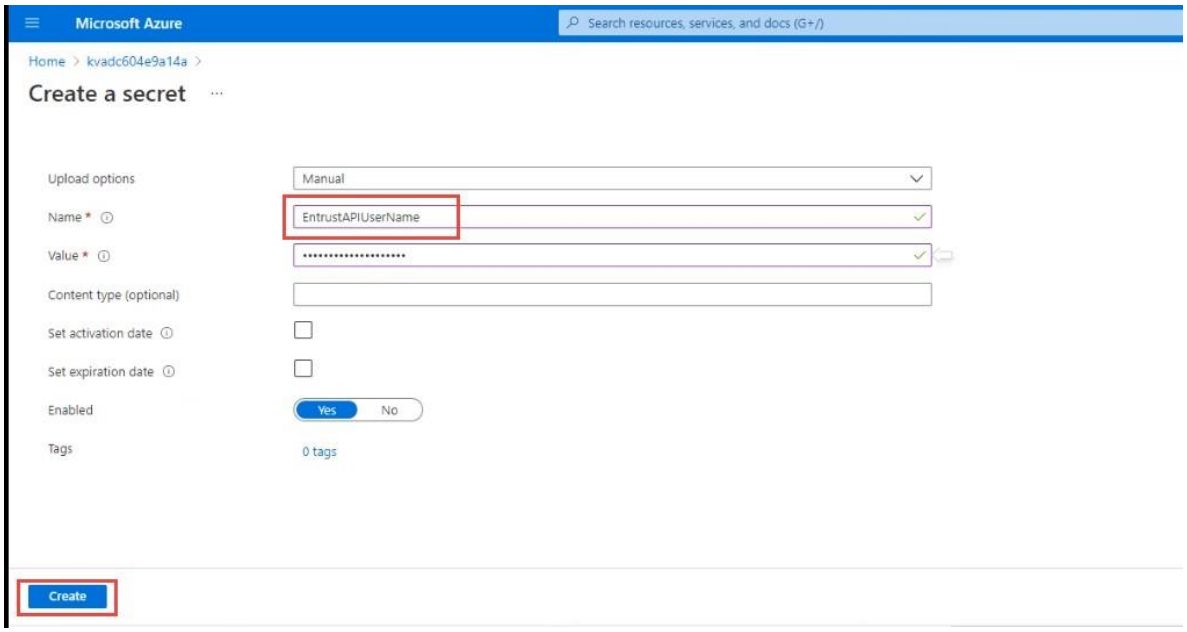
In this step, you will add the Secrets to the Key vault you just created.

For more information about the attributes of secrets, see https://docs.microsoft.com/en-us/azure/key-vault/secrets/quick-create-portal.

1. Navigate to your new Key vault.
2. In **Settings**, click **Secrets**.

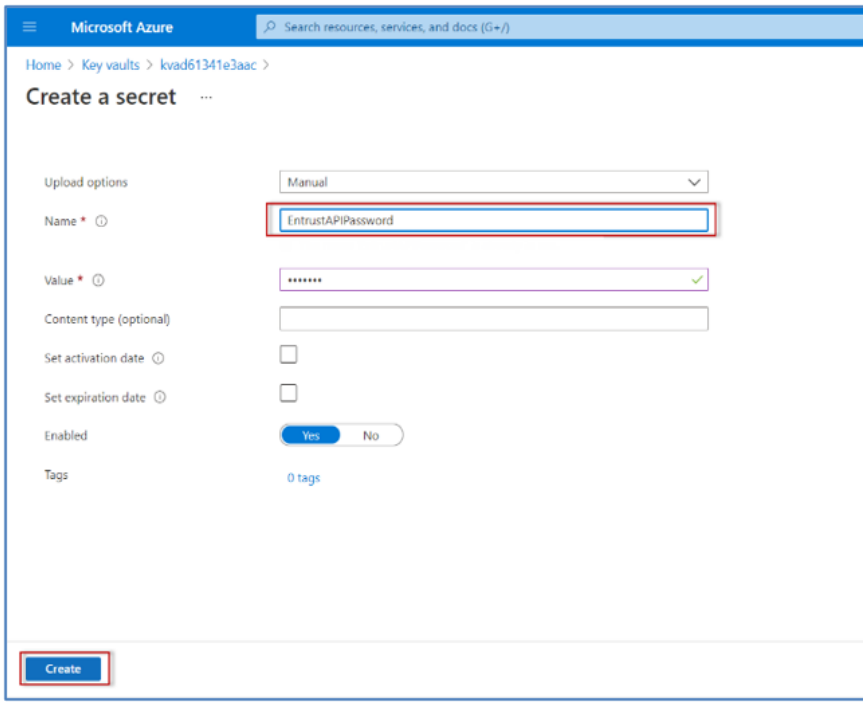


3. On the **Secrets** page, click **Generate/Import**.



4. On the **Create a secret** page, enter the following information.
   a. **Upload options**: Select **Manual**.
   b. **Name**: Enter `EntrustAPIUserName`. Type this value exactly as it appears here.
   c. **Value:** Enter the Certificate Services REST API user name that was generated when creating the API key in the Certificate Services Enterprise portal.
5. Click **Create**.

   You will see a confirmation message when the secret has been created successfully.
6. To store the Entrust REST API password, return to the **Secrets** page.
7. On the **Secrets** page, click **Generate/Import**.



8. On the **Create a secret** page, enter the following information.
   a. **Upload options**: Select **Manual**.
   b. **Name**: Enter `EntrustAPIPassword`.
   c. **Value:** Enter the Certificate Services REST API password that was generated when creating the API key in the Certificate Services Enterprise portal.
9. Click **Create**.

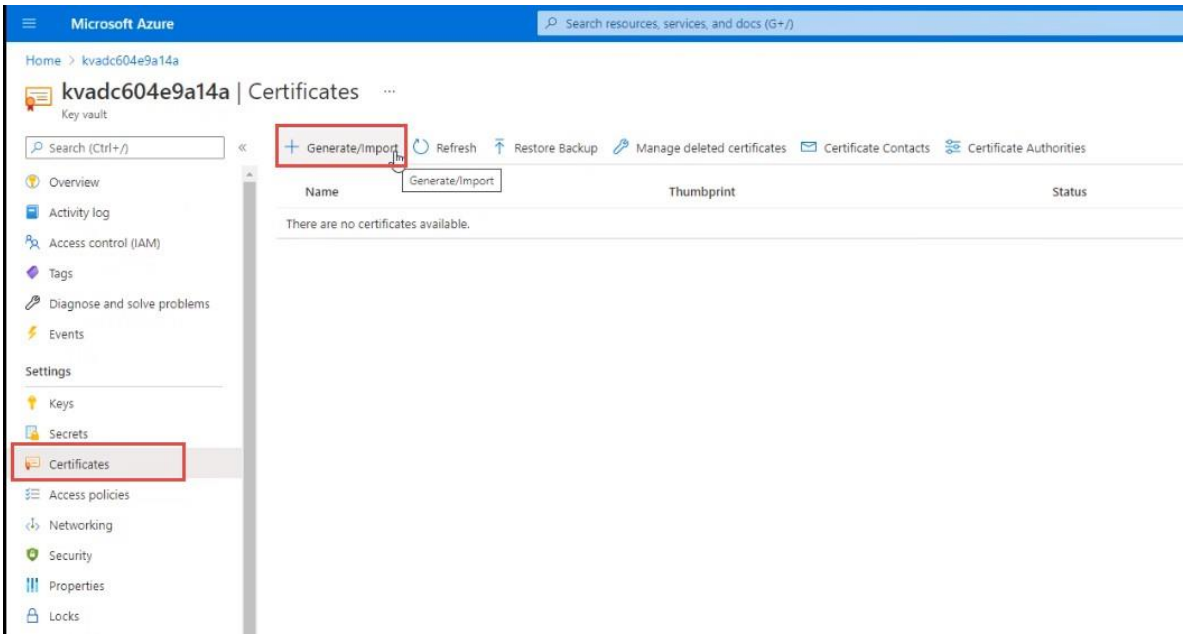
   You will see a confirmation message when the secret has been created successfully.

