

# 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.

Entrust and the Hexagon Logo are trademarks, registered trademarks and/or services marks of Entrust Corporation in the U.S. and/or other countries. All other brand or product names are the property of their respective owners. Because we are continuously improving our products and services, Entrust Corporation reserves the right to change specifications without prior notice. Entrust is an equal opportunity employer.

© 2024 Entrust. All rights reserved.

## Table of contents

<b>Revision, audience, and guide information .....</b>	<b>4</b>
<i>Revisions .....</i>	<i>4</i>
<i>Audience.....</i>	<i>4</i>
<i>Viewing this guide.....</i>	<i>4</i>
<i>Prerequisites .....</i>	<i>4</i>
<b>About Entrust Connect for Microsoft Azure .....</b>	<b>5</b>
<b>Integrating with Microsoft Azure Key Vault.....</b>	<b>6</b>
<i>Step 1: Create a new Azure Key Vault .....</i>	<i>6</i>
<i>Step 2: Generate ECS REST API Credentials .....</i>	<i>7</i>
<i>Step 3: Store the Entrust Certificate Services API user name and password .....</i>	<i>8</i>
<i>Step 4: Store the Entrust API certificate.....</i>	<i>10</i>
<i>Step 5: Create the App Service.....</i>	<i>12</i>
<i>Step 6: Update Server URL.....</i>	<i>18</i>
<i>Step 6: Add a system-assigned identity.....</i>	<i>19</i>
<i>Step 7: Assign an access policy for the App Service .....</i>	<i>20</i>
<i>Step 8: Enable Azure App Service Authentication .....</i>	<i>24</i>
<b>Troubleshooting .....</b>	<b>29</b>
<i>“There is no API user role configured with the Entrust Certificate Services account.” .....</i>	<i>29</i>
<i>“The API user role within the Entrust Certificate Services account is not configured correctly.” .....</i>	<i>29</i>

# 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:

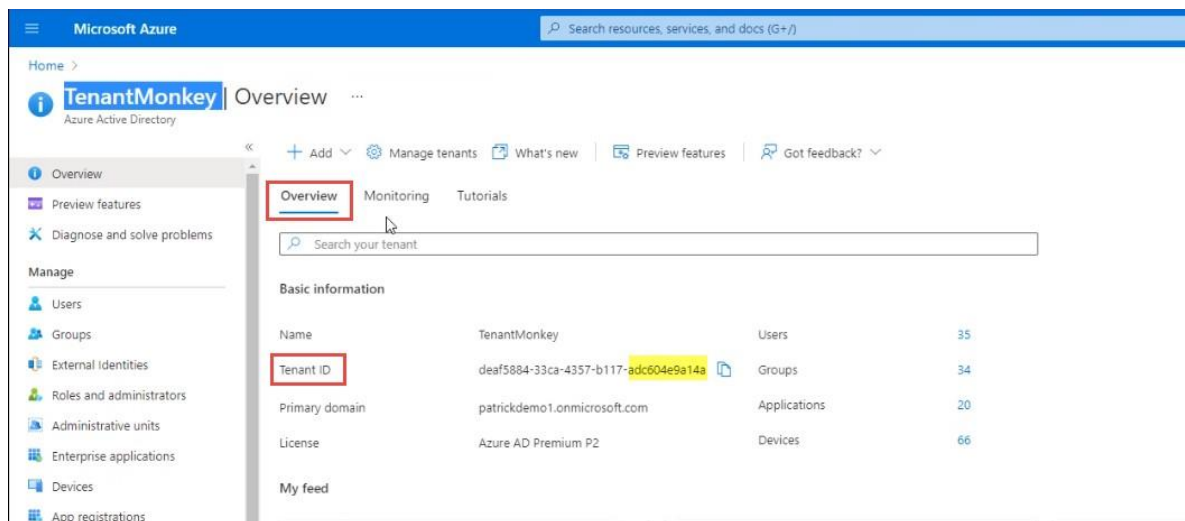
<https://www.entrust.com/resources/tools/entrust-connect-microsoft-azure>

# Integrating with Microsoft Azure Key Vault

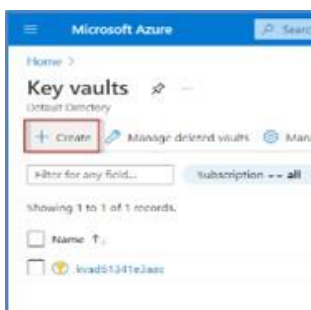
Follow these steps to set up an Azure Key Vault and integrate it with Entrust Connect for Azure.

