FIDO Pre-reg with PingOne

Yubico

Apr 15, 2026

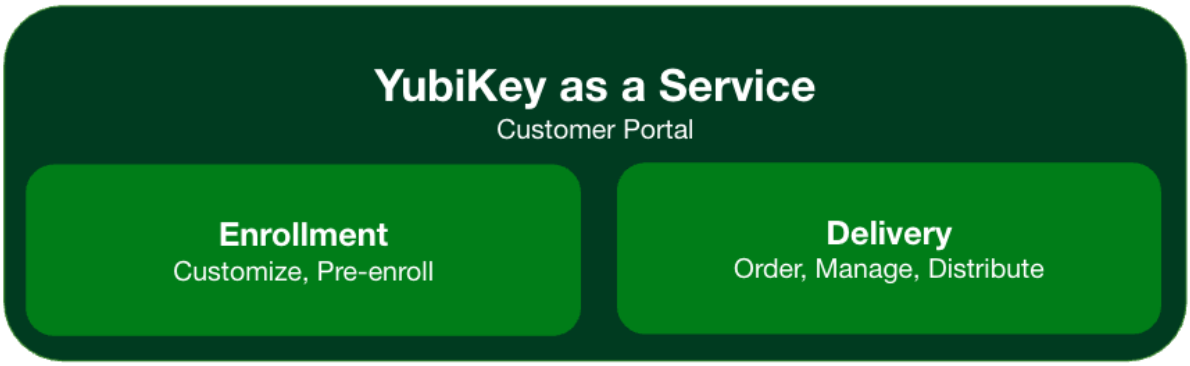
INTEGRATION GUIDE

CHAPTER

ONE

INTRODUCTION

FIDO Pre-reg, part of *YubiKey as a Service - Enrollment*, provides a fully managed service that delivers pre-enrolled YubiKeys directly to end users, enabling secure onboarding from the start.



With FIDO Pre-reg the IT administrator (IT admin) for an organization can use the YubiEnterprise API together with the WebAuthn API of an Identity Provider (IdP) and automated workflows to order pre-enrolled YubiKeys for end users. The YubiKeys are pre-enrolled and shipped directly to the specific end user who received a randomly generated PIN separately.

The following sections describe how to integrate FIDO Pre-reg with PingOne PingID and PingOne AIC (Advanced Identity Cloud). The instructions are intended for solution architects and IT admins setting up shipments of pre-enrolled YubiKeys for their organization's end users in an Microsoft Azure-based environment with PingOne PingID or PingOne AIC, and Microsoft Entra ID.

The instructions assume you have IT administration skills and knowledge of the the YubiEnterprise Delivery API, Microsoft Azure, Entra ID, and PingOne PingID or PingOne AIC. Listed tasks include steps performed both in the Yubico Customer Portal and Microsoft Azure, Entra ID, PingOne PingID and PingOne AIC.

Important: Before you start implementing FIDO Pre-reg, ensure you have the Customization IDs and Product IDs for the YubiKey models you will be shipping to end users. These IDs are provided by Yubico during onboarding of your organization. For more information, see *Prerequisites*.

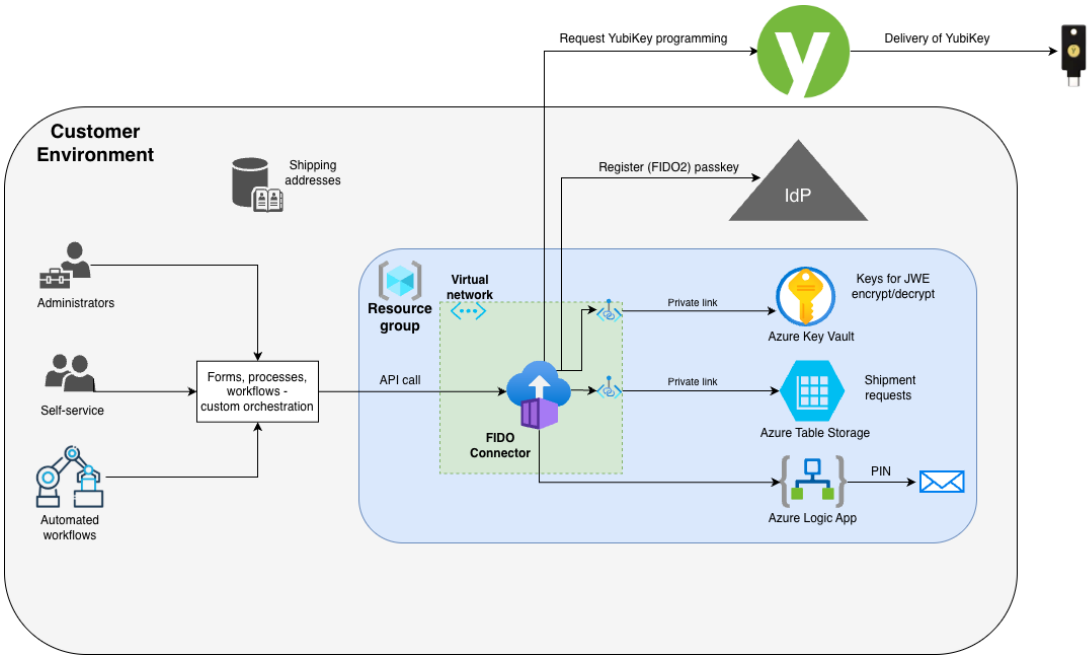
CHAPTER

TWO

ABOUT FIDO PRE-REG WITH PINGONE

The FIDO Pre-reg integration streamlines the deployment process with improved ease of use and enhanced security. End users receive a YubiKey, already pre-enrolled in the customer's Entra ID tenant, directly from Yubico, ready to be used. All use cases such as new and existing employees as well as replacements are supported.

The image below provides an example of a customer environment setup using Microsoft Azure components and **PingOne PingID** or **PingOne AIC** as IdP.



2.1 Process Flow

The following steps illustrate the end-to-end pre-enrolled YubiKey delivery flow:

1. An authorized user (or process) triggers a request for shipment of a pre-enrolled YubiKey for a PingOne PingID/AIC end user via processes and workflows in the customer orchestration environment. The shipping address for the targeted end user is retrieved from supporting systems.
2. The YubiKey request is received by the FIDO Connector deployed in the customer environment.
3. The FIDO Connector makes a request to PingOne PingID/AIC to obtain the required credential creation parameters. When using PingOne AIC, the FIDO Connector calls the PingOne AIC Journey API to initiate the enrollment process and obtain the credential creation parameters.