Step 4: Store the Entrust API certificate

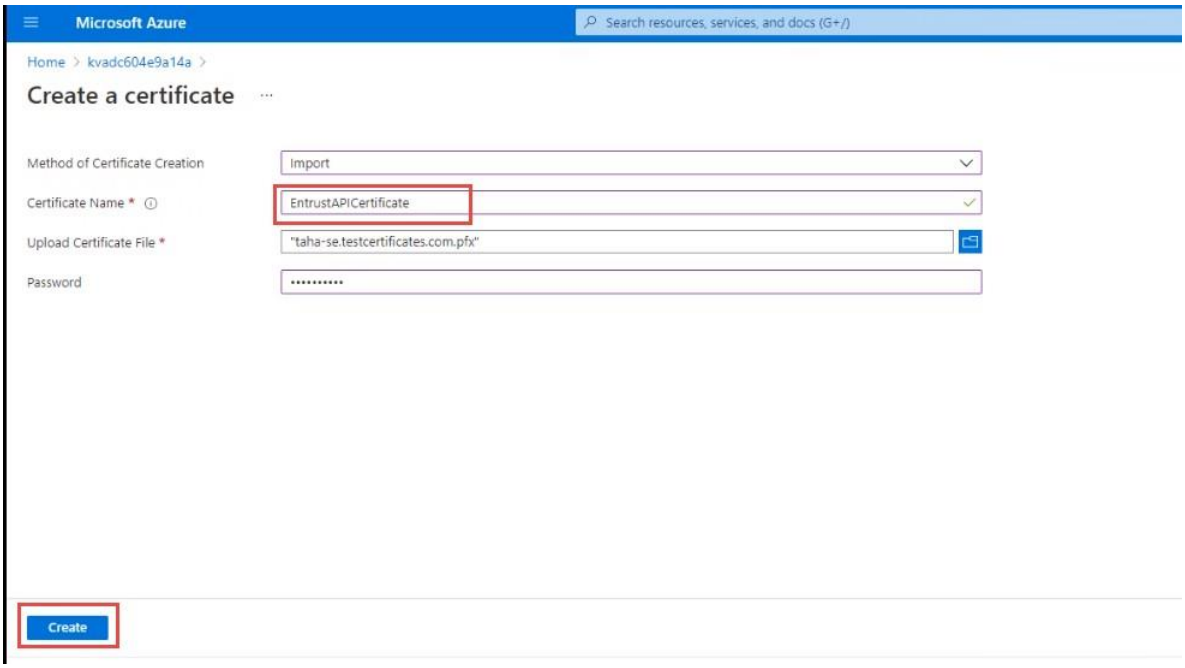
Before performing this procedure, you must have the Certificate Services REST API certificate (PFX file) stored in an accessible location. See step 2.

For more information about importing certificates, see https://docs.microsoft.com/en-us/azure/key-vault/certificates/tutorial-import-certificate.

1. Navigate to your key vault and select **Certificates**.



2. Click **Generate/Import**.

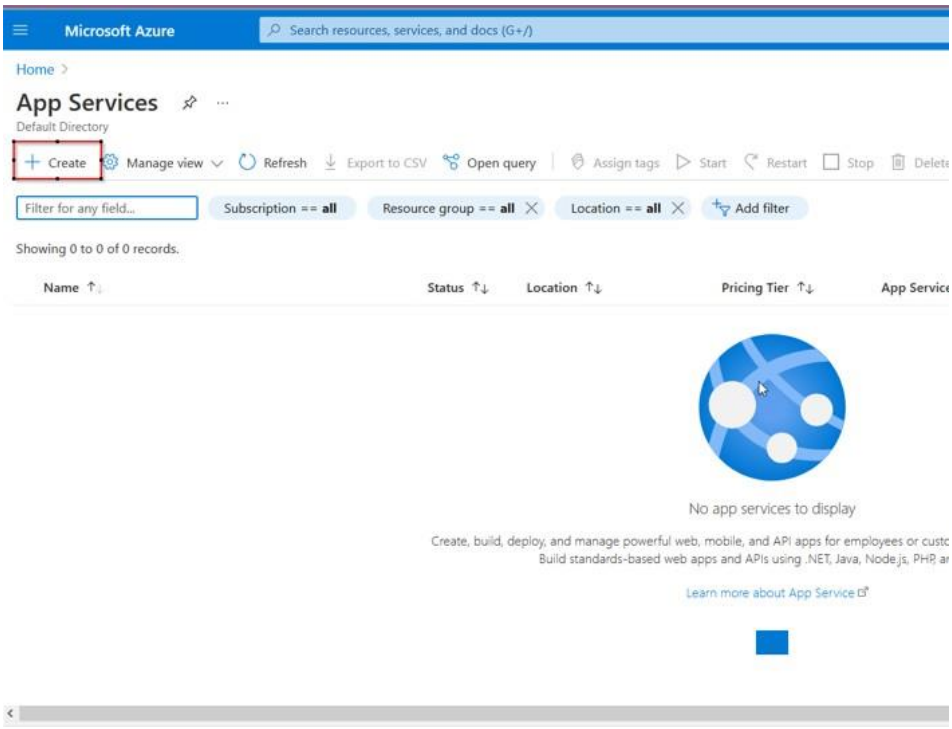


3. On the **Create a certificate** page, enter the following information.
    a. **Method of Certificate Creation**: Select **Import**.
    b. **Certificate Name**: Enter `EntrustAPICertificate`.
    c. **Upload Certificate File**: Select the PFX API certificate file.
    d. **Password**: If the certificate file is password-protected, enter the certificate password.
4. Click **Create**.

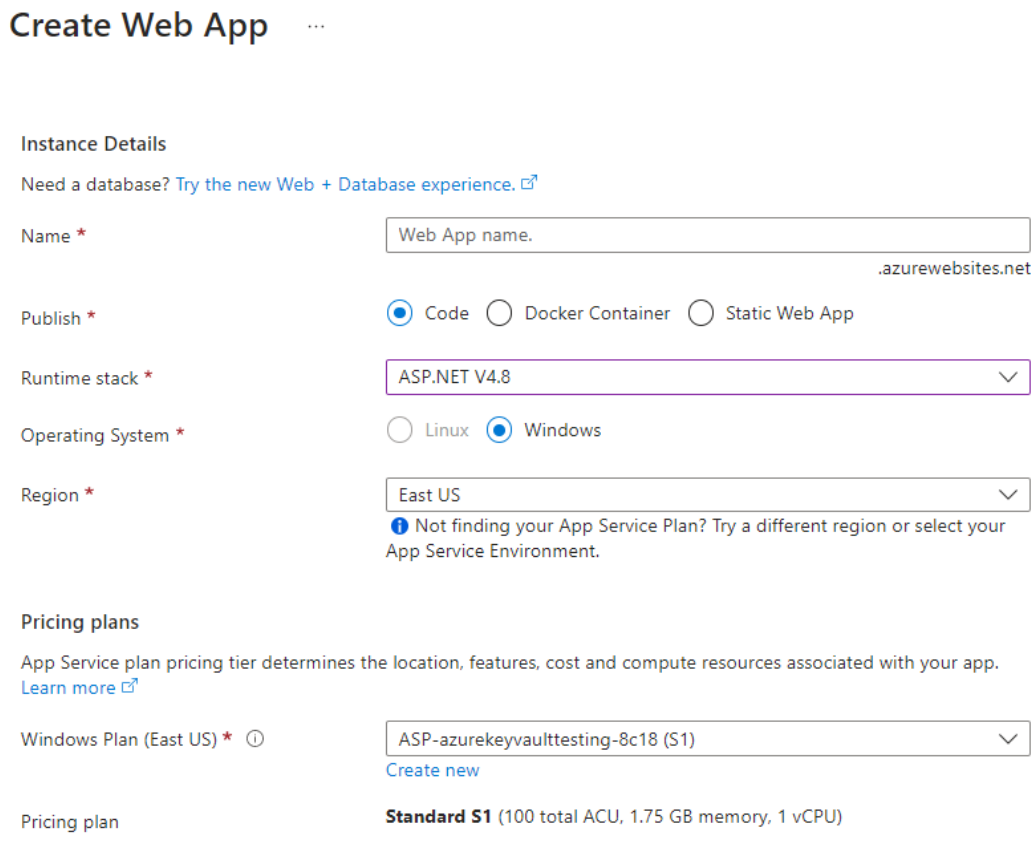
Step 5: Create the App Service

Note: Remember to follow these steps in the order they are given. The solution may not work properly if the steps are not done in the proper order.