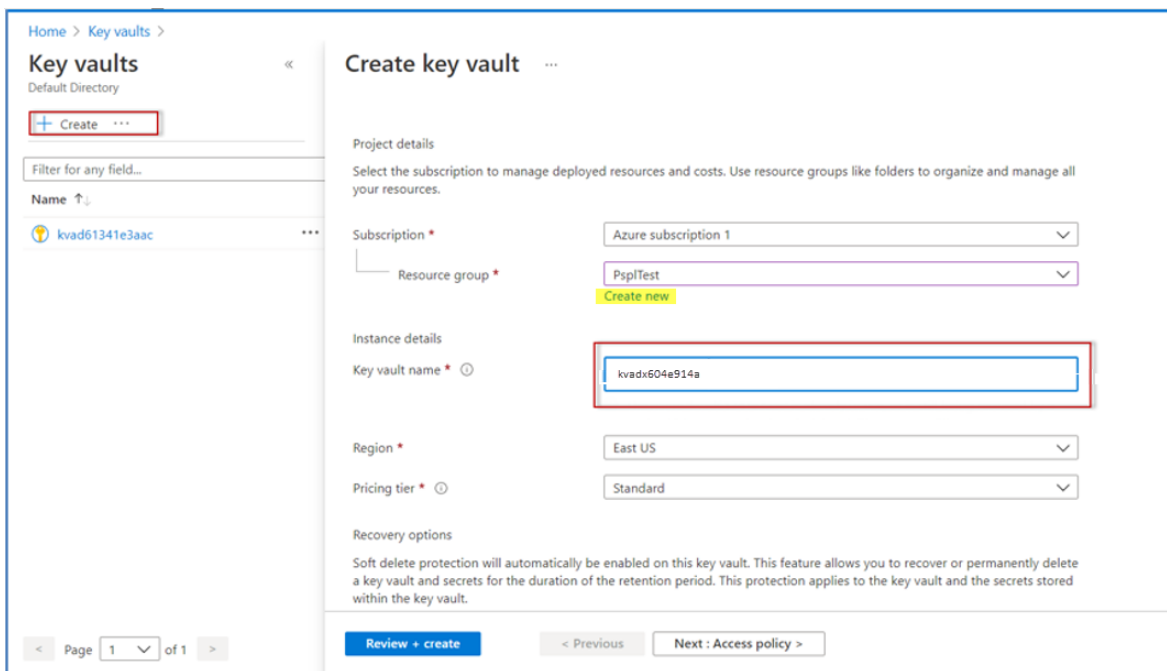
## Step 1: Create a new Azure Key Vault

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, `kvadx604e914a`. Do not 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.

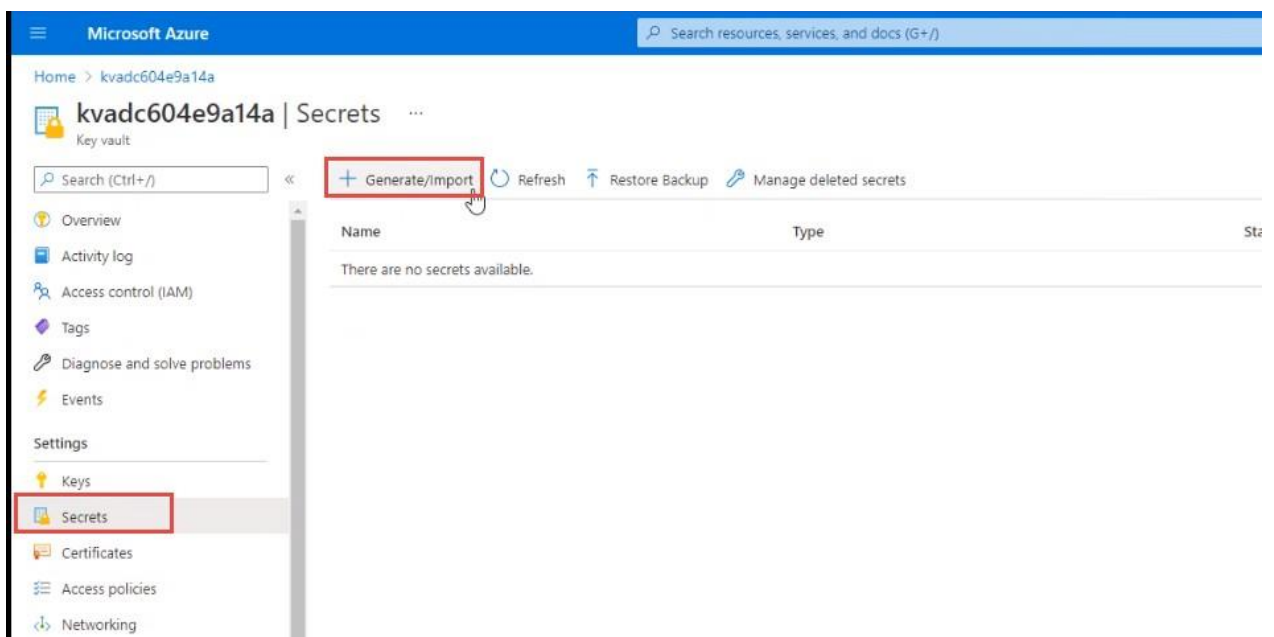
6. Store the newly generated credentials in a safe spot. Note: The API Key role must be set to **Super**

### 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**.



The screenshot shows the 'Create a secret' page in the Microsoft Azure portal. The page title is 'Create a secret' and the breadcrumb is 'Home > kvadc604e9a14a >'. The form includes the following fields and options:

- Upload options:** A dropdown menu set to 'Manual'.
- Name \*:** A text input field containing 'EntrustAPIUserName'.
- Value \*:** A text input field containing a series of dots, indicating a masked value.
- Content type (optional):** An empty text input field.
- Set activation date:** A checkbox that is unchecked.
- Set expiration date:** A checkbox that is unchecked.
- Enabled:** A toggle switch set to 'Yes'.
- Tags:** A label indicating '0 tags'.

A blue 'Create' button is located at the bottom left of the form.

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**.

The screenshot shows the 'Create a secret' page in the Microsoft Azure portal. The breadcrumb navigation is 'Home > Key vaults > kvad61341e3aac >'. The page title is 'Create a secret'. The form includes the following fields and options:

- Upload options:** A dropdown menu set to 'Manual'.
- Name:** A text input field containing 'EntrustAPIPassword'.
- Value:** A text input field with masked characters (asterisks) and a green checkmark on the right.
- Content type (optional):** An empty text input field.
- Set activation date:** An unchecked checkbox.
- Set expiration date:** An unchecked checkbox.
- Enabled:** A toggle switch set to 'Yes'.
- Tags:** A label indicating '0 tags'.

A 'Create' button is located at the bottom left of the form.

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**.

