Yubico FIDO Pre-reg with Microsoft

Yubico

Sep 08, 2025

INTEGRATION GUIDE

CHAPTER

ONE

INTRODUCTION

Note: Yubico FIDO Pre-reg with Microsoft is currently in Early Access for identity provider Microsoft Entra ID. For more information, see Yubico FIDO Pre-reg.

With Yubico 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-registered YubiKeys for end users. The YubiKeys are pre-registered and shipped directly to the specific end user who received a randomly generated PIN separately.

The following sections describe how to integrate Yubico FIDO Pre-reg with Microsoft Entra ID. The instructions are intended for IT admins who are setting up shipments of pre-registered YubiKeys for their organization's end users in an environment using Microsoft Entra as IdP.

The instructions assume you have IT administration skills and knowledge of YubiEnterprise Delivery API, Microsoft Azure, and Entra ID. Listed tasks include steps performed both in the YubiEnterprise Console and Microsoft Azure/Entra ID. Refer to the Microsoft documentation for more details.

Important: Before you start implementing Yubico 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*.

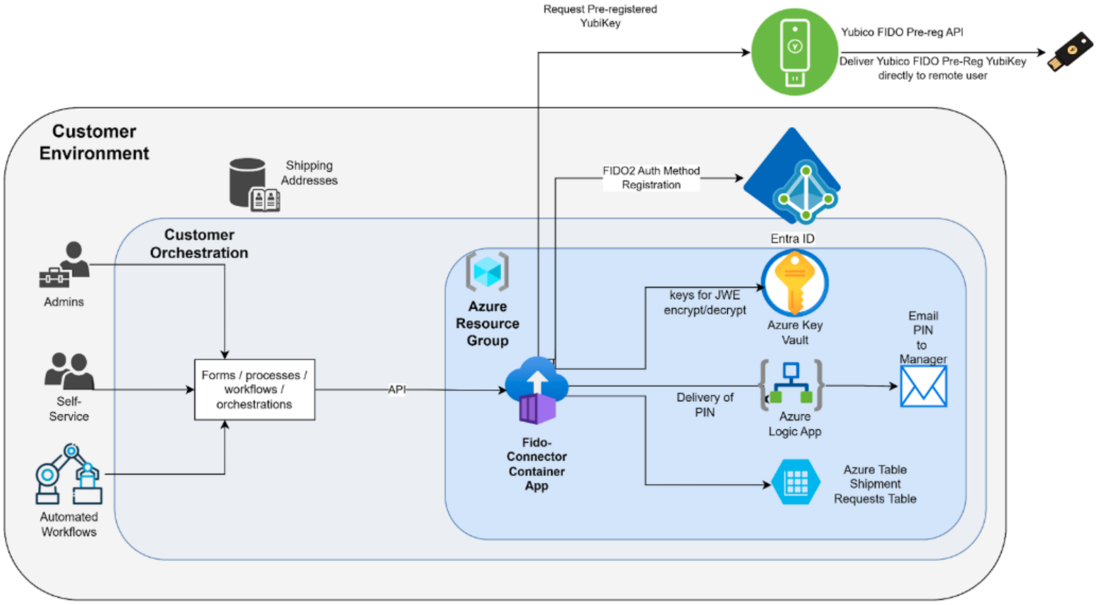
CHAPTER

TWO

ABOUT FIDO PRE-REG WITH MICROSOFT

The Yubico FIDO Pre-reg integration streamlines the deployment process with improved ease of use and enhanced security. End users receive a YubiKey, already pre-registered 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 based on Microsoft components.