1. In the Azure search panel, search for `App Services`.
2. On the **App Services** screen, click **Create**.



3. On the **Create Web App** screen, fill in the **Instance Details** and **App Service Plan**.



a. In **Runtime stack**, select **SP.NET V4.8**.

b. In **Operating System**, select **Windows**.

c. Select an appropriate **App Service Plan**.

Note: The Entrust Connect for Microsoft Azure App supports Microsoft Windows. Linux is not supported.

4. Click **Review and create** and complete creation of the **App Service**.
5. **Download** the Entrust Connect for Microsoft Azure App binaries from the following link:

   https://www.entrust.com/resources/tools/entrust-connect-microsoft-azure
6. Unzip the file containing the Entrust Connect for Microsoft Azure App binaries.

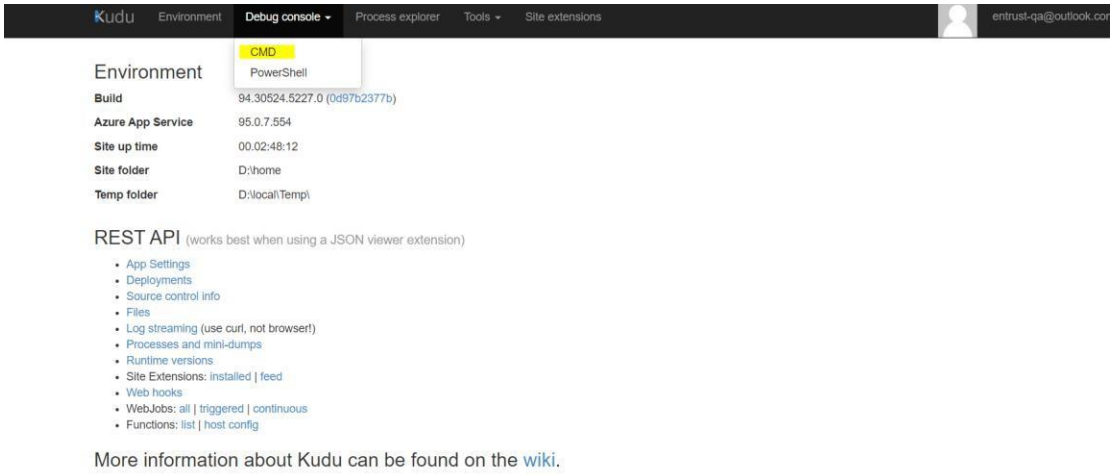
   Option 1: Upload the Connect for Microsoft Azure App Binaries to the App Services via the Microsoft Azure portal

   a. Click the App Service you just created.

   b. On the left panel, select **Advanced Tools**.

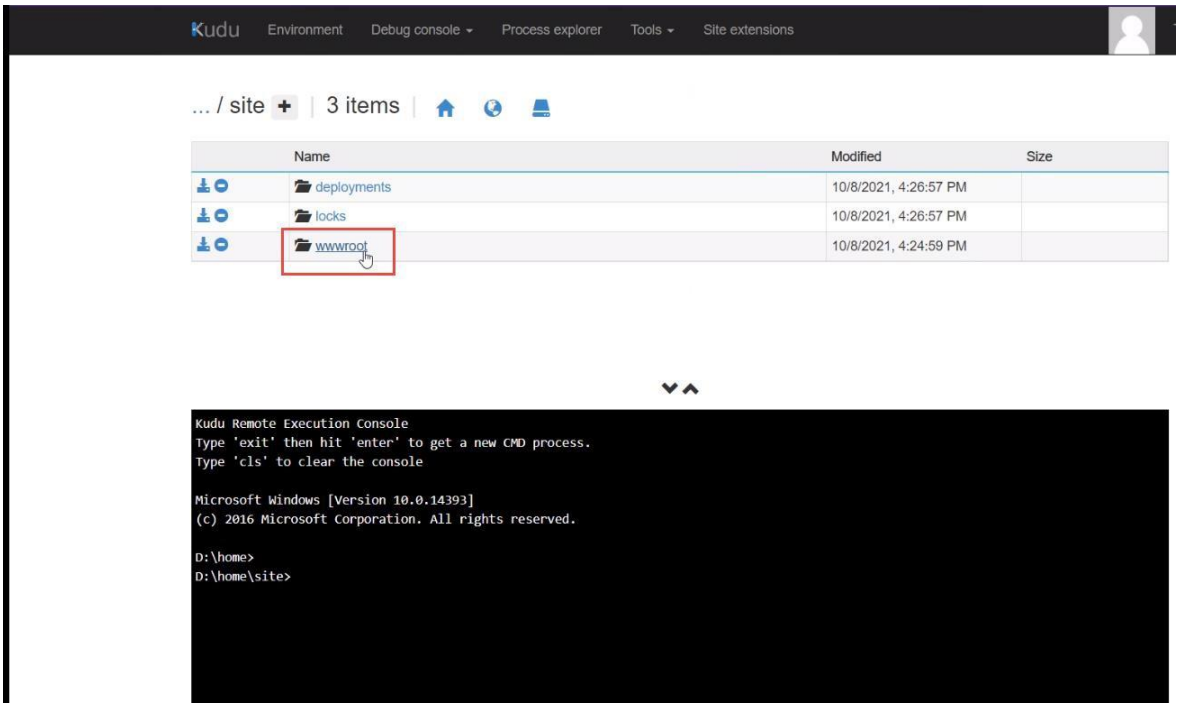
c. Click **Go**. You will be directed to log in using your Entrust credentials. A new page will open.

d. Click **Debug Console** > **CMD**.