4. PingOne PingID/AIC returns the credential creation parameters for the targeted end user to the FIDO Connector which then encrypts the information as a credential request.
5. If the Yubico Delivery service is used, the FIDO Connector creates a shipment request to the Delivery service, including the model and shipping information, and attaches the encrypted credential request.
6. After passing through the Delivery service, Yubico decrypts the credential request and creates the credential (user private key) for the specified YubiKey model. The private key is stored in Azure Key Vault.
7. The attestation response from the credential creation along with the PIN are then encrypted.
8. Yubico ships the YubiKey to the targeted end user. Once the shipment is created, the Yubico APIs return a Shipment ID which is store in the Azure Table Storage.
9. The FIDO Connector continuously checks the Delivery service for updated shipment status.
10. When the shipment reaches status "Shipped" in the Delivery service, the FIDO Connector captures the shipping information including tracking number, serial number, firmware version, and encrypted credential response which includes the PIN.
11. The credentials and the PIN are decrypted by the FIDO Connector, and registered in PingOne PingID, or with PingOne AIC using the Journey API.
12. The PIN is communicated to the targeted end user through a preferred delivery method, for example in an email triggered in the Azure Logic App.
13. The end user authenticates to PingOne PingID or PingOne AIC using their YubiKey and the provided PIN. If the PIN was configured for one-time use, the user will be prompted to change the PIN.

The following sections provide an overview of solution features and components.

2.2 Customer Orchestration

The *custom-developed orchestration component* in the customer environment connects the various solution components and drives the interaction between them:

- Interacts with an HR system or other sources to get user addresses for shipments.
- Interacts with PingOne PingID/AIC to initiate the registration of YubiKeys on behalf of end users.
- Interacts with Yubico APIs to request shipment of YubiKeys to end users.
- Might communicate with end users to provide the PIN, separate from the YubiKey delivery.

The customer orchestration represents an aggregate of functional requirements for the orchestration, and can be implemented in any number of platforms, automation tools, or code. For example for Microsoft Azure customers, the orchestration requirements can be fulfilled using services like Azure Logic App, Azure Function App, or other services in their Microsoft Azure subscription.

Yubico provides the FIDO Connector App that can be deployed to Microsoft Azure to perform the most complex orchestration parts. For more information, see *FIDO Connector App*.

Different components and orchestrations can be used for different use cases. Some onboarding YubiKey issuing workflows can be completely automated using Identity Governance and Administration (IGA) tooling. Other self-service workflows or admin-requested YubiKeys might involve manager approvals using ITSM tooling like ServiceNow.

The customer orchestration implements the client-side of the encryption/decryption scheme. It supports the encryption/decryption of individual elements in the credential request and response messages so that the PIN and other passkey (FIDO2) credential information remains accessible only to the customer orchestration. For more information, see *Security Features*.

The customer orchestration components can be configured, customized, and deployed by an IT administrator or a customer orchestration developer.

2.3 FIDO Connector App

The Yubico-developed FIDO Connector app is easily deployed to a Microsoft Azure subscription and handles most of the customer orchestration complexities:

- Exposes an API that can be called from forms, processes, and workflows.
- Performs all interactions with PingOne PingID/AIC for registering YubiKeys in a PingOne PingID tenant.
- Performs all transport encryption before securely transmitting the credential information from the customer orchestration to the FIDO Pre-reg service.
- Keeps track of pending shipments and actively polls the FIDO Pre-reg service to check on status and updates to pending FIDO Pre-reg requests.
- Once the shipment request is ready, the app decrypts and verifies the authenticity of the response from the FIDO Pre-reg service.
- Completes the registration of the YubiKey by calling the PingOne PingID API, or the PingOne AIC Journey API.
- Emails the PIN to the specific contact or end user.

Note: Currently an instance of the FIDO Connector can only be configured for *one IdP at a time*, either Microsoft Entra ID or PingOne PingID. As default, the FIDO Connector is configured to be used with Microsoft Entra ID. To change this, see *Configuring Environment Variables*. Multiple FIDO Connectors can be deployed and call the same Delivery service for FIDO Pre-reg requests.

2.4 PingOne AIC Journeys APIs

PingOne AIC comes with pre-configured *end-user journeys*. A Journey is an end-to-end workflow invoked by a device or an end user. Ping One AIC provides templates for common end-user Journeys, for example to register an account and sign-in.

The Journeys APIs are a set of REST-based endpoints that allow developers to build, manage, and execute complex authentication and authorization flows programmatically.

For more information, see Journeys (PingOne AIC documentation) .

2.5 FIDO Pre-reg APIs

The FIDO Pre-reg API provides a shipping request API to the customer orchestration and generates fulfillment requests to Yubico. The API supports the communication of encrypted credential registration data between the customer orchestration and Yubico, and is an extension of the YubiEnterprise API for the Delivery service facilitating the global distribution of YubiKeys.

The FIDO Connector also has an API that is deployed and used in the customers Azure environment to orchestrate shipment and credential requests. For more information, see API Reference.

2.6 Security Features

The following provides an overview of security features in an implementation of FIDO Pre-reg with PingOne PingID or PingOne AIC.

2.6.1 PingOne PingID/AIC Access

Yubico has no access to enroll and/or activate user passkey (FIDO2) credentials directly into a customer's Entra ID, or PingOne PingID/AIC tenant.

2.6.2 Pre-enrolled Credentials

Because Yubico has no access to the customer's PingOne PingID/AIC tenant, Yubico registers authenticators (YubiKeys) using the passkey credential creation parameters provided in a customer-initiated shipment request. The credential responses are then returned for retrieval by the customer orchestration, and the credential details are used by the customer orchestration to register YubiKeys with PingOne PingID/AIC.

2.6.3 PIN Provisioning

Yubico generates a PIN for a given YubiKey and returns it to the Delivery service for retrieval by the customer orchestration, which then decides how that PIN gets communicated to the end user.

2.6.4 Transport Encryption

To mitigate the risk of exposing sensitive information, for example creation parameters, serial numbers, and PIN related to YubiKey assignments within the Delivery service, all data transferred from the Yubico environment to the customer orchestration system is encrypted using a secure transfer mechanism. This ensures that Yubico personnel and systems have no access to or visibility into, any credential-related data at any stage of the process.

CHAPTER

THREE

INTEGRATION PROCEDURE

The following provides an overview of the steps to get started using FIDO Pre-reg with Microsoft Azure components and PingOne PingID/AIC, and create a shipment of a pre-enrolled YubiKey.

3.1 Prerequisites

Ensure you have the following before starting the implementation procedure:

- Provided by Yubico:
  - A Yubico subscription plan. For questions about Yubico subscription services, contact your Yubico sales representative.
  - Yubico Customer Portal access with FIDO Pre-reg enabled. This is provided during onboarding of your organization.
  - Customization ID (CID), Product ID, and Inventory ID for the YubiKey delivery.
  - An ARM (Azure Resource Manager) template JSON file and a Docker image for deploying components in Azure.
  - Credentials for the Yubico container registry for the FIDO Connector app.
  - An Azure Resource Group permissions template.
  - PingOne Ping AIC Journey configuration templates.
- A PingOne PingID or PingOne AIC instance with FIDO2 passkeys/security keys support.
- An Azure Portal Subscription with a Resource group supporting the Container app, Azure table, Key Vault, and Logic App resource types.
- An Office 365 License or another preferred email service to send PINs to end users.
- A defined method for sourcing shipping addresses for the YubiKey recipients.
- A defined preference for how recipients will receive YubiKey PINs, for example via email.
- The following administrative roles are required for the implementation:
  - *Authentication Policy Administrator* role in PingOne PingID/AIC.
  - *Application Administrator* role in PingOne PingID/AIC.
  - *Application Administrator* role in Microsoft Entra ID.
  - *Authentication Policy Administrator* role in Microsoft Entra ID.
  - *Global Administrator* role in Microsoft Entra ID.