Request Pre-registered YubiKey
Yubico FIDO Pre-reg API
Deliver Yubico FIDO Pre-Reg YubiKey directly to remote user
Customer Environment
Shipping Addresses
FIDO2 Auth Method Registration
Entra ID
Customer Orchestration
Admins
Azure Resource Group
keys for JWE encrypt/decrypt
Azure Key Vault
Email PIN to Manager
Self-Service
Forms / processes / workflows / orchestrations
API
Delivery of PIN
Azure Logic App
Fido-Connector Container App
Azure Table Shipment Requests Table
Automated Workflows

2.1 Process Flow

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

1. An authorized user (or process) triggers a YubiKey request for a user in Microsoft Entra ID via a front-end or IT Service Management (ITSM) orchestration platform.
2. The YubiKey request is received by the Yubico FIDO Connector App which is deployed on the customer infrastructure. The connector then makes a request over Microsoft Graph API to retrieve the necessary parameters required to create a device-bound passkey credential.
3. Microsoft Entra ID returns the passkey credential creation parameters for the target user to the Yubico FIDO Connector App which then encrypts the information as a *credential request*.
4. The Yubico FIDO Connector App creates a shipment request to the YubiEnterprise Delivery service including the form factor and shipping information, and attaches the encrypted credential request.

5. After passing through the YubiEnterprise Delivery service, Yubico decrypts the credential request and creates the credential (user private key) for the specified YubiKey form factor. The attestation response from the credential creation is then encrypted.
6. Yubico ships the YubiKey to the intended end user.
7. The Yubico FIDO Connector App continuously checks the YubiEnterprise Delivery service for updated shipment status.
8. When the shipment reaches status "Shipped" in the YubiEnterprise Delivery service, the Yubico FIDO Connector App captures the shipping information including tracking number, serial number, firmware version, and *encrypted* credential response and PIN.
9. The shipment status is updated in the customer's front-end system of choice.
10. The credential response is decrypted by the Yubico FIDO Connector app.
11. The YubiKey device-bound passkey credential (user public key) is registered in Microsoft Entra ID through the Microsoft Graph API.
12. The PIN is decrypted and provided to the customer's delivery system of choice.
13. The PIN is communicated to the targeted end user.
14. The end user authenticates to Microsoft Entra ID using their YubiKey and 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 Customer Orchestration 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 Microsoft Entra ID to initiate the registration of YubiKeys on behalf of end users.
- Interacts with Yubico APIs to request shipment of YubiKeys to end users.
- Communicates 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 customers, the orchestration requirements can be fulfilled using services like Azure Logic Apps, Azure Function Apps, 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 *Yubico 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 Yubico 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 easily called from forms, processes, workflows.
- Performs all interactions with the Microsoft Graph API for registering YubiKeys in a Microsoft Entra ID tenant.
- Performs all transport encryption before securely transmitting the credential information from the Customer Orchestration to the Yubico FIDO Pre-reg service.
- Keeps track of pending shipments and actively polls the Yubico FIDO Pre-reg service to check on status and updates to pending Yubico FIDO Pre-reg requests.
- Once the shipment request is ready, the app decrypts and verifies the authenticity of the response from the Yubico FIDO Pre-reg service.
- Completes the registration of the YubiKey by calling the Microsoft Graph API.
- Emails the PIN to the correct contact (end user's manager by default).

2.4 Yubico FIDO Pre-reg API

The Yubico 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.

The Yubico FIDO Pre-reg API is an extension of the YubiEnterprise API which is a cloud-based service facilitating the global distribution of YubiKeys. For more information, see YubiEnterprise API.

2.5 Security Features

The following provides an overview of security features of an implementation of FIDO Pre-reg with Microsoft.

2.5.1 Microsoft Entra ID Access

Yubico has no access to enroll and/or activate user passkey (FIDO2) credentials directly into a customer's Entra ID tenant. Instead, all interactions with Microsoft Entra ID are handled by the Customer Orchestration.

2.5.2 Pre-Registered Credentials

Since Yubico has no access to the customer's Microsoft Entra ID 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 Microsoft Entra ID.

2.5.3 PIN Provisioning

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

2.5.4 Transport Encryption

To mitigate the risk of exposing sensitive information, for example creation parameters, serial numbers, and PIN related to YubiKey assignments within the YubiEnterprise 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 Yubico FIDO Pre-reg with Microsoft Azure components and Entra ID to create a first shipment of a pre-registered YubiKey.