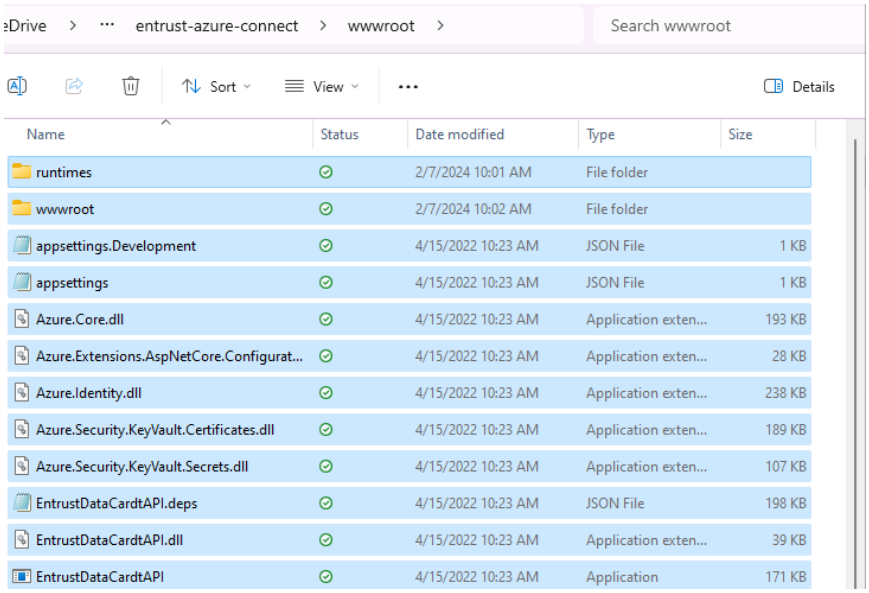


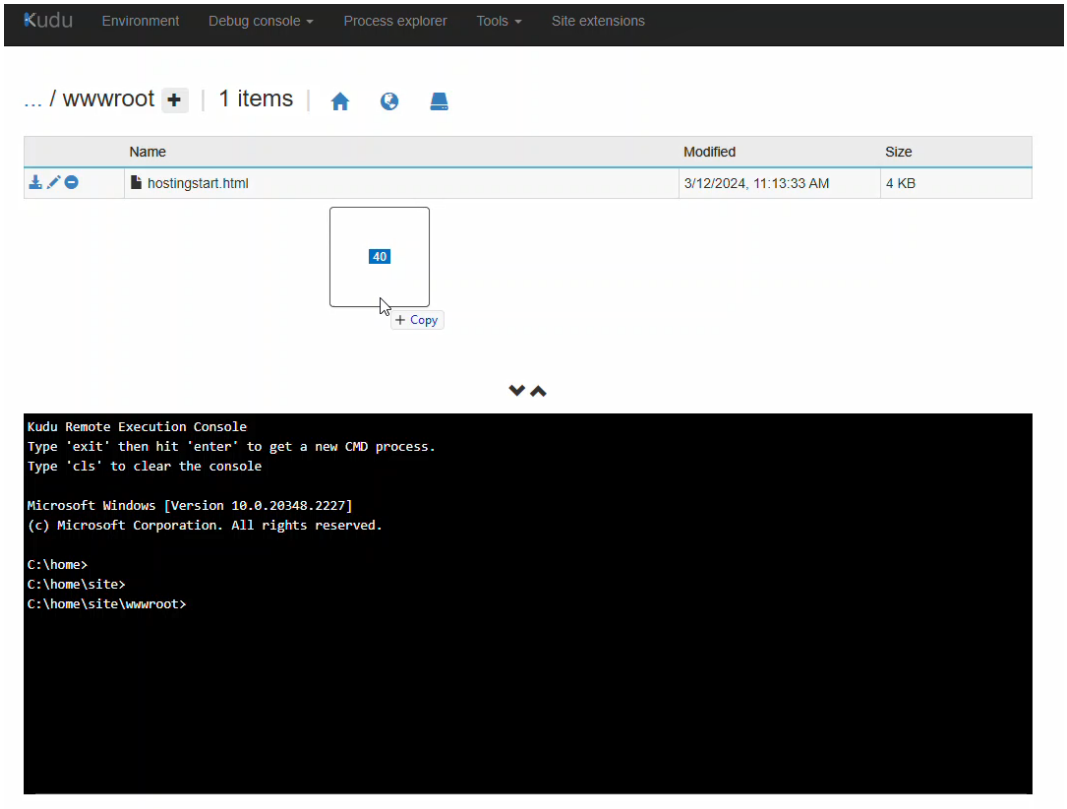
e. In the screen that appears, click the `site` folder.

f. Click the `wwwroot` folder.



g. Drag and drop the Connect for Microsoft Azure App binary files located into the `wwwroot` folder.





h. Go back to the Azure Portal.

i. Click **App Services**.

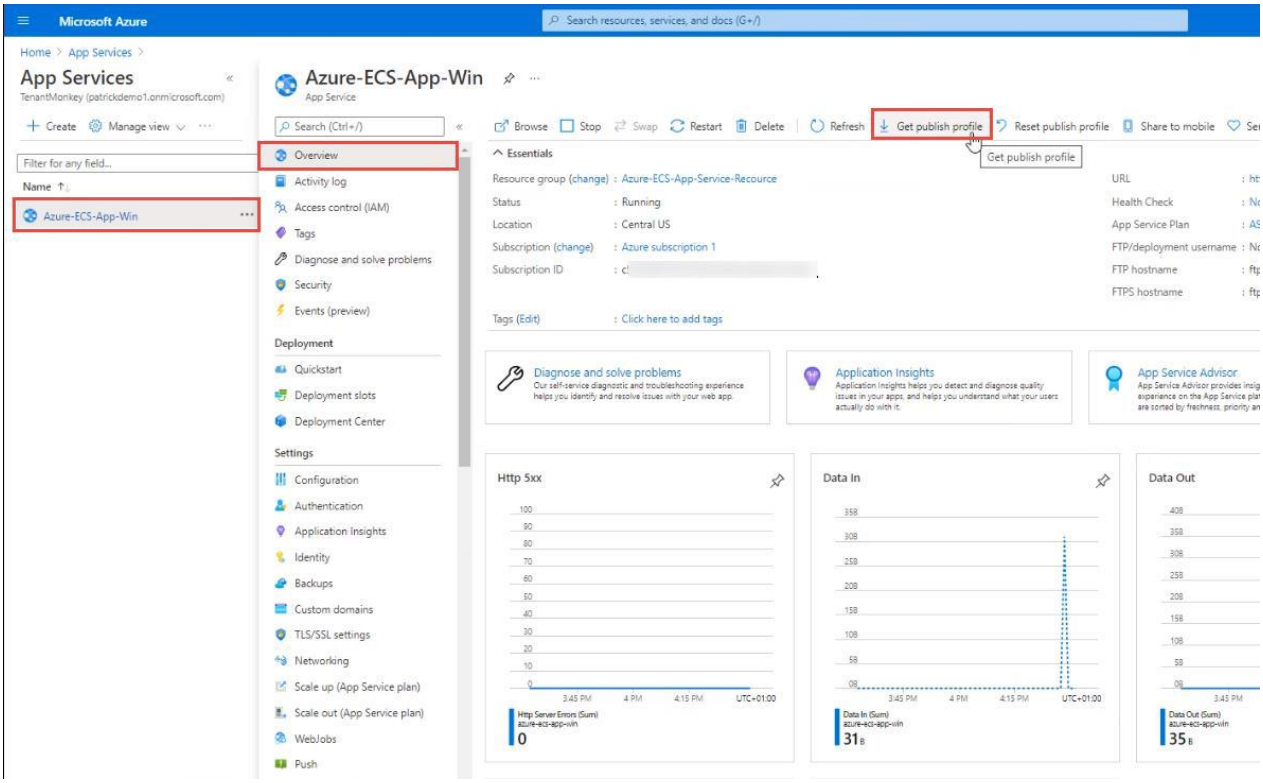
j. Click the name of the application you created.

k. Click the **Application URL** to verify that the application is up and running.

Option 2: Upload the Entrust Connect for Microsoft Azure App binaries to the App Services via Get publish profile

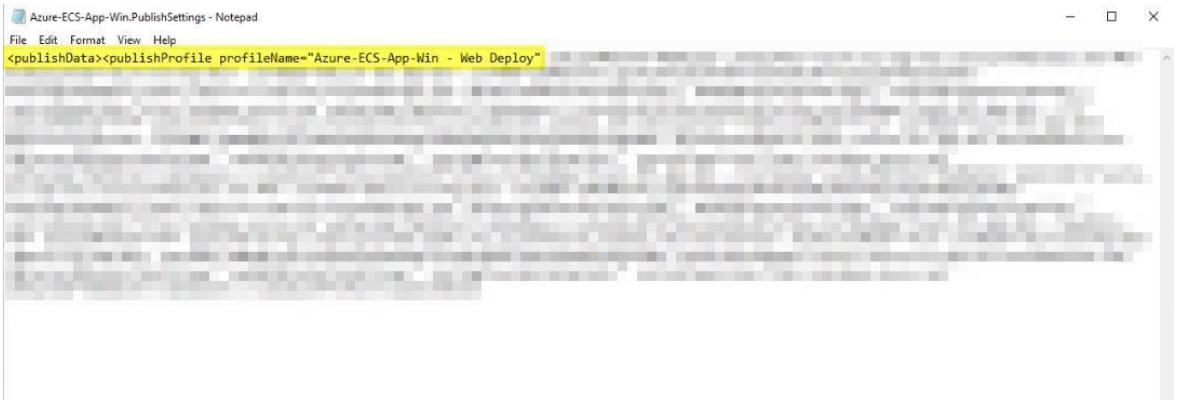
This option allows you to upload using any FTP client.

a. On the App Services screen, select the name of your new App Service.



b. Click **Get publish profile** to download the profile.