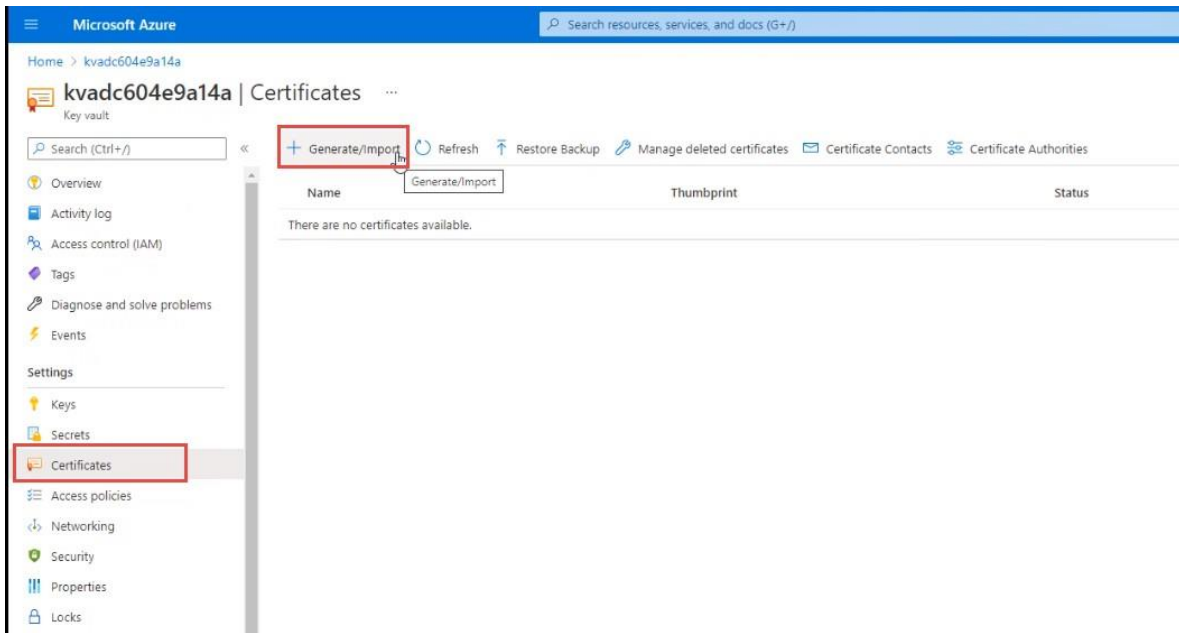
You will see a confirmation message when the secret has been created successfully.

## 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**.

Microsoft Azure

Home > kvadc604e9a14a >

### Create a certificate

Method of Certificate Creation: Import

Certificate Name \*

Upload Certificate File \*

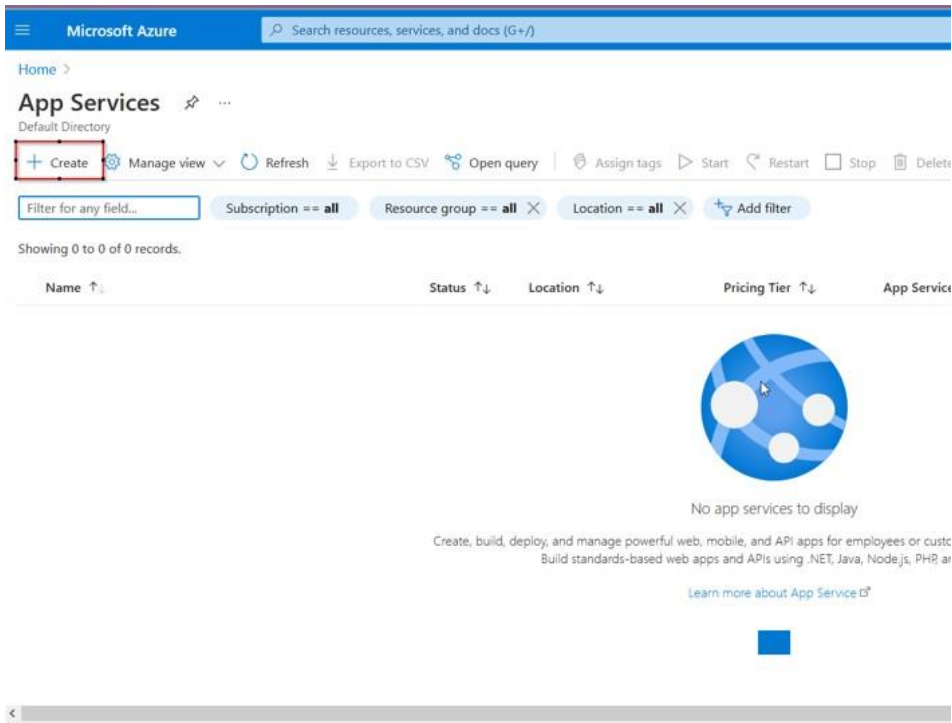
Password: .....

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**.

## Create Web App ...

### Instance Details

Need a database? [Try the new Web + Database experience.](#)

Name \*  .azurewebsites.net

Publish \*  Code  Docker Container  Static Web App

Runtime stack \*

Operating System \*  Linux  Windows

Region \*    
 ⓘ Not finding your App Service Plan? Try a different region or select your App Service Environment.

### Pricing plans

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. [Learn more](#)

Windows Plan (East US) \* ⓘ    
 [Create new](#)

Pricing plan **Standard S1** (100 total ACU, 1.75 GB memory, 1 vCPU)

- 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:

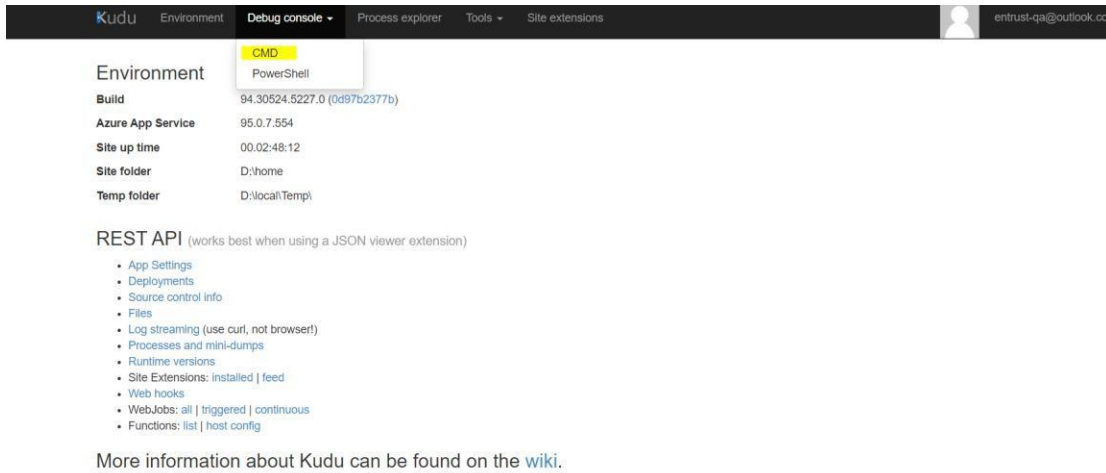
<https://www.entrust.com/resources/tools/entrust-connect-microsoft-azure>