- *Privileged Role Administrator* role in Azure.

3.2 Integration Steps

Note: Currently an instance of the FIDO Connector can only be configured for one IdP at a time, either Microsoft Entra ID or PingOne PingID/AIC. As default, the FIDO Connector is configured to be used with Microsoft Entra ID. To change this to PingOne PingID/AIC, see *Configuring Environment Variables*.

The following steps lets you set up the FIDO Pre-reg integration and create a first shipment of a pre-enrolled YubiKey:

1. Configure PingOne for policy authentication and on-behalf of registration, either one of the following:
   - *Configure PingOne PingID*
   - *Configure PingOne AIC*
2. *Configure Microsoft Entra ID* to enable container authentication.
3. *Deploy Azure components* such as Resource group and ARM template.
4. *Test and verify the deployment* using for example a Test client.
5. *Create shipment of a pre-enrolled YubiKey* from your organization's IT environment.

The sections in the following describe each step in detail.

CHAPTER

FOUR

CONFIGURING PINGONE PINGID

The following sections describe the configuration steps required in PingOne PingID. If you are using *PingOne AIC*, see *Configuring PingOne AIC*.

4.1 FIDO Policy Authentication

Note: You will need a user with the *Authentication Policy Administrator* role in PingOne PingID to complete the configuration steps.

To configure the PingOne PingID authentication policies, do the following:

1. Sign in to the PingOne PingID console.
2. Go to **Authentication > FIDO Policies**.
3. Click **+** to create a policy, or click the **Edit** icon for the desired policy in the **Enhanced FIDO Policies** section.
4. Configure the policy as follows:
    - **Device Display Name:** For example "Security Key". Controls how the FIDO authenticator is displayed to the user. Use "Label" for a static, non-translated name, or "Translatable Keys" for a localized display of the device name.
    - **FIDO Device Aggregation:** When set to "On" (recommended), all devices of the same type (for example security keys) appear as one entry using a single display name during user authentication. When set to "Off", each device is listed separately with its unique name.
    - **Relying Party ID:** Specifies the domain identifier that Ping Identity asserts as the FIDO authenticator's origin during registration and sign in. Select "PingOne" to use a standard PingOne domain such as "pingone.com".
    - **Discoverable Credentials:** Controls whether the FIDO policy encourages or enforces the use of passkeys (resident credentials) that are stored directly on the authenticator itself. Select "Preferred".
    - **Authenticator Attachment:** Defines which physical type of FIDO authenticator the policy allows a user to register and use. Select "Cross-platform" to require an external device like a USB security key or a phone.
    - **Manage verification settings:** Controls whether the authenticator must enforce a secondary verification factor like a PIN, or biometric scan, for high assurance.
        - **User Verification:** Selecting "Preferred" is recommended to avoid blocking users. Contact your Yubico Professional Services team to discuss options for this setting in your specific environment.
        - **Enforce PIN Length:** Select "Disabled".

  - Select “Enforce During Authentication”.

- **User Presence Timeout:** Defines the maximum duration (minutes or seconds) that PingOne PingID will wait for the user to interact with their FIDO authenticator after the challenge is issued. Set to for example “2 Minutes”.
- **Backup Eligibility:** Defines whether the FIDO policy allows authentication using cloud-synced passkeys. Select “Disallow” (recommended).
- **User Display Name:** Defines the text the FIDO authenticator displays to the user for account selection during sign-in. Select for example “Email Address”, “Name (Given, Family)”, “Username”.
- **Attestation Type:** Determines the level of cryptographic proof required from the FIDO authenticator during the registration process to confirm the device’s legitimacy and origin. Select “Direct” (recommended).
- **Attestation Requirements:** Select “Allow FIDO Certified Authenticators”. If specific YubiKey models or AAGUIDs are required, search for “YubiKey”, and select the desired YubiKey models in the list that is displayed. See YubiKey hardware FIDO2 AAGUIDs.

5. Click **Save**.

For more information about PingOne PingID policies, see FIDO Policies (PingOne PingID documentation).

4.2 Enabling On-Behalf of Registration

Note: You will need a user with the Application Administrator role in PingOne PingID to complete the configuration steps.

4.2.1 Creating an Application

To register a FIDO Pre-reg service application in PingOne PingID, do the following:

1. Sign in to the PingOne PingID console.
2. Go to **Applications** > **Applications**.
3. Click **+** next to **Applications** to add new application.
4. Provide a descriptive **Application name**, for example “Yubico FIDO Pre-reg Service”.
5. Select “Worker” as the **Application Type**.
6. Click **Save**.

4.2.2 Granting Role to Worker App

To add a role after successful registration of the Worker app, do the following:

1. In the PingOne PingID console, click **Grant Roles**, or go to **Roles**.
2. From the **Available responsibilities**, expand the **Identity Data Admin** role.
3. Select the appropriate **Environment**.
4. Click **Save**.

4.2.3 Enabling the Worker App

To enable the successfully registered Worker app, do the following:

1. In the PingOne PingID console for the worker app, go to the **Overview** tab.
2. Save the **Client ID** value to be used later for the `FIDO_Connector_Ping_Client_Id` parameter.
3. Save the **Client Secret** value to be used later for the `FIDO_Connector_Ping_Client_Secret` parameter.
4. Save the **Environment ID** value to be used later for the `PINGONE_ENVIRONMENT` parameter.
5. Enable the Worker app by toggling the **Enable toggle** to on.

For more information, see Adding an application and Configuring roles for a worker application (PingOne PingID documentation).

CHAPTER

FIVE

CONFIGURING PINGONE AIC

The following sections describe the configuration steps required in PingOne AIC. If you are using *PingOne PingID*, see *Configuring PingOne PingID*.

5.1 Adding a Secrets Variable

To add a Secrets variable, do the following:

1. Sign in to the PingOne AIC console.
2. Go to **Tenant Settings** from your profile on the right side top corner.
3. Select the **Variables** tab.
4. Click **Add below var**.
5. Provide a **Description**, for example "YFPR Service - Client Secret", and leave the recommended **Expires** option as-is.
6. Click **Add**.
7. Save the value of the **Secret**, this will be used later as the `FIDO_Connector_Client_Secret` parameter.

5.2 Creating a Registration Journey

To create a Journey for the *credential registration*, do the following:

1. Sign in to the PingOne AIC console.
2. Create/Import the **Registration Journey** template *provided by Yubico*.

5.3 Creating an Authentication Journey

To create a Journey for *authentication*, do the following:

1. Sign in to the PingOne AIC console.
2. Create/import the **Authentication Journey** template *provided by Yubico*.
3. Make this the *default* authentication journey.