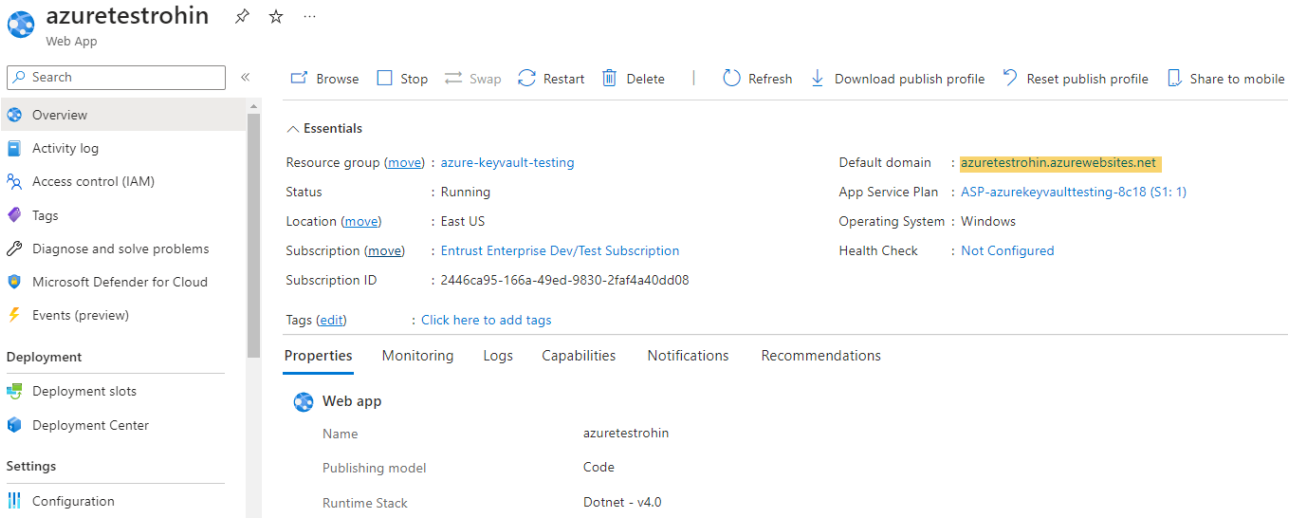
You will see the FTP credentials in the PublishSettings file. The Get publish profile will make it easier to FTP the Connect for Microsoft Azure App binaries to the App Services using any FTP Client.



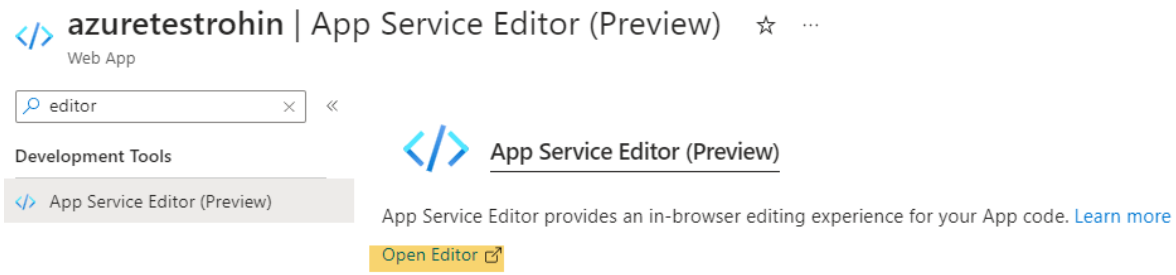
Step 6: Update Server URL

Note: This step will be eliminated in a future version of the app.

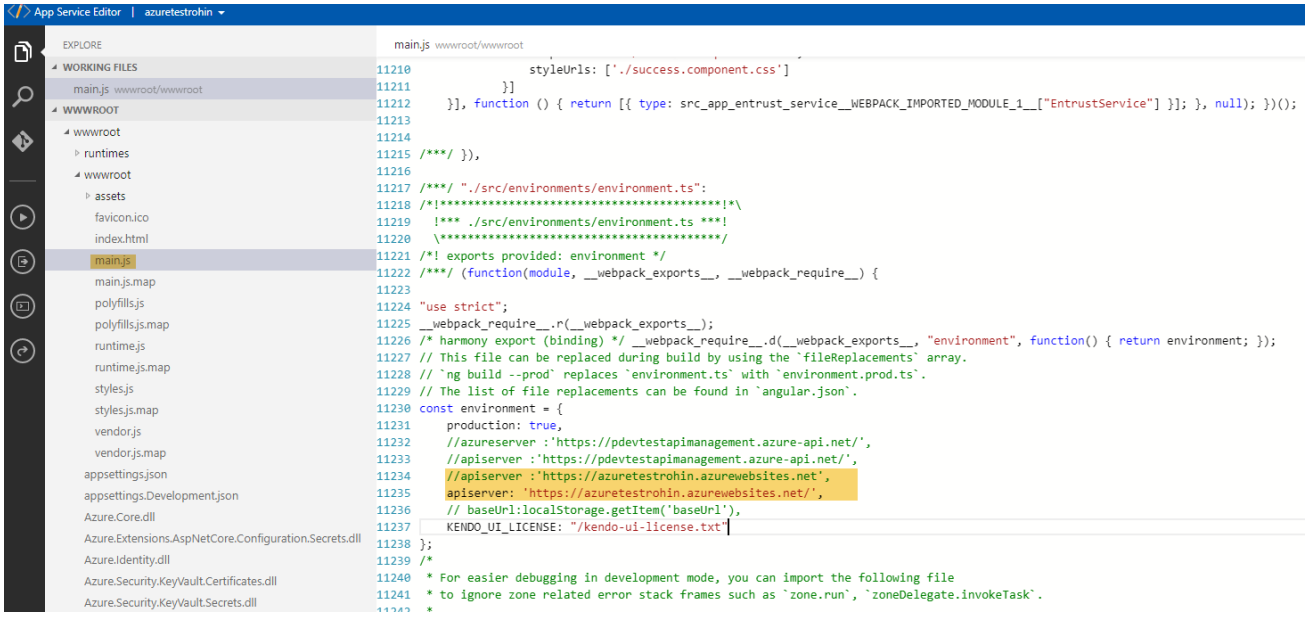
1. **Update Server URL**: Open your new app from *App Services*
2. Copy the *Default Domain* URL



3. Open *App Service Editor*



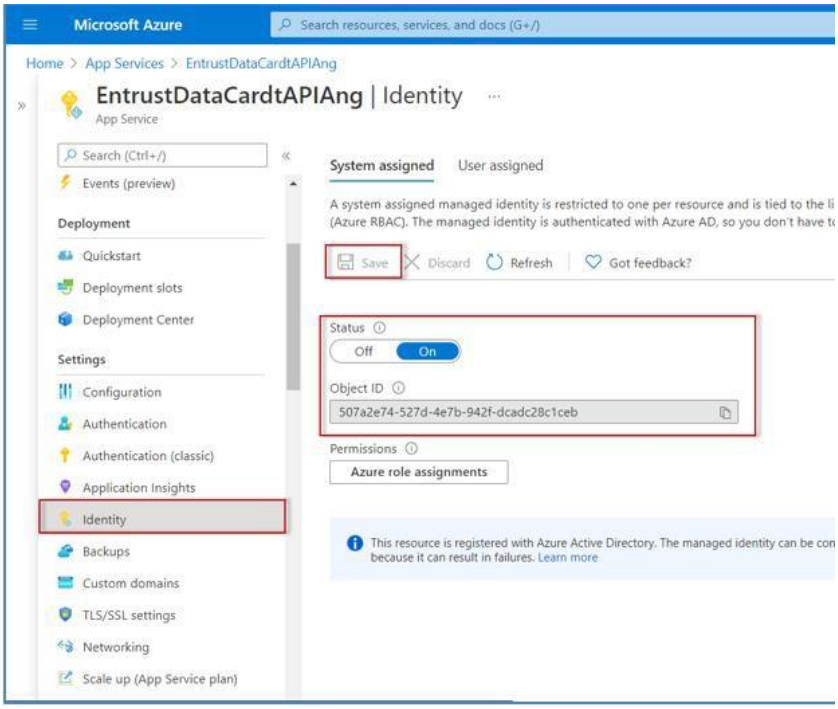
4. Open Main.js file and update lines 11234, 11235 with the *Default Domain* URL you copied in previous step.