6. Unzip the file containing the Entrust Connect for Microsoft Azure App binaries.

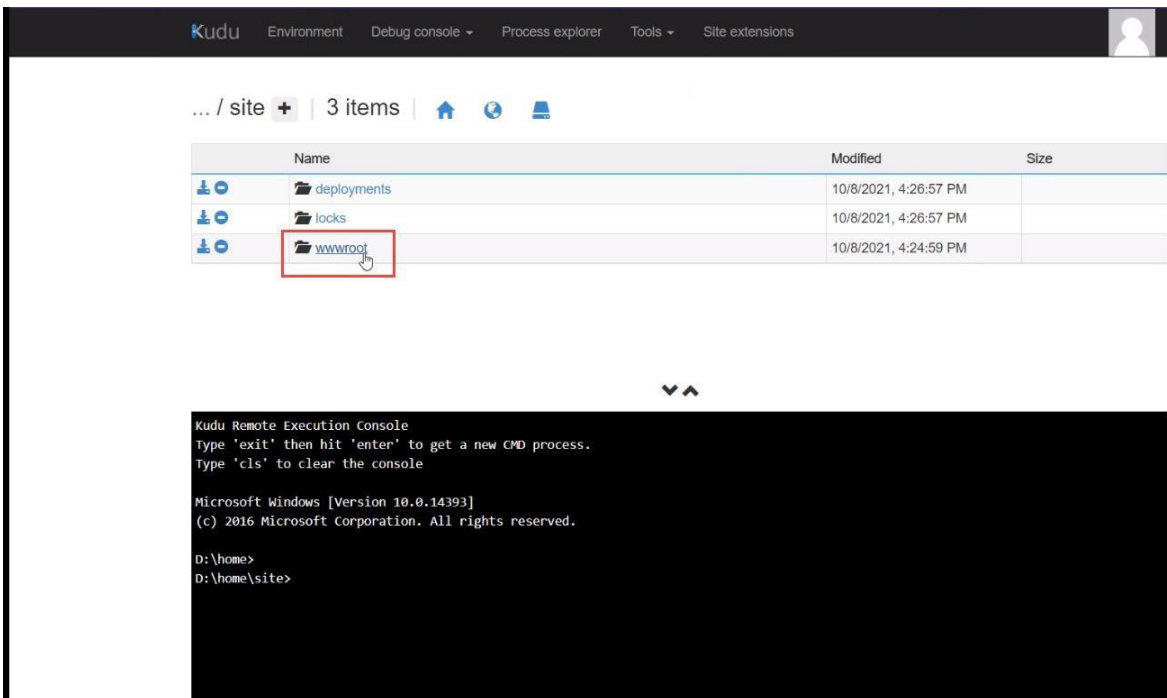
**Option 1: Upload the Connect for Microsoft Azure App Binaries to the App Services via the Microsoft Azure portal**

- 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.

The screenshot shows a file explorer window for the 'wwwroot' folder. The table below lists the files and folders:

Name	Status	Date modified	Type	Size
runtimes	🟢	2/7/2024 10:01 AM	File folder	
wwwroot	🟢	2/7/2024 10:02 AM	File folder	
appsettings.Development	🟢	4/15/2022 10:23 AM	JSON File	1 KB
appsettings	🟢	4/15/2022 10:23 AM	JSON File	1 KB
Azure.Core.dll	🟢	4/15/2022 10:23 AM	Application exten...	193 KB
Azure.Extensions.AspNetCore.Configurat...	🟢	4/15/2022 10:23 AM	Application exten...	28 KB
Azure.Identity.dll	🟢	4/15/2022 10:23 AM	Application exten...	238 KB
Azure.Security.KeyVault.Certificates.dll	🟢	4/15/2022 10:23 AM	Application exten...	189 KB
Azure.Security.KeyVault.Secrets.dll	🟢	4/15/2022 10:23 AM	Application exten...	107 KB
EntrustDataCardAPI.deps	🟢	4/15/2022 10:23 AM	JSON File	198 KB
EntrustDataCardAPI.dll	🟢	4/15/2022 10:23 AM	Application exten...	39 KB
EntrustDataCardAPI	🟢	4/15/2022 10:23 AM	Application	171 KB

Below the file explorer, the Kudu interface shows the 'wwwroot' folder with one item, 'hostingstart.html', and a terminal window showing the current directory path:

```

Kudu Remote Execution Console
Type 'exit' then hit 'enter' to get a new CMD process.
Type 'cls' to clear the console

Microsoft Windows [Version 10.0.20348.2227]
(c) Microsoft Corporation. All rights reserved.

C:\home>
C:\home\site>
C:\home\site\wwwroot>

```

- h. Go back to the Azure Portal.
- i. Click **App Services**.