5.4 Enabling On-behalf of Registration

In this step you will create a client application that will be used by the FIDO Connector to call the *previously created Registration Journey*, and retrieve the Client ID and Client Secret values.

To create and register the client application, do the following:

1. Sign in to the PingAIC console.
2. Go to **Applications**.
3. Click **+ Custom Application**.
4. Create an **OIDC Service Application** with a confidential secret.
5. Provide a descriptive **Application Name**, for example "Yubico FIDO Pre-reg Service".
6. Click **Save**.
7. After successfully registering the app, go to **OAuth2 Clients**.
8. Select the previously created application and go to **Sign On**.
9. Save the value of **Client ID**, this will be used later as the `FIDO_Connector_PingOne_AIC_Client_Id` parameter in the ARM template.
10. Save the value of the **Client Secret**, this will be used later as the `FIDO_Connector_PingOne_AIC_Client_Secret` in the ARM template.
11. Configure the following variables, for values see *Configuring Environment Variables*:
    a. `PING_AIC_REALM`
    b. `PING_AIC_AUTH_BASE_URL`
    c. `PING_AIC_API_BASE_URL`

CHAPTER

SIX

CONFIGURING MICROSOFT ENTRA ID

The steps in this section register the container APIs and expose them so they can be used by the calling applications, for example a web app or ITSMs in the customer environment.

Note: Most of the registration steps can be performed by an admin user with the *Application Administrator* role. However, to complete some steps a user with the *Global Administrator* role is required as indicated in the procedure.

6.1 Registering Apps

In this step you will register the FIDO Connector app and the FIDO Pre-reg Test Client (optional) for testing your deployment.

6.1.1 FIDO Connector App

To register the FIDO Connector App, do the following:

1. Sign in to the Microsoft Entra admin center and expand the **Entra ID** section.
2. Click **App registrations**.
3. Click **+ New registration**.
4. Provide a descriptive **Name**, for example "Yubico FIDO Pre-reg Client App".
5. Select the appropriate **Supported account types**, this defines the account types that can use the app or access the API. For this deployment, "Single tenant only" is sufficient.
6. Click **Register**.
7. Under the **Manage** section for the app, click **Expose an API**.
8. Click **Add** next to the **Application ID URI**.
9. Edit the **Application ID URI** to a value like `api://fido-connector-api.{verified domain name}`.
    - The verified domain name can be either a custom domain that has been verified by the tenant, or you can use the default domain that ends with ".onmicrosoft.com".
    - The Application ID URI represents the scope that clients will use when authenticating to call the API. This value will be used in the ARM template for `FIDO_Connector_Allowed_Audiences`. The URI does not need to be resolvable, but should have a descriptive scope name.
    - Save the value of the **URI** for later use.

10. Click **Save**.
11. Under **Manage > Expose an API**, click **+ Add a scope** and set the following:
    - For **Scope name** and **Admin consent display name**, enter "**create_request**".
    - For **Consent**, select "Admins only".
    - For the **Admin consent description**, enter "Allows Yubico FIDO Pre-reg requests".
12. Click **Add scope**.
13. Under **Manage**, click **Certificates & secrets**.
14. Click **+ New client secret**.
15. Provide a **Description**, for example "YFPR Service - Client Secret", and use the recommended **Expires** option.
16. Click **Add**.
17. Save the **Value** of the **Secret ID** for later use in the ARM template for the `FIDO_Connector_Client_Secret`.
18. In the Microsoft Entra Admin center, click **Overview** for the FIDO Connector app.
19. Save the **Application (client) ID** value for later use in the ARM template for the `FIDO_Connector_Client_Id`.

For more information, see Register an application with the Microsoft identity platform (Microsoft documentation).

6.1.2 FIDO Pre-reg Test Client

Registering this app is *optional*. However, the app is useful when testing direct calls to the FIDO Connector App. The application credentials created here can be used in a Postman test client or any other HTTP test client when testing the app deployment.

To register the FIDO Pre-reg Test Client app, do the following:

1. Sign in to the Microsoft Entra admin center and expand the **Entra ID** section.
2. Click **App registrations**.
3. Click **+ New registration**.
4. Provide a descriptive **Name** like "Yubico FIDO Pre-reg Test Client" and click **Register**.
5. Under **Manage**, select **API permissions**.
6. Click **+ Add a permission**.
7. Click **APIs my organization uses** at the top.
8. Search for **Credential-Container-API** and select the API in the list.
9. Select **create_request**.
10. Click **Add permissions**.
11. Under **Manage**, click **Certificates & secrets**.
12. Click **+ New client secret**.
13. Provide a **Description**, for example "YFPR Service - Client Secret", and use the recommended **Expires** option.
14. Click **Add**.

The app credentials you created here will be used later when testing the app deployment. For more information, see *Testing the Deployment*.

CHAPTER

SEVEN

DEPLOYING TO AZURE

Following these steps you will deploy the FIDO Connector app itself along with the underlying infrastructure and required configuration changes. Before you start the deployment, ensure that you have successfully completed the previous steps, and that you have the appropriate permissions to deploy Azure services. See *Prerequisites*.

7.1 Creating an API Token

To create a Yubico API authentication token, sign in to the Customer Portal with the account for the application that will be calling the YubiEnterprise API. Click the organization name on the top of the left menu and select **Manage API token**. In the token dialog that appears, click **Create API token** and save the token for future use. For more information, see Creating API Tokens.

7.2 Creating a Resource Group

Note: The *Subscription Owner* role or equivalent is required for this step.

To create a Resource group, do the following:

1. Login to the Azure Portal.
2. Search for **Resource groups**.
3. Click **Create**.
4. Select the appropriate **Subscription** and **Region**, and provide the appropriate **Resource groupname**, for example "Yubico FIDO Pre-reg Service".
5. Click **Review + create**.

7.3 Creating a Custom Role

Note: This step is *optional* if you have the *Global Administrator* role or, are the owner of the subscription. Otherwise, you will need a role that lets you create a Custom Role.

To create a Custom role, do the following:

1. In a text editor, open the file *yubico-fpr-deploy-custom-role-permissions.json* and do the following:

   a. Find and replace `{role_name}` with a descriptive role name, for example "Yubico FIDO Pre-reg Custom Role".
   b. Find and replace `{subscription_id}` with the appropriate subscription ID.
   c. Find and replace `{rg_name}` with the appropriate resource group name.
2. Save the JSON file.
3. In the Azure portal, go to the *previously created Resource Group*.
4. Go to **Access control (IAM)**, click **Add** and select "Add Custom Role".
5. For **Baseline permissions**, select "Start from JSON".
6. Select the previously edited **yubico-fpr-deploy-custom-role-permissions.json**.
7. The **Custom role name** field and **Assignable scopes** tab should have been populated according to the updates made to the JSON file.
8. Click **Review + create**.
9. Verify that everything looks correct and click **Create**.

7.4 Assigning the Custom Role

Note: This step is *optional* if you have the *Global Administrator* role or, are the owner of the subscription.