Step 6: Add a system-assigned identity

Set up an Azure Service to create a managed identity.

1. In the Azure portal, in the top search panel, search for `App Services` and configure the new App Service that you created in **Step 4**.



2. In the existing App Service, click **Identity**.
3. On the **Identity** page, select the **System assigned** tab.
4. Click the **Status** switch to **On**.
5. In **Object ID**, copy and save the alphanumeric code. You will need this in the next step.
6. Click **Save**.

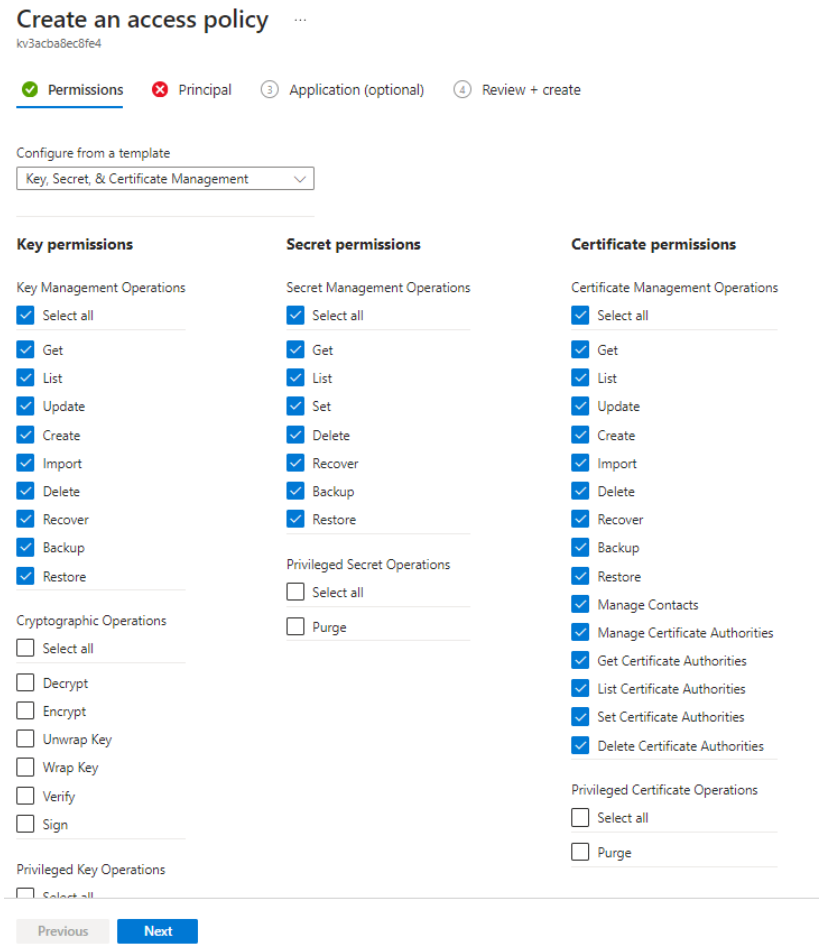
Step 7: Assign an access policy for the App Service

Assign an access policy for the App Service in your key vault.

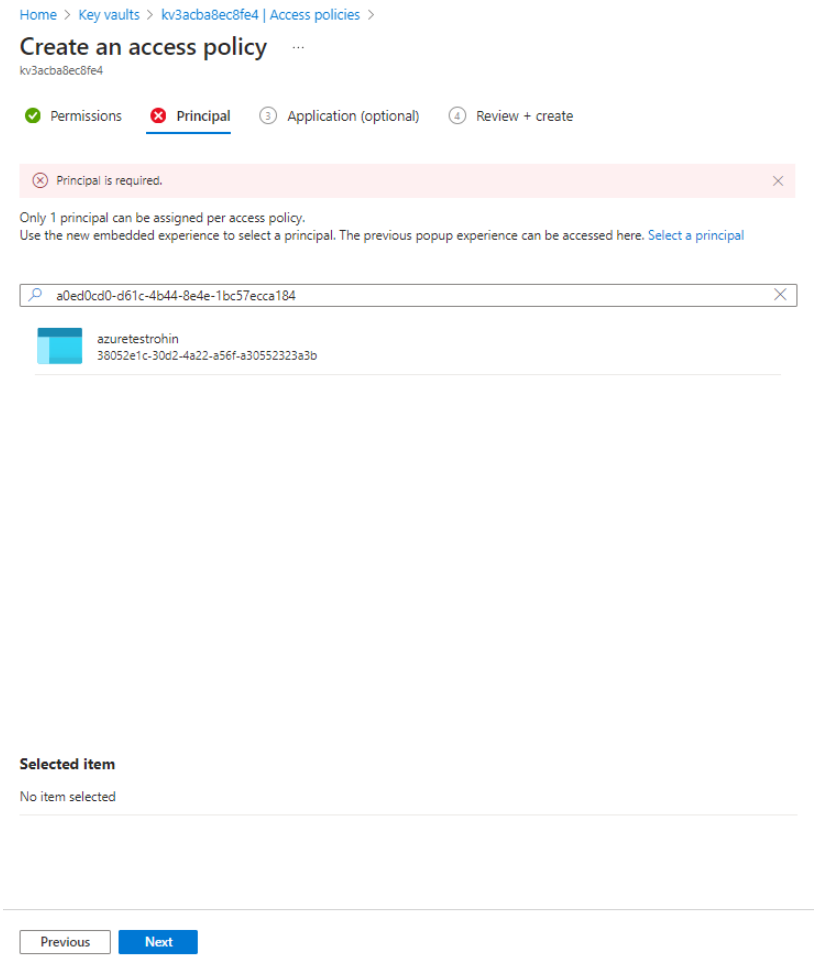
For more information on managed identities for App Services, see https://docs.microsoft.com/en-us/azure/key-vault/general/assign-access-policy-portal.

1. Navigate to your new key vault.

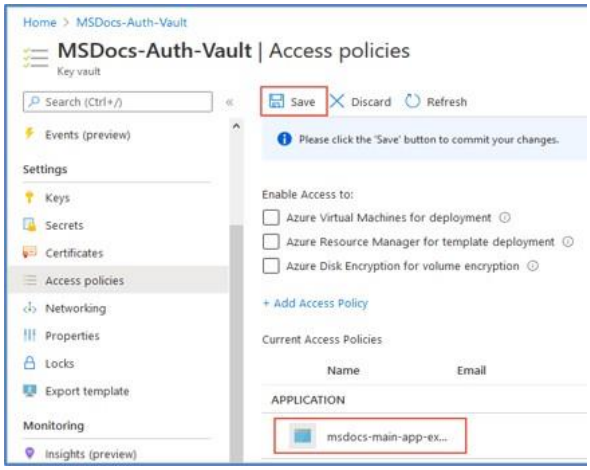
2. In **Settings**, select **Access policies**.
3. Click **Add Access Policy**.



4. On the **Add access policy** screen, select the following:
    a. **Configure from template (optional)**: Select **Key, Secret, & Certificate Management**.
    b. **Select Principal**: Select **None selected**. The **Principal** pane appears.
    c. Paste in the **Object ID** you copied in the last step.
    d. Click **Select**.



5. Under **Add access policy** on the left, click **Add**.



6. On the **Access policies** page for your key vault, confirm that the new access policy

appears in the **Current Access Policies** list.

Note: The new access policy is not applied until you confirm and **Save** on the **Access polices** page.