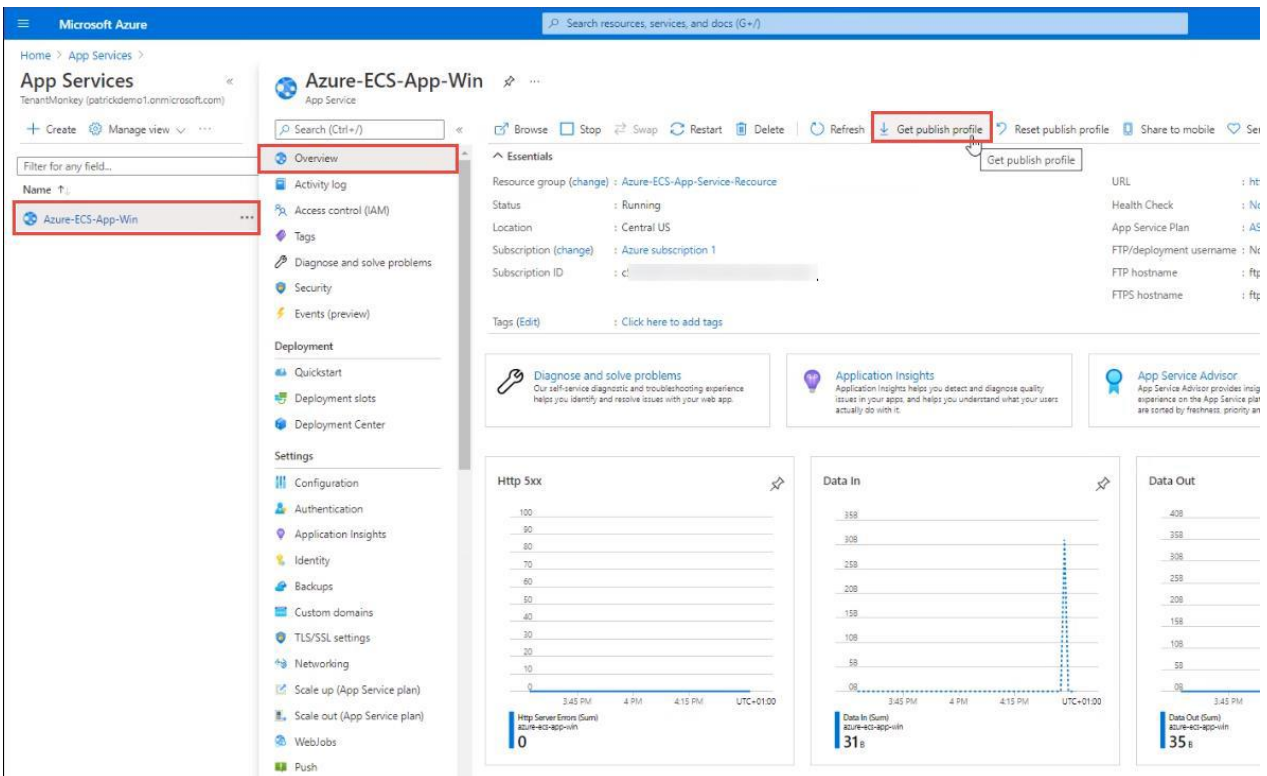


- j. Click the name of the application you created.
- k. Click the **Application URL** to verify that the application is up and running.

## 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.

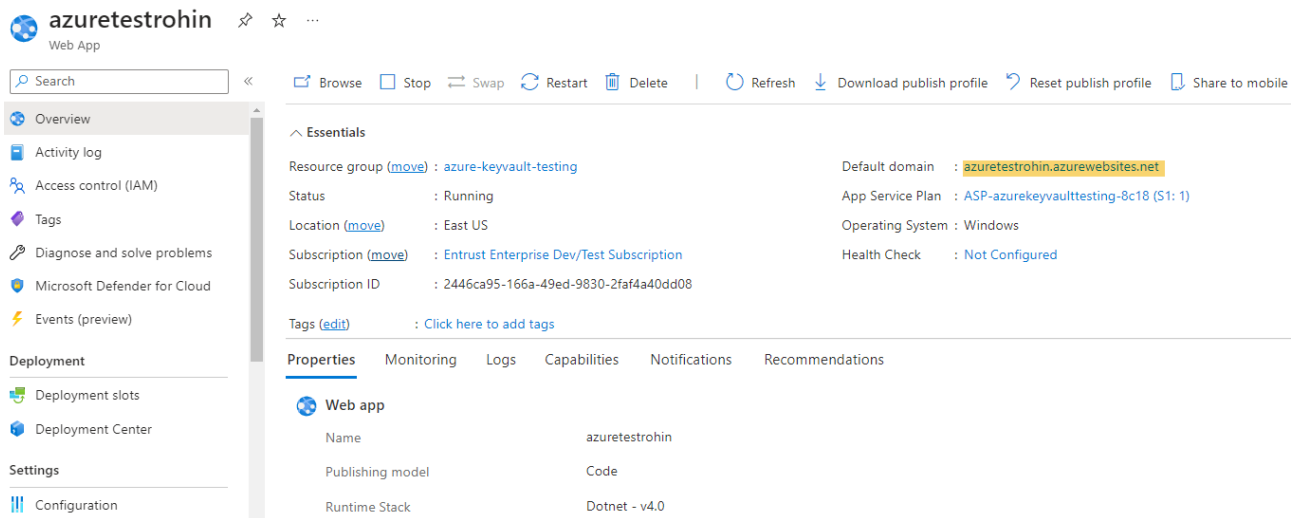
You will see the FTP credentials in the PublishSettings file. The Get publish profile will make it easier to FTP the Connect for Microsoft Azure App binaries to the App Services using any FTP Client.