3.1 Prerequisites

Ensure you have the following before starting the implementation procedure:

- Enterprise Plus plan subscription. For questions about Yubico subscription services, contact Yubico Support.
- YubiEnterprise Console access with FIDO Pre-reg enabled. This is provided by Yubico during onboarding of your organization.
- Customization IDs (CID), Product IDs, and Subscription IDs for the YubiKey models you will be shipping to end users. Provided by Yubico.
- A YubiEnterprise API token, see Generating API Tokens.
- An ARM Template JSON file, provided by Yubico.
- A Docker Image for the Yubico FIDO Connector app, provided by Yubico.
- An Azure Resource Group permissions template provided by Yubico.
- The following administrative roles are required for the implementation:
  - *Application Administrator* - when registering apps (Microsoft Entra ID).
  - *Authentication Policy Administrator* - when enabling passkey authentication (Microsoft Entra ID).
  - *Global Administrator* - when registering apps and granting admin consent for tenant (Microsoft Entra ID).
  - *Privileged Role Administrator* - when granting Logic App permissions (Azure deployment).

3.2 Integration Steps

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

1. *Configure required Azure permissions*
2. *Configure authentication and register apps in Microsoft Entra ID*
3. *Deploy apps and infrastructure components in Azure*
4. *Test and verify the Azure deployment*

5. *Create your first pre-registered YubiKey shipment request*

The sections in the following describe each step in detail.

CHAPTER

FOUR

CONFIGURING AZURE PERMISSIONS

In this step you will add permissions required for developers that will deploy and configure the applications in Azure.

When adding the permissions, use one of the following options:

- Use an existing account with the required permissions as described in *Prerequisites*.
- Create a Resource Group and add a custom role using the Azure Resource Group *predefined permissions template* provided by Yubico. The steps to create the group are described in the following.

4.1 Creating a Resource Group

If not already available, you must first create an Azure Resource Group to be able to add the required user permissions.

To create a Resource group, do the following:

1. Log in to the Azure Portal.
2. Search for and select “Resource groups”.
3. Click **Create**.
4. Select the appropriate Subscription and Region, and provide a descriptive Resource group name, for example “Yubico FIDO Pre-reg Service”.
5. Click **Review + create**.

4.2 Adding a Custom Role

To add a custom role with the required permissions, do the following:

1. In the **Azure portal**, create a custom role with the permissions from the *predefined permissions template* scoped to the previously created Resource group.
2. When the custom role is created, assign the new “Privileged administrator role” to the user or the security group that is deploying the resources.

Note: The “Microsoft.Authorization/roleAssignments/write” permission results in the new role being a “Privileged administrator role”.

4.3 Assigning an Email License

To support the PIN mailing function, the designated sender account will need to have the required licensing. The setup in this example uses the Microsoft 365 email service. If you want to use a different email service, you can update the "Send_shipment_pin Logic App flow" after the deployment to use your preferred delivery service.

To assign an Microsoft 365 license to the account, do the following:

1. Log in to the Microsoft 365 admin center.
2. Go to **Billing > Licenses** and assign a license granting access to Microsoft 365. If your organization requires additional licenses you might need to reach out to your Billing Account Owner or Billing Account Contributor.

CHAPTER

FIVE

CONFIGURING MICROSOFT ENTRA ID

In this step you will configure Microsoft Entra ID for authentication and authorization management.

To complete all configuration steps described in this section, the following roles are required:

- *Application Administrator*
- *Authentication Policy Administrator*
- *Global Administrator*
- *Privileged Role Administrator*

5.1 Enabling Passkey (FIDO2) Authentication

In this step you will configure the authentication methods policies used in Microsoft Entra ID.

Note: To complete these configuration steps you must have either the Authentication Policy Administrator or the Global Administrator role.