7. Click **Save**.

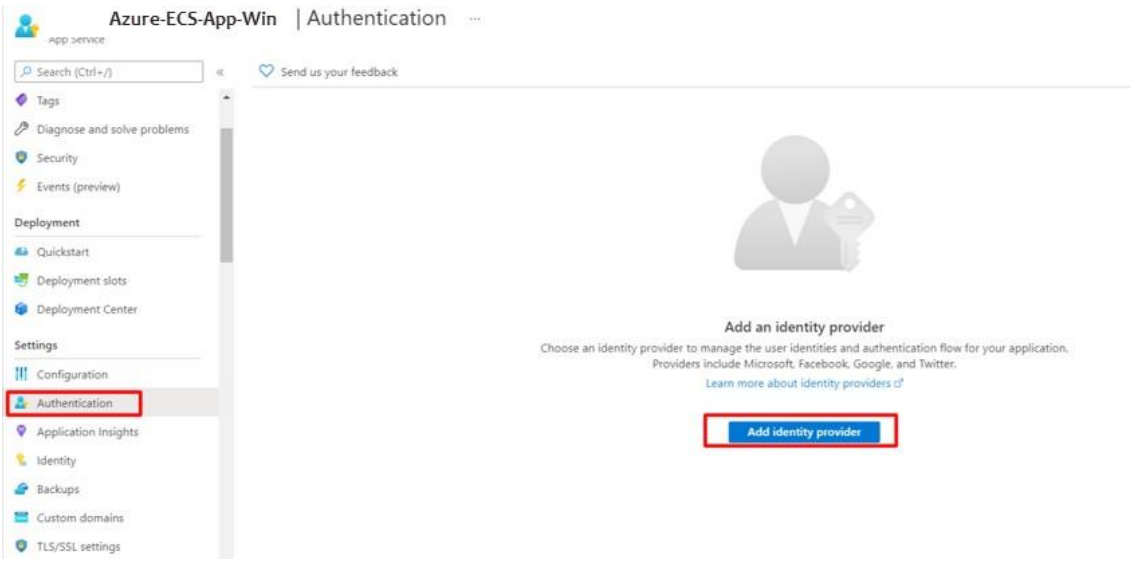
After deploying the Connect for Microsoft Azure App Service, Microsoft, by default, will publish a public facing URL; e.g., https://azure-ecs-app-win.azurewebsites.net

To avoid unauthorized users from accessing the application, set up an identity provider by following the steps in the next section.

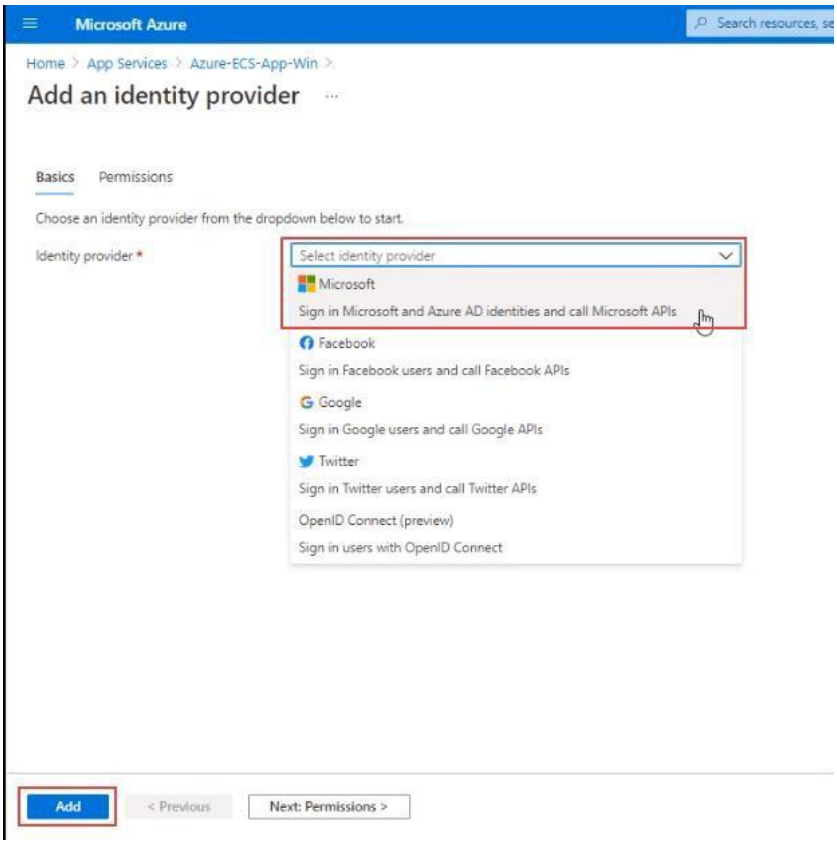
Step 8: Enable Azure App Service Authentication

The goal of this step is to provide your users with an authentication process to access the Entrust Connect for Microsoft Azure App.

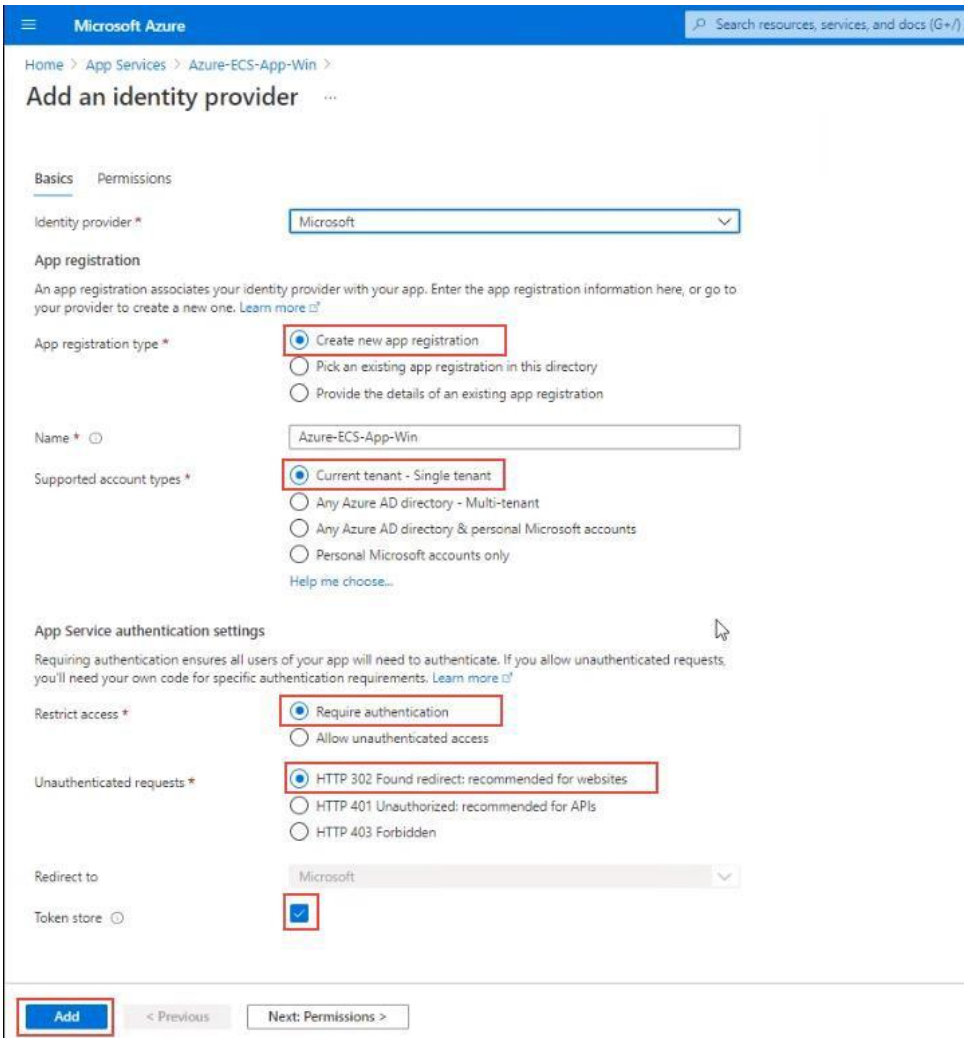
1. In Azure App Services, click your application.
2. In the left panel click **Authentication**.