To assign the Custom role to users, do the following:

1. In the Azure portal, go to the *previously created Resource Group*.
2. Go to **Access control (IAM)**.
3. Click **Add > Add role assignment**.
4. Select the **Privileged administrator roles** tab.
5. Search for and select the *previously created Custom role name*.
6. Click **Next**.
7. On the **Members** tab, verify that the selected role is correct, and select the appropriate members to assign this role to.
8. Click **Next**.
9. On the **Conditions** tab, verify that the selected role is correct, and select **Allow user to assign all roles (highly privileged)**.
10. Click **Next**.
11. Click **Review + assign**.
12. Verify the information and click **Review + assign**.

7.5 Verifying Custom Role Assignment

Note: This step is *optional* if you have the *Global Administrator* role or, are the owner of the subscription.

To verify custom role assignments, do the following:

1. In the Azure portal, go to the *previously created Resource Group*.
2. Go to **Access control (IAM)**.
3. Click **Check access**.
4. Search for and select *users previously assigned to the custom role*.
5. Under **Role assignments**, verify that the custom role was assigned to the user.

7.6 Deploying the ARM Template

Note: The previously created *Custom Role*, *Global Administrator*, or *Subscription Owner* role is required for this part of the deployment.

To deploy the ARM template, do the following:

1. Sign in to the Azure portal.
2. Search for and select **Deploy a custom template**.
3. Click **Build your own template in the editor**.
4. Click **Load file**, then select the ARM template file provided by Yubico.
5. Click **Save**.
6. In the configuration menu, provide the following values:
    - **Subscription:** Select the appropriate subscription.
    - **Resource group:** Select or create a resource group for this deployment.
    - **Region:** Select the appropriate region.
    - **MS_Login_Online_Endpoint:** Use default, only change if your tenant uses a different Microsoft endpoint.
    - **MS_Graph_Endpoint:** Use default, only change if your tenant uses a different Microsoft endpoint.
    - **Azure_Mgmt_Endpoint:** Use default, only change if your tenant uses a different Microsoft endpoint.
    - **Azure_Vault:** Use default, only change if your tenant uses a different Microsoft Login endpoint.
    - **Key Vault_Resource_Name:** Provide a unique name for your key vault instance.
    - **Azure_Storage:** Use default, only change if your tenant uses a different Microsoft endpoint.
    - **Storage Account_Resource_Name:** Provide a unique name for the storage instance.
    - **YED_API_TOKEN:** Paste the value you saved when *creating the API token*.
    - **Container_App_Name:** Provide a unique name in *lower case*.
    - **Container_Registry_Name:** The Registry name *provided by Yubico*.

- **Container_Image_Name_Tag:** The Registry Container Image name and version Tag *provided by Yubico*.
- **Container_Registry_User:** The Registry user name *provided by Yubico*.
- **Container_Registry_Password:** The Registry password *provided by Yubico*.
- **FIDO_Connector_Client_Id:** Client ID value from the *app registration*.
- **FIDO_Connector_Client_Secret:** Client Secret value from the *app registration*.
- **FIDO_Connector_Allowed_Audiences:** Value from Exposing the API when *registering the app*. List of scopes/audiences that a client application must use for calling the app's API. Default value `api://fido-connector-api.{verified domain name}`. Ensure this is formatted as an array of strings, for example `["scope_1", "scope_2"]`.
- **FIDO_Connector_Allowed_Client_Apps:** Value from Exposing the API when *registering the app*. List of app registrations that are allowed to call this app's API, as registered in app registrations. The optional app registration, if performed, can be used as the ID string. Ensure that the formatting is an array of strings including each client app ID. Example: `["client_app_id_1"]`.
- **Workflows_Send_shipment_pin_name:** Use default, or set a name based on your preferred naming convention.
- The ARM template includes a reference implementation of the private endpoints listed below, used by the FIDO Connector Container app (default values do not need to be changed):
  - **virtualNetworkName:** Use default, or set a name based on your preferred naming convention.
  - **virtualNetworkAddressPrefix:** Use default, or set a desired IP address range.
  - **subnetName:** Use default, or set a name based on preferred naming convention.
  - **subnetAddressPrefix:** Use default, or set a desired IP address range.
  - **privateEndpointSubnetName:** Use default, or set a name based on preferred naming convention.
  - **privateEndpointSubnetAddressPrefix:** Use default, or set a desired IP address range.
  - **keyVaultPrivateEndpointName:** Use default, or set a name based on preferred naming convention.
  - **tableStorageAccountPrivateEndpointName:** Use default, or set a name based on preferred naming convention.
- **For PingOne PingID:**
  - **FIDO_Connector_Ping_Client_Id:** Enter the value from *Enabling the Worker App*.
  - **FIDO_Connector_Ping_Client_Secret:** Enter the value from *Enabling the Worker App*.
- **For PingOne AIC:**
  - **FIDO_Connector_PingOne_AIC_Client_Id:** Enter the value from *Enabling On-behalf Registration (AIC)*.
  - **FIDO_Connector_PingOne_AIC_Client_Secret:** Enter the value from *Enabling On-behalf Registration (AIC)*.

7. Click **Review + create**.
8. After successful deployment, verify that the resources were created.
9. Open the **Container app** and save the **Application Url** value for the parameter `FIDO_Connector_Host_URL` for later use.

7.7 Configuring Container App Permissions

Note: This step requires the *Subscription Owner* role, or role that can create role assignments.

To configure Key Vault and Storage permissions for the Container App, do the following:

1. In the Azure portal, go to **Resource Group > Container App**.
2. In the left navigation, click **Security > Identity**.
3. Click **Azure role assignments**.
4. Ensure the correct subscription is selected.
5. Click **Add role assignment** and and configure as follows:
    a. For **Scope**, select "Key Vault".
    b. For **Subscription**, enter your subscription.
    c. For **Resource**, enter the Key Vault you deployed with this template.
    d. For **Role**, select "Key Vault Administrator".
    e. Click **Save**.
6. Click **Add role assignment** and configure as follows:
    a. For **Scope**, select "Storage".
    b. For **Subscription**, enter your subscription.
    c. For **Resource**, enter the Storage Account you deployed with this template.
    d. For **Role**, select "Storage Table Data Contributor".
    e. Click **Save**.
7. Click **Refresh** and verify that the two roles were successfully added.

7.8 Authorizing Logic App Office 365 Usage

To authorize the Logic App to call the *Outlook/Office365 connector*, do the following:

1. In the Azure portal, go to **Resource Group > Send_shipment_pin Logic App**.
2. In the left navigation, click **Development Tools > API connections**.
3. Select **office365**.
4. Go to **General > Edit API connection**.
5. Click **Authorize**.
6. Click **Authorize** again.
7. Sign in with the account that will be used as sender of FIDO Pre-reg PIN emails.
8. After signing in, select **Save**.

7.9 Configuring Environment Variables

Note: To use PingOne PingID or PingOne AIC as the default IdP, you need to change the environment variables configured in the Container App using the values described in the following. Restart the application when done.