To configure the Microsoft Entra ID policies to allow the use of YubiKeys, do the following:

1. Log in to the Microsoft Entra admin center with *at least the Authentication Policy Administrator* role.
2. Go to **Protection > Authentication methods > Policies**.
3. Under the method **Passkey (FIDO2)**, set the toggle to “Enable”. Select “All users” or “Add groups” to select specific groups. Only security groups are supported (you cannot use dynamic groups or individual users).
4. **Save** the configuration.

The **Configure** tab has additional settings to control the type of passkeys supported in the customer tenant, and their registration requirements:

- **Allow self-service set up:** Must be set to “Yes”. If this is disabled, YubiKeys cannot be registered, not even using administrative registration processes.
- **Enforce attestation:** Recommended setting “Yes”. Using cryptographic evidence attestation ensures that registered authenticators are genuine YubiKeys and not fraudulent products or low-assurance passkey credentials (which might not be able to support attestation).
- **Enforce key restrictions:** Recommended setting “Yes”. This lets your organization allowlist specific YubiKey models by their associated Authenticator Attestation GUID (AAGUID). For more information, see YubiKey hardware FIDO2 AAGUIDs.

Important: If security keys such as device-bound passkeys or other types of passkeys are already used in your Microsoft Entra ID environment, ensure that these configuration changes do not break the sign-in for existing users.

For more information, see Enable passkeys (FIDO2) for your organization (Microsoft documentation).

5.2 Registering Apps

In this step you will register the Yubico FIDO Connector App and the Yubico FIDO Pre-reg Test Client (optional).

An app must be registered to allow the app itself to connect to the Microsoft Graph API, and to allow other clients such as Entra ID IGA, test clients, ServiceNow and other custom applications, to connect to the app to invoke requests.

It is recommended that any forms, processes, and workflows used to call the Yubico FIDO Connector App follow a similar registration pattern with distinct credentials as described in the following.

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

5.2.1 Yubico FIDO Connector App

To register the Yubico FIDO Connector App, do the following:

1. Log in to the Microsoft Entra admin center and go to **Applications** > **App registrations**.
2. Click **+ New registration**.
3. Provide a descriptive name, for example "Yubico FIDO Pre-reg Service", and click **Register**.
4. Under **Manage**, click **API permissions**.
5. Click **+ Add a permission**.
6. Select "Microsoft Graph".
7. Click **Application permissions**.
8. Search for "UserAuthMethod-Passkey.ReadWrite.All" and select the permission.
9. Click **Add permissions**.
10. Next to the list of permissions, select "Grant admin consent for {tenant name}".

    Note: The *Global Administrator* role is required for this step.

11. Under **Manage**, click **Expose an API**.
12. Click **Add** next to the **Application ID URI**.
13. 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 populated as an ARM template parameter `FIDO_Connector_Allowed_Audiences`. The URI does not need to be resolvable, but should have a descriptive scope name.

14. Click **+ Add a scope** and set the following:
    - **Scope name:** "create_request"
    - **Display name fields:** "create_request"
    - **Description fields:** "Allows Yubico FIDO Pre-reg requests"
15. Click **Add scope**.
16. Under **Manage**, click **Certificates & secrets**.
17. Click **+ New client secret**.
18. Provide a name for your client secret and accept the recommended expiration.
19. Click **Add**.
20. Copy the client secret. This will be used in the ARM template as `FIDO_Connector_Client_Secret`.
21. Go to **Overview** and copy the **Application (client) ID** value. This will be used in the ARM template as `FIDO_Connector_Client_Id`.

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

5.2.2 Yubico FIDO Pre-reg Test Client

Registering this app is *optional*. However, the app is useful when testing direct calls to the Yubico 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 Yubico FIDO Pre-reg Test Client app, do the following:

1. Log in to the Microsoft Entra admin center and go to **Applications** > **App registrations**.
2. Click **+ New registration**.
3. Provide a descriptive name like "Yubico FIDO Pre-reg Test Client" and click **Register**.
4. Under **Manage**, click **Certificates & secrets**.
5. Click **+ New client secret**.
6. Provide a name for your client secret and accept the recommended expiration.
7. 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

SIX

DEPLOYING TO AZURE

In this step you will deploy the Yubico FIDO Connector App itself along with the underlying infrastructure and required configuration changes.