3. Click **Add identity provider**.
4. In the screen that appears, select **Microsoft**.

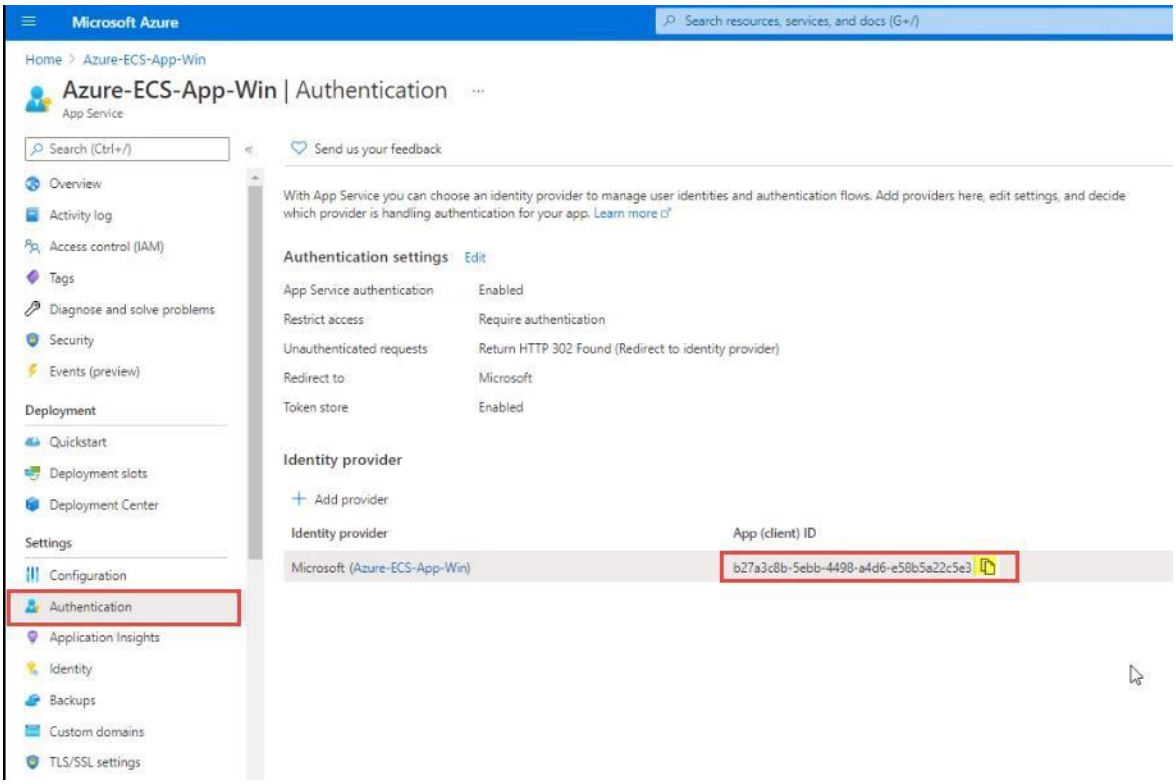


A new screen appears.

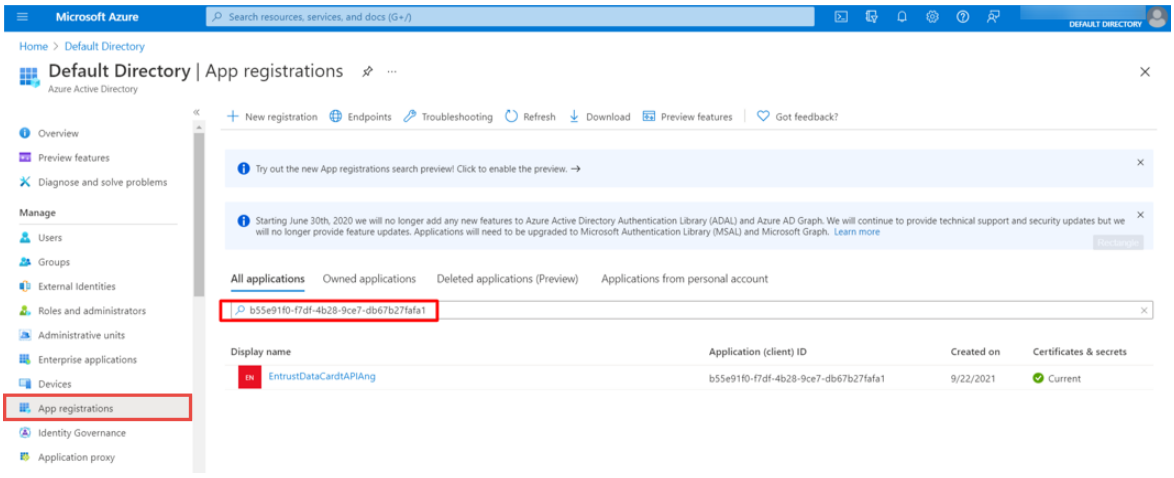


a. In **App registration type**, select **Create new app registration**.
b. In **Supported account types**, select **Current tenant**.
c. In **Restrict access**, select **Require Authentication**.
d. In **Unauthenticated requests**, select **HTTP 302**.
e. Select the **Token store** checkbox.

5. Click **Add**.

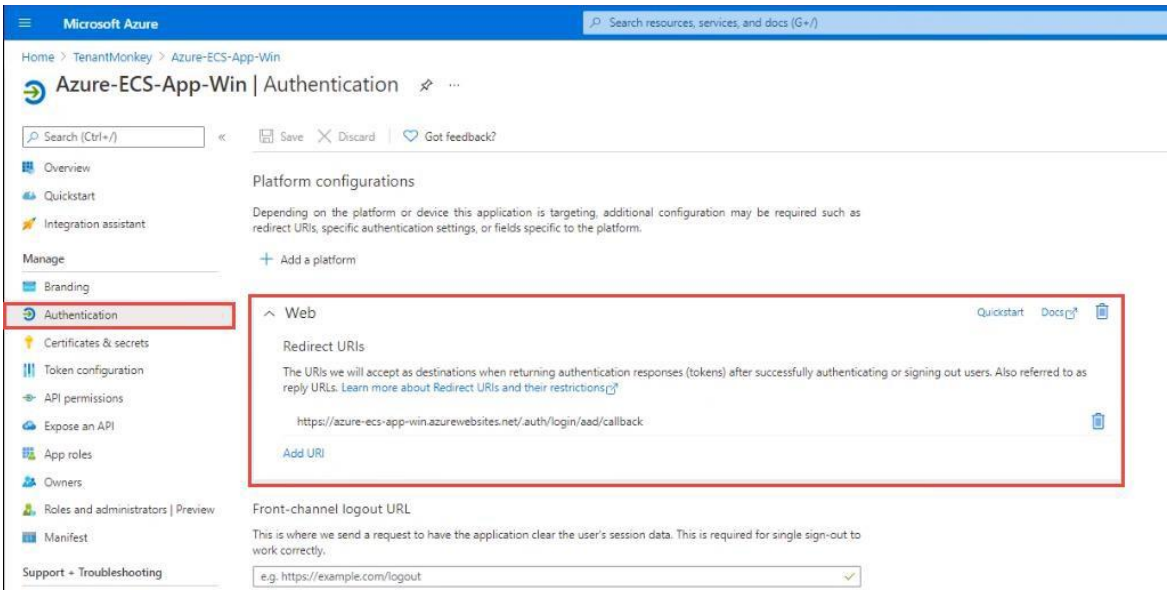


6. In the screen that appears, copy the **App (client) ID** of the new Identity.
7. Search for **Microsoft Entra ID.**
8. In the left panel, select **App registrations.**



9. In the search box, paste the **App (client) ID** you copied in the previous step.

10. Redirect URIs are generated.



Troubleshooting

This section lists problems or error messages you might encounter during or after the Azure integration, along with advice for their resolution.

"There is no API user role configured with the Entrust Certificate Services account."

Cause of the problem:

You will see this error message if the user credential for the ECS REST API has been deleted but is still configured in the Entrust Connect for Microsoft Azure Application.

How to fix the problem:

Update the user credential for the ECS REST API in the Connect for Microsoft Azure Application to match the credentials in the Entrust Certificate Services account.

"The API user role within the Entrust Certificate Services account is not configured correctly."

Cause of the problem:

You will see this error message if the API user role in the Entrust Certificate Services account is not correctly configured.

How to fix the problem:

Ensure that the API user role is correctly configured as follows:

1. An active and issued TLS/SSL client certificate is bound to the API.
2. **Access Permission** is set to **Super**.
3. **Auto Approve** is enabled.