To configure environment variables for the Container app, do the following:

1. In the Azure portal, go to **Resource Group > Send_shipment_pin Logic App**.
2. Save the **Workflow URL**, this will be used for the `Send_PIN_URL` value below.
3. Go to **Resource Group > Container App**.
4. In the left navigation, click **Application > Containers**.
5. Select the **Environment variables** tab.
6. Update the value for **EMAIL_API_SEND_ENDPOINT** to the value of parameter `Send_PIN_URL` saved in step 2.
7. Click **Add** and add the following **Environment variables** with source as "Manual Entry":

   For PingOne PingID:

Name	Value
IDP_DEFAULT	pingone
PINGONE_ENVIRONMENT	Your PingOne PingID Environment ID, see *Enabling the Worker App*
PINGONE_DEFAULT_RELYING_PARTY	pingone.com or custom value.
PINGONE_AUTH_BASE_URL	https://auth.pingone.com or custom value.
PINGONE_API_BASE_URL	https://api.pingone.com/v1 or or custom value.
PINGONE_PRE_REGISTRATION_TIMEOUT	15

   For PingOne AIC:

Name	Value
IDP_DEFAULT	ping-aic
PING_AIC_REALM	Your PingOne AIC environment ID or realm name.
PING_AIC_DEFAULT_RELYING_PARTY	Custom value.
PING_AIC_AUTH_BASE_URL	Custom value.
PING_AIC_API_BASE_URL	Custom value.
PING_AIC_PRE_REGISTRATION_TIMEOUT	Ping AIC timeout is configured in the Journey.
PING_AIC_JOURNEY	Registration Journey name, see *Creating a Registration Journey*

8. Click **Save as a new revision**.
9. Click **Overview** in the left navigation.
10. **Stop** then **Start** the container to ensure the new environment variables are loaded.

CHAPTER

EIGHT

TESTING THE DEPLOYMENT

To test the deployment you will retrieve an access token and make an API call to ensure that the app was correctly deployed to your environment. In the test you can leverage the APIs directly, for example by using a client like Postman, or any HTTP client. The test assumes that you have registered the FIDO Pre-reg Test Client as described in *Registering Apps*.

To *retrieve an access token*, do the following:

1. Go to the previously created *FIDO Pre-reg Test Client*.
2. From your client, make an API call using the following request:
    - **Method:** `POST`
    - **URL:** `https://login.microsoftonline.com/{your azure tenant domain}/oauth2/v2.0/token`
    - **Header:** Content-Type - `application/x-www-form-urlencoded`
    - **Body:**
        - **grant_type:** `client_credentials`
        - **client_id:** Client ID created for the *Yubico FIDO Pre-reg Test Client*.
        - **client_secret:** Client secret from when you *created the test client*.
        - **scope:** `api://fido-connector-api.{verified domain name}/.default`
3. Send the request.
4. From the response, copy the `access_token value`.

To *call the API*, do the following:

1. From your client, make an API call using the following request:
    - **Method:** `GET`
    - **URL:** `https://{url of your container app}/v1/status`. For base URL, copy the Application URL from your Container App.
    - **Header:**
        - **Authorization** - Bearer `{access_token from previous step}` Example: `Bearer eyJ0...`
        - **Content-Type** - `application/json`
2. From this API call you should receive a 200 status code, with a response payload that outlines the different environment configurations that were made during setup of the components. Double-check these responses to ensure that they are correct.

8.1 Troubleshooting

The following provides basic troubleshooting steps for common deployment issues.

8.1.1 Where to Start?

1. What is the error message that you are getting?
2. Verify the environment variables and key vault values:
    a. *Key Vault Administrator* is required to view key vault entries.
    b. Verify secrets entries for the YubiEnterprise API. Must be a valid token retrieved from the Customer Portal, see *Creating an API Token*.
3. Review response message from the Credential API Container App.
4. Check Container App Logs.
5. Verify Environment Variables.
6. Verify Azure Key Vault values.

8.1.2 Verifying Shipment Status in Storage Browser

1. Sign in to the Azure portal.
2. Go to the **Resource Group**.
3. Go to **Storage Account** > **Storage Browser** > **Tables**.
4. Click the **fprshipments** table.
5. Find the desired shipment by `shipmentId`.
6. Verify that the state of the shipment is complete. A `shipmentId` status that is not updated to "complete" will continue to retry. Once you investigate and resolve the issue, the status can be manually updated to "complete".
7. If a `shipmentId` has encountered an error during processing, it will be recorded in the **fprshipments** table fields `error_kind` and `error_message`.
8. Once you have investigated and resolved the issue, the shipment will be reprocessed during the next scheduled run to "complete" status. Alternatively, the status can be manually updated to "complete" if the cause of the error cannot be resolved.

Note: The shipment status and processing error recorded in the **fprshipments** table can also be obtained by calling the API as described in Get Shipment Request Status in the API Reference. You can find more details to understand the error in *Checking FIDO Connector Logs*.

8.1.3 Verifying Delivery Status of YubiKey PIN

By default FIDO Pre-reg is configured to send emails to the end user's manager. If the manager relationship for the end user is not set up, or the manager does not have an email address configured, the PIN delivery will fail.

To verify that the PIN delivery was successful, do the following:

1. Sign in to the Azure portal.
2. Go to **Resource Group** > **Logic App**.
3. In the left menu, click **Development tools** > **Run history**.
4. Verify that you have a record with **Status** "Succeeded".
5. If the status is "Succeeded":
    a. Open the history record.
    b. Select the connector for "HTTP - Get User Manager Details".
    c. In the **Parameters** tab of the **Outputs** section, verify that **Body** has a field for the "mail" attribute populated with the email address of the end user's manager.
6. If the status is "Failed":
    a. Open the history record.
    b. Review which connector had an error and investigate the details of the error by clicking the connector.

8.1.4 Checking FIDO Connector Logs

1. Sign in to the Azure portal.
2. Go to the **Resource Group** where the FIDO Connector App is deployed.
3. Select the **Container App**.
4. Select **Monitoring** > **Logs**.
5. Within the open tab, if not already selected, in the drop-down, change from "Simple mode" to "KQL mode" (using Kusto Query Language).
6. Paste a KQL query similar to the following to begin identifying errors and timeframes to investigate:

```
ContainerAppConsoleLogs_CL
| where Log_s contains "WARN" or Log_s contains "ERROR" or Log_s contains "Fail"
| project TimeGenerated, ContainerName_s, Log_s
```

Note: If the environment variables for the Container app have been modified, the container might need to be manually restarted to recognize the new variables.

CHAPTER

NINE

CREATING SHIPMENT REQUESTS

The method for creating shipment requests for pre-enrolled YubiKeys depends on how your FIDO Pre-reg solution is set up in your Customer Orchestration environment.

As an IT administrator, you can for example trigger a shipment request for a pre-enrolled YubiKey through your front-end IT Helpdesk system. Or, you can have some other integration process in your environment trigger the shipment request.