6.1 Prerequisites

Before you start the deployment, ensure you have the following:

- Access to a YubiEnterprise Console organization with FIDO Pre-reg enabled, along with a YubiEnterprise API token. See Generating API Tokens.
- An ARM Template JSON file, provided by Yubico.
- A Docker Image for the Container app, provided by Yubico. The Docker image contains the Registry name/password used in the deployment.
- Completed all steps in the *Configuring Microsoft Entra ID*. This includes developer permissions to deploy Azure services, along with FIDO policies, as well as *App registrations*.

6.2 Deployment Steps

Follow these steps to deploy the components in Azure:

1. *Deploy the ARM template*
2. *Modify private endpoints for Azure Vault and Storage*
3. *Configure the Container app*
4. *Grant permissions to the Container app*
5. *Grant permissions to the Logic App*
6. *Authorize Logic App to use Office 365 connector*

Each step is described in detail in the following.

6.2.1 Deploying ARM Template

To deploy the ARM template, do the following:

1. Log in to the Azure portal.
2. In the **Home** view, 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 your Azure Subscription.
   - **Resource group:** Select or create a resource group for this deployment.
   - **Region:** Leave as default, all resources are deployed to the local region of the resource group.
   - **YED_API_TOKEN:** Paste in the token generated in *Prerequisites*.
   - **Key Vault_Resource_Name:** Provide a unique name for your key vault instance.
   - **Container_App_Name:** Provide a unique name for your container app.
   - **Container_Registry_Name:** Use the Registry name from *Prerequisites*.
   - **Container_Image_Name_Tag:** Use the Registry Container Image name and version Tag from *Prerequisites*.
   - **Container_Registry_User:** Use the Registry user name from *Prerequisites*.
   - **Container_Registry_Password:** Use the Registry password from *Prerequisites*.
   - **FIDO_Connector_Client_Id:** Client ID from the *app registration*.
   - **FIDO_Connector_Client_Secret:** Client Secret from the *app registration*.
   - **FIDO_Connector_Allowed_Audiences:** List of scopes/audiences that a client application must use for calling the app’s API. The default value used earlier was `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:** List of app registrations that are allowed to call this app’s API, as registered in *client 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"]`.
   - **Storage Account_Resource_Name:** Provide a unique name for your storage instance.
   - **Workflows_Send_shipment_pin_name:** Leave as default, or enter a name based on your preferred naming convention.
7. Click **Review + create**.
8. When the validation completes, click **Create** and wait for your application to deploy.

Note: The following parameters in the ARM template have appropriate predefined values for standard Microsoft Azure deployments. They do not need to be changed unless specifically advised by your IT department, for example for government deployments:

- `MS_Login_Online_Endpoint`
- `MS_Graph_Endpoint`

- `Azure_Mgmt_Endpoint`
- `Azure_Storage`
- `Azure_Vault`

6.2.2 Modifying Private Endpoints

The ARM template includes a reference implementation of a virtual network, subnet and private endpoints for Azure Vault and Storage resources used by the FIDO Connector Container App. You can use this reference as a basis to further modify as per the network settings of your environment. The predefined values of the following parameters typically do not need to be changed.

- `virtualNetworkName`
- `virtualNetworkAddressPrefix`
- `subnetName`
- `subnetAddressPrefix`
- `keyVaultPrivateEndpointName`
- `tableStorageAccountPrivateEndpointName`

6.2.3 Configuring Container App

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

1. In your **Container App** resource, go to **Application** > **Containers**.
2. Click **Edit and deploy**.
3. In the **Properties** tab, set