## 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*

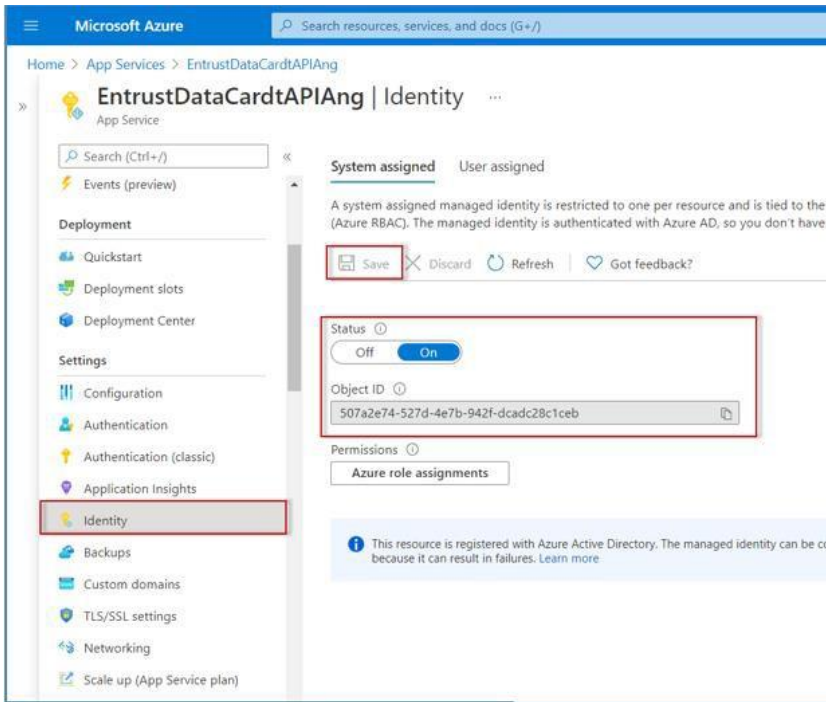
4. Open Main.js file and update lines 11234, 11235 with the *Default Domain* URL you copied in previous step.

```
11210     styleUrls: ['./success.component.css']
11211   }
11212   }, function () { return [{ type: src_app_entrust_service__WEBPACK_IMPORTED_MODULE_1__["EntrustService"] }]; }, null);})();
11213
11214
11215  /**/ }},
11216
11217  /**/ " ./src/environments/environment.ts":
11218  /*|*****|*
11219  |*** ./src/environments/environment.ts ***|
11220  \*****|
11221  /*| exports provided: environment */
11222  /**/ (function(module, __webpack_exports__, __webpack_require__) {
11223
11224    "use strict";
11225    __webpack_require__._r(__webpack_exports__);
11226    /* harmony export (binding) */ __webpack_require__.d(__webpack_exports__, "environment", function() { return environment; });
11227    // This file can be replaced during build by using the `fileReplacements` array.
11228    // `ng build --prod` replaces `environment.ts` with `environment.prod.ts`.
11229    // The list of file replacements can be found in `angular.json`.
11230    const environment = {
11231      production: true,
11232      //azureserver : 'https://pdevtestapimanagement.azure-api.net/',
11233      //apiserver : 'https://pdevtestapimanagement.azure-api.net/',
11234      apiserver: 'https://azuretestrohin.azurewebsites.net',
11235      apiserver: 'https://azuretestrohin.azurewebsites.net/',
11236      // baseUrl: localStorage.getItem('baseUrl'),
11237      KENDO_UI_LICENSE: "/kendo-ui-license.txt"
11238    };
11239    /*
11240     * For easier debugging in development mode, you can import the following file
11241     * to ignore zone related error stack frames such as `zone.run`, `zoneDelegate.invokeTask`.
11242     */
```

## 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**.

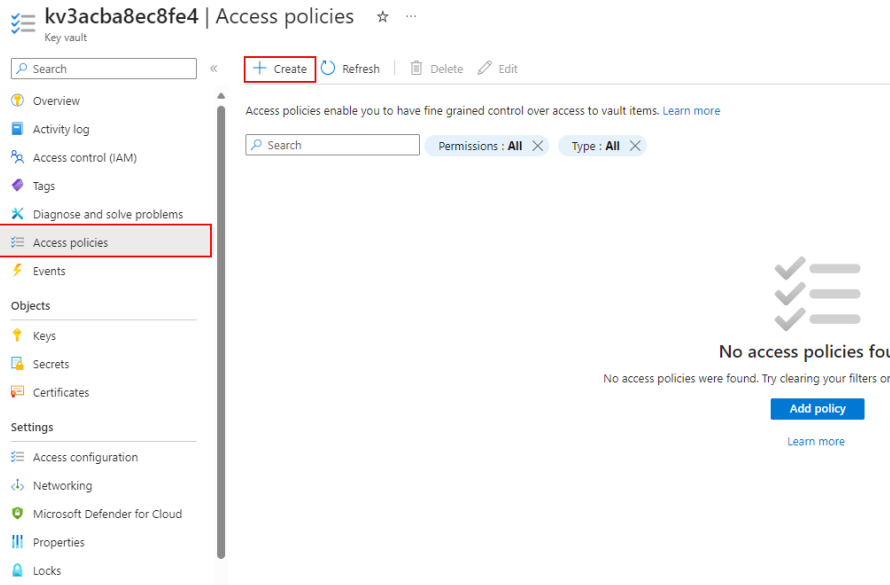
## Step 7: Assign an access policy for the App Service

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**.

## Create an access policy ...

kv3acba8ec8fe4

✔ Permissions ✖ Principal ③ Application (optional) ④ Review + create

Configure from a template

Key, Secret, & Certificate Management

### Key permissions

Key Management Operations

- Select all
- Get
- List
- Update
- Create
- Import
- Delete
- Recover
- Backup
- Restore

Cryptographic Operations

- Select all
- Decrypt
- Encrypt
- Unwrap Key
- Wrap Key
- Verify
- Sign

Privileged Key Operations

- Select all

### Secret permissions

Secret Management Operations

- Select all
- Get
- List
- Set
- Delete
- Recover
- Backup
- Restore

Privileged Secret Operations

- Select all
- Purge

### Certificate permissions

Certificate Management Operations

- Select all
- Get
- List
- Update
- Create
- Import
- Delete
- Recover
- Backup
- Restore
- Manage Contacts
- Manage Certificate Authorities
- Get Certificate Authorities
- List Certificate Authorities
- Set Certificate Authorities
- Delete Certificate Authorities

Privileged Certificate Operations

- Select all
- Purge

Previous

Next

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

Home > Key vaults > kv3acba8ec8fe4 | Access policies >

## Create an access policy ...

kv3acba8ec8fe4

1 Permissions 2 **Principal** 3 Application (optional) 4 Review + create

Principal is required.

Only 1 principal can be assigned per access policy.

Use the new embedded experience to select a principal. The previous popup experience can be accessed here. [Select a principal](#)

a0ed0cd0-d61c-4b44-8e4e-1bc57ecca184

azuretestrohin  
38052e1c-30d2-4a22-a56f-a30552323a3b

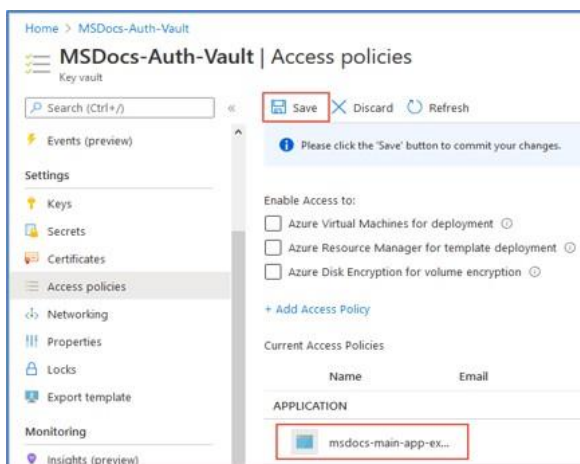
### Selected item

No item selected

Previous

Next

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 policies** page.

7. Click **Save**.

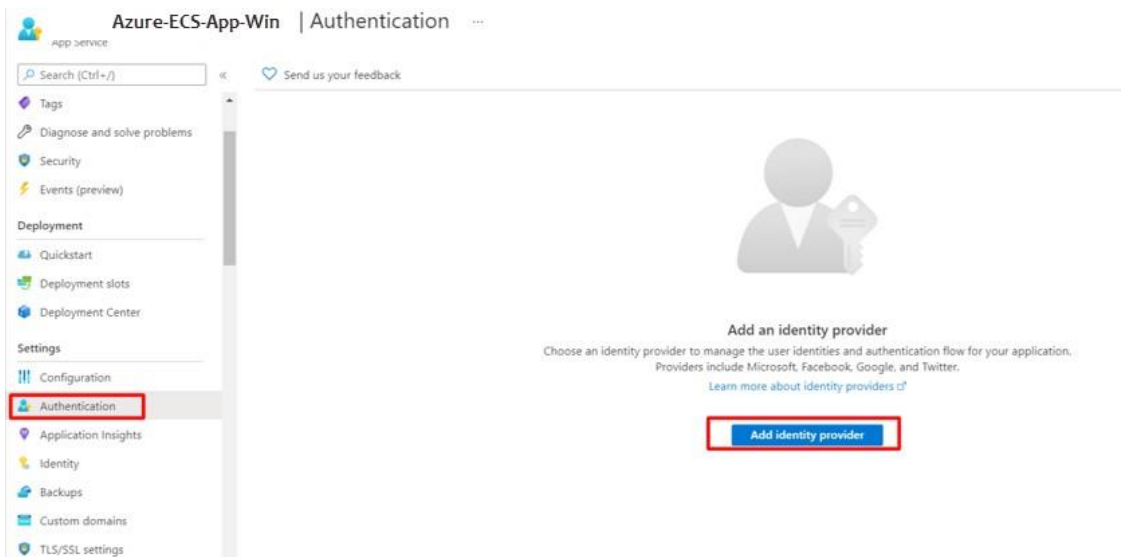
After deploying the Connect for Microsoft Azure App Service, Microsoft, by default, will publish a public facing URL; e.g., <https://azure-ecs-app-win.azurewebsites.net>

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

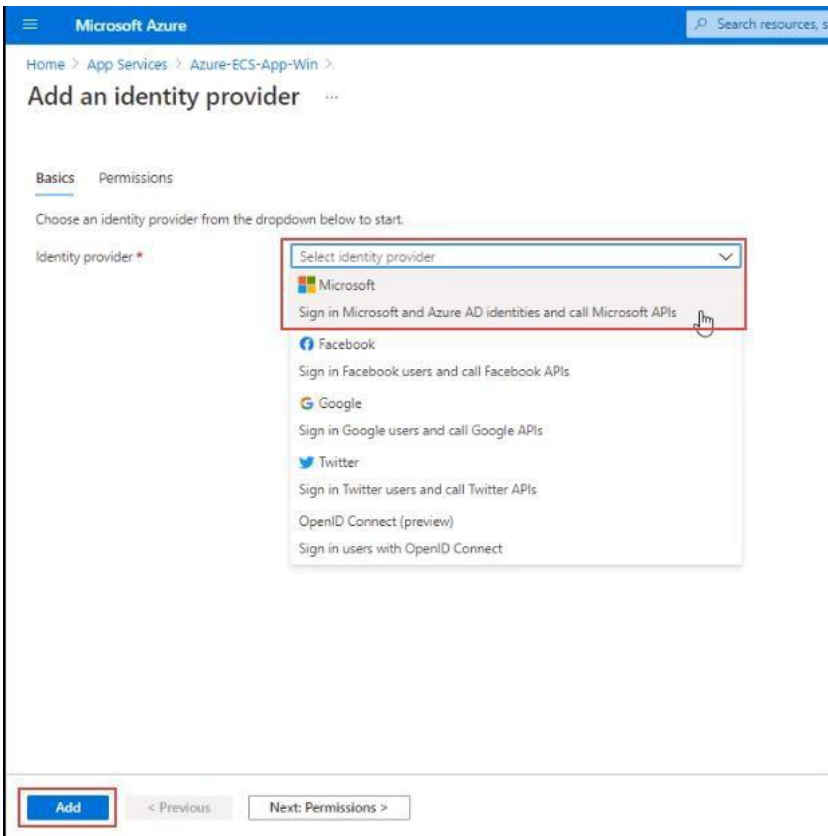
The goal of this step is to provide your users with an authentication process to access the Entrust Connect for Microsoft Azure App.

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.

Microsoft Azure Search resources, services, and docs (G+/)

Home > App Services > Azure-ECS-App-Win >

## Add an identity provider

**Basics** | Permissions

Identity provider\* Microsoft

**App registration**

An app registration associates your identity provider with your app. Enter the app registration information here, or go to your provider to create a new one. [Learn more](#)

App registration type\*   
 Create new app registration   
 Pick an existing app registration in this directory   
 Provide the details of an existing app registration

Name\* Azure-ECS-App-Win

Supported account types\*   
 Current tenant - Single tenant   
 Any Azure AD directory - Multi-tenant   
 Any Azure AD directory & personal Microsoft accounts   
 Personal Microsoft accounts only   
[Help me choose...](#)

**App Service authentication settings**

Requiring authentication ensures all users of your app will need to authenticate. If you allow unauthenticated requests, you'll need your own code for specific authentication requirements. [Learn more](#)

Restrict access\*   
 Require authentication   
 Allow unauthenticated access

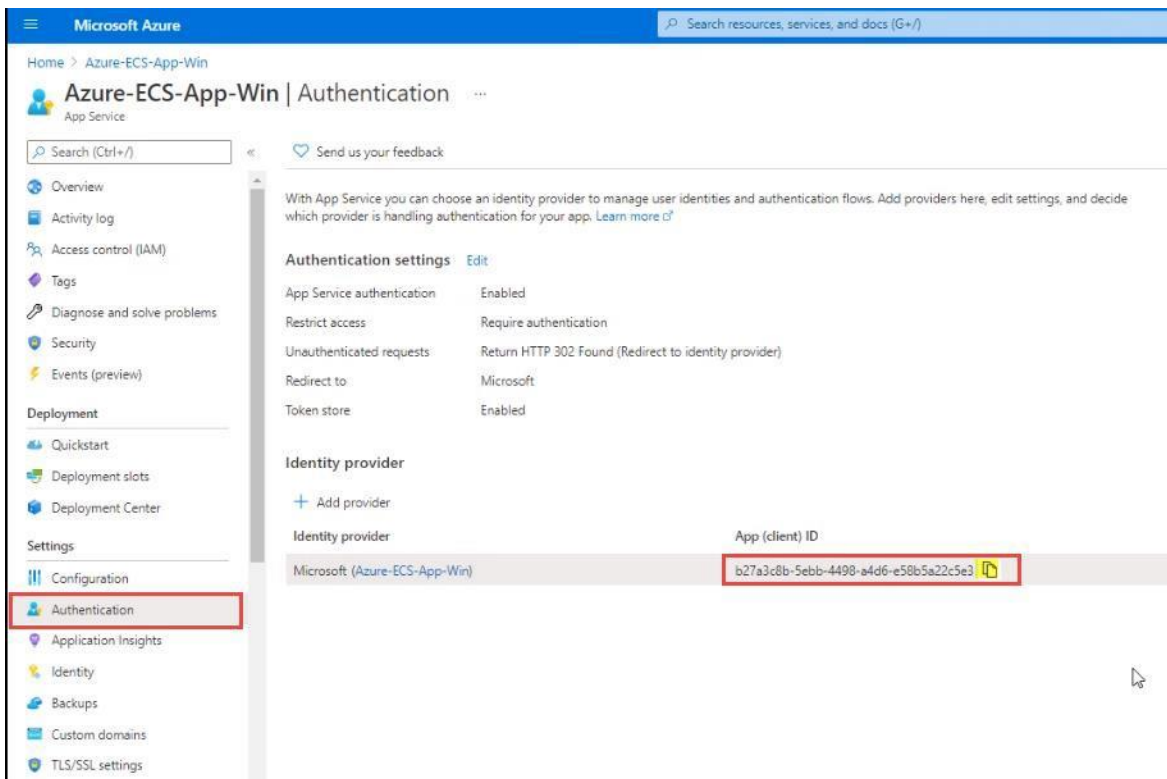
Unauthenticated requests\*   
 HTTP 302 Found redirect: recommended for websites   
 HTTP 401 Unauthorized: recommended for APIs   
 HTTP 403 Forbidden

Redirect to Microsoft

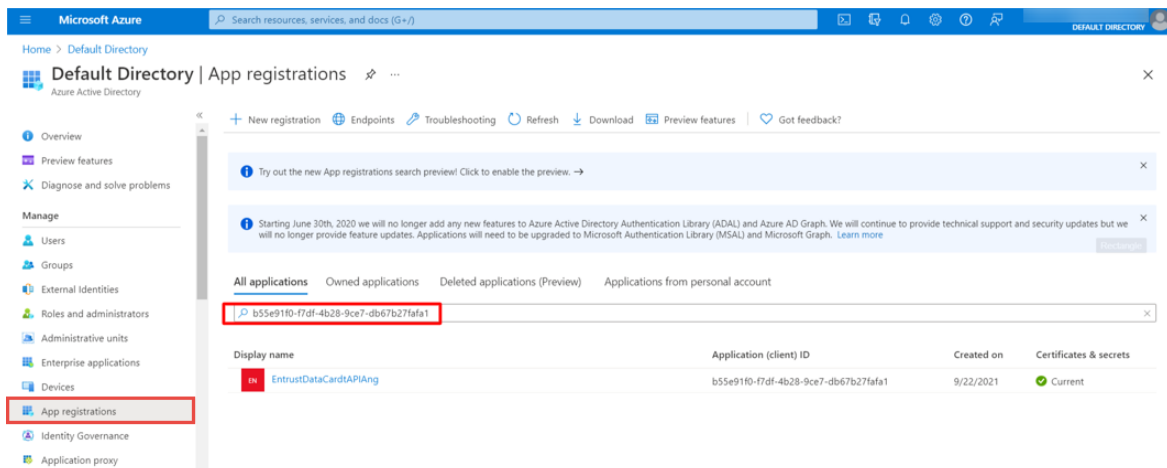
Token store

**Add** < Previous Next: Permissions >

- 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.

The screenshot shows the Microsoft Azure portal interface for configuring an application. The breadcrumb path is Home > TenantMonkey > Azure-ECS-App-Win. The main heading is "Azure-ECS-App-Win | Authentication". The left-hand navigation pane includes sections for Overview, Quickstart, Integration assistant, Manage (Branding, Authentication, Certificates & secrets, Token configuration, API permissions, Expose an API, App roles, Owners, Roles and administrators | Preview, Manifest), and Support + Troubleshooting. The "Authentication" section is highlighted with a red box. The main content area is titled "Platform configurations" and includes a sub-section for "Web" (also highlighted with a red box). Under "Web", there is a "Redirect URIs" section with a description: "The URIs we will accept as destinations when returning authentication responses (tokens) after successfully authenticating or signing out users. Also referred to as reply URLs. Learn more about Redirect URIs and their restrictions". A single URI is listed: "https://azure-ecs-app-win.azurewebsites.net/auth/login/aad/callback". Below this, there is an "Add URI" link. Further down, the "Front-channel logout URL" section is visible, with a text input field containing "e.g. https://example.com/logout".

# 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.