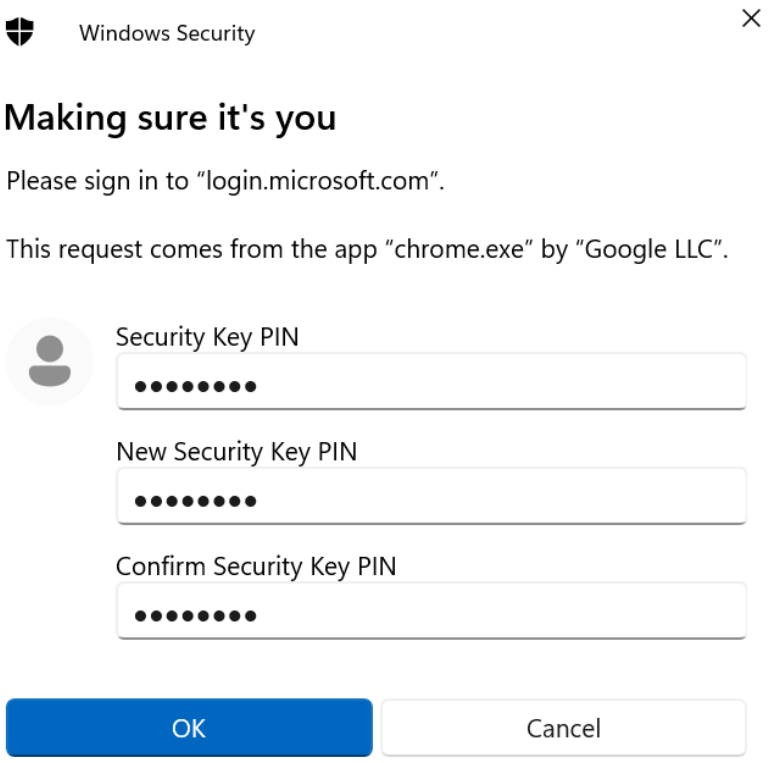
The shipment request is received by the Yubico FIDO Connector App which manages the credential encryption, requests recipient information from the customer's system, and creates a shipment request to the Delivery service. For more information, see *Process Flow* and API Reference.

Yubico receives a request for a pre-enrolled YubiKey. The request contains all information needed to program and ship the key. When the request is fulfilled and the credential is activated, the randomly generated PIN associated with the YubiKey is emailed to the end user.

Note: Once the credential is programmed onto the YubiKey, the challenge and credential data, including PIN, is purged from Yubico systems.

9.1 Initial Authentication

To authenticate with the identity provider, the end user presents their YubiKey and enters the provided PIN. If "Force PIN change" is set (and if supported by the platform), the end user is prompted to change the PIN when using the YubiKey for the first time, as in this example.

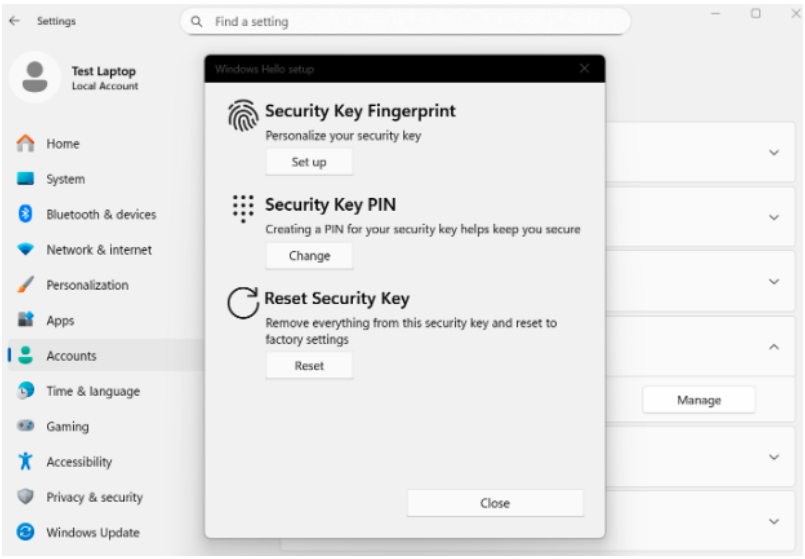


If "Force PIN change" was not set, the end user will be able to log in without changing the provided PIN, when using the YubiKey for the first time.

The previous step also applies when using a *YubiKey Bio (FIDO and Multi-Protocol Editions)* where the end user authenticates primarily using fingerprint(s) enrolled on the key (a PIN is required as fallback also when using fingerprint authentication).

Note: In most cases, the end user will *not* be automatically prompted to enroll a fingerprint when using a pre-enrolled YubiKey Bio the first time. Enrolling a fingerprint must be done by the end user in a separate step as described in the following.

Here is an example of how to enroll fingerprints when using *Windows 11*: Navigate to **Settings > Accounts > Sign-in options**, select **Security key**, and click **Manage**. Enter the PIN for the YubiKey Bio, and follow the on-screen instructions, which will prompt you to insert the security key and touch it to enroll a fingerprint. When done, you will be prompted to provide the fingerprint (instead of the PIN) when logging in.



Different platforms (device/OS/browser) will have different flows with regards to enrolling fingerprints on YubiKeys. Refer to the account security settings information for each platform for instructions on how to enroll fingerprints. For more examples of fingerprint enrollment, see YubiKey Bio Series Specifics.

Yubico Authenticator is a convenient tool that can also be used to enroll fingerprints on a YubiKey Bio. For instructions on how to install Yubico Authenticator and enroll fingerprints, see Install the App and Enroll a fingerprint.

Select the method that is applicable to your organization's IT platform when providing recommendations for your end users on how to enroll fingerprints on their pre-enrolled YubiKey Bio.

CHAPTER

TEN

MANAGING SHIPMENTS

When a shipment request for a pre-enrolled YubiKey has been successfully processed it will return a `shipmentId`. This shipment ID can be used in the YubiEnterprise Shipment API or the Customer Portal to track the request status and get shipping updates and possible error states.

The status for a shipment can be viewed in the Customer Portal for your organization. You can edit a shipment request, for example to update the recipient and address information or delivery type, as well as delete a shipment request if needed. For more information, see Managing Pre-reg Shipments.

CHAPTER

ELEVEN

POST-DEPLOYMENT OPERATIONS

The following describes useful runtime parameters and some important maintenance operations that are needed to avoid service interruption due to for example rotation of YubiEnterprise API tokens.

11.1 Runtime Environment Parameters

The following table lists useful parameters for the FIDO Connector App that you can change in the Azure Container if needed. To add or change these variables, navigate to the container's environment as described in *Configuring Environment Variables*.

The effective values of the various runtime environment variables can be obtained by the API, see Check Deployment Component Status in the API Reference. The Process Shipment job can also be run outside of the scheduled time to process shipments on demand by using the API, see the Process Shipments in the API Reference.

Note: A manual restart of the container might be needed for the updates to the environment variables to be recognized.

11.1.1 Common Runtime Parameters

Variable	Default value (if not changed by Environment Variable)	Comment
CRON_PROCESS_SHIPMENT_SCI	0 0 * * * *	The cron expression for the schedule that checks ongoing shipments and processes them. The default setting is to run every hour at the top of the hour. It can be changed to any valid cron expression, for example to run every 30 minutes: 0 */30 * * * *
LOGGING_LEVEL_COM_YUBICC	INFO	DEBUG
ENTRA_FIDO_API_CHALLENGE_ NUTES	20160	Numeric value representing time in minutes that the Entra FIDO challenge is valid for.
CRON_DATA_CLEANUP_COMPLI	0	The below cron schedule will purge ‘complete‘ shipment records that are older than this value. 0 = do nothing, no records will be deleted, the schedule is effectively disabled. n = integer value means delete ‘complete‘ shipment items older than today - n days.
CRON_DATA_CLEANUP_SCHEDU	0 30 * * * *	The cron expression for the cleanup schedule that purges ‘complete’ shipments and their secrets from the Azure table and vault. The default setting is to run at every 30 minutes past the hour.

11.1.2 PingOne PingID/Ping AIC Parameters

The environment variables for PingOne PingID and PingOne AIC are used by the FIDO Connector when interacting with components in the customer environment. For more information, see *Configuring Environment Variables*.

11.2 Configuring Key Vault and Storage Permissions

To be able to change the YubEnterprise API token or to view the shipments status table from the Azure portal, you need to configure appropriate user and firewall rules. The following describes how to set up these permissions in Key Vault and Storage.

Azure Key Vault Access configuration uses a permission model based on Azure RBAC (role-based access control). Access to Azure Key Vault is defined at a specific scope level by assigning appropriate Azure roles.

By default, the Resource Group Owner does not have permissions to view the contents of the Azure Key Vault. Therefore a special *Key Vault Secrets Officer* role can be assigned to a User or User Group for these to be able to access the Azure Key Vault.

To assign the "Key Vault Secrets Officer" role, do the following:

1. Sign in to the Azure Portal.
2. Go to your **Resource Group > Key Vault > Access control (IAM)**.
3. Add role assignment "Key Vault Secrets Officer".
4. Select a **User Group** or **User** to assign this role to.

As a user with this permission you can now go to **Resource Group > Key Vault > Secrets** and view the Key Vault Secrets.

In this reference implementation, Azure Vault and Storage public network access is disabled. To allow IT administrators access to Azure Vault or Storage, the IP address or range of your on-premises network must be allowed on the Azure Vault and Storage firewall settings.

To allow the on-premises network through the firewall on Azure Vault and Storage, do the following:

1. Sign in to the Azure Portal.
2. Go to your **Resource Group > Key Vault > Settings > Networking**.
3. On the tab **Firewalls and virtual networks**, scroll down to the **Firewall** section. Add your client IP address or range, for example "10.0.10.0". Microsoft trusted services are allowed by default, but you can disable "Allow trusted Microsoft services to bypass this firewall" depending on your organizations' preferences.
4. Go to your **Resource Group > Storage account > Security + networking > Networking**.
5. On the tab **Firewalls and virtual networks**, scroll down to the **Firewall** section. Add your client IP address or range, for example "10.0.10.0". Microsoft trusted services are allowed by default, but you can disable "Allow Azure services on the trusted services list to access this storage account" depending on your organizations' preferences.

11.3 Rotating API Tokens

YubiEnterprise API tokens expire one year after creation. Since a user (API caller) can have only one API token at a time, you must have a plan to roll over to a new API token before the old one expires.

To avoid service interruptions, it is recommended to regularly rotate the API tokens. The YubiEnterprise API token can be easily changed from the Key Vault objects without having to perform a complete deployment.

Note: Ensure that the user performing the API token rotation in Azure Key Vault has the *Key Vault Secrets Officer* role, see *Configuring Key Vault and Storage Permissions*.

To manage the API tokens, do the the following:

1. Sign in to the Azure Portal.
2. Go to your **Resource Group > Key Vault > Secrets**.
3. List the **YubiEnterprise API** tokens and click to view versions. Open and view the current version, add a new version, and disable the previous version. For information on how to manage API tokens, see Creating API Tokens.

11.4 Rotating Application Client Secrets

The Application Client Secret created as part of the application registration steps for *Yubico FIDO Connector App* have an expiration set by the administrator.

To perform regular rotation, the Microsoft Entra ID Administrator can also delete an existing Client Secret and create a new Client Secret to be used by the registered Application.

The `FIDO_Connector_Client_Secret` can be easily changed from the Key Vault objects without having to perform a complete deployment.

Note: Ensure that the user performing the API token rotation in Azure Key Vault has the *Key Vault Secrets Officer* role, see *Configuring Key Vault and Storage Permissions*.

To manage the Client Secret, do the the following:

1. Sign in to the Azure Portal.
2. Go to your **Resource Group > Key Vault > Secrets**.
3. List the **ENTRA-FIDO-API-CLIENT-SECRET** and click to view versions. Open and view the current version, add a new version, and disable the previous version.

11.5 Changing Microsoft 365 Email Account

The Microsoft 365 email account is used for example to send the FIDO Pre-reg PIN to end users. If needed, the email account can be changed as described in the following.

To change the Microsoft 365 email account, do the following:

1. Sign in to the Azure Portal.
2. Go to **Resource Group > Logic App**.
3. In the left menu, click **Development tools > API connections**.
4. Select **Microsoft 365**.
5. Go to **General > Edit API connection**.
6. Click **Authorize**.
7. Click **Authorize** again.
8. Sign in with the account that will be used as the sender of emails for FIDO Pre-reg PINs.
9. When logged in, click **Save**.

11.6 Resending PIN Email

The PIN email can be resent for a successfully completed shipment by using the API, see the API Reference.

CHAPTER

TWELVE

API REFERENCE

The various APIs used for creating requests for shipments of pre-enrolled YubiKeys, as well as requests for pre-programming of credentials, are described in more detail in the FIDO Pre-reg- FIDO Connector API Reference.

CHAPTER

THIRTEEN

RELEASE NOTES - FIDO CONNECTOR

For information about public releases of new features, resolved issues, and known limitations for new versions of the FIDO Pre-reg APIs and FIDO Connector app, see FIDO Pre-reg - FIDO Connector Release Notes.

CHAPTER

FOURTEEN

COPYRIGHT

14.1 Trademarks

Yubico and YubiKey are registered trademarks of Yubico AB. All other trademarks are the property of their respective owners.

14.2 Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design, and manufacturing. Yubico shall have no liability for any error or damages of any kind resulting from the use of this document.

All Yubico Software referenced in this document is licensed to you under the terms and conditions accompanying the software or as otherwise agreed between you or the company that you are representing.

14.3 Contact Information

Yubico AB
Gävlegatan 22
113 30 Stockholm
Sweden

14.4 Getting Help

Documentation is continuously updated on https://docs.yubico.com/ (this site). Additional support resources are available in the Yubico Knowledge Base.

Click the links to:

- Submit a support ticket for YubiKey as a Service and Customer Portal
- Submit other support requests
- Contact our sales team

14.5 Feedback

Yubico values and welcomes your feedback. If you think you may have discovered a flaw in our product, please submit a support request at https://support.yubico.com/s/requests and provide as much detail as you can.

14.6 Document Updated

2026-04-15 08:37:21 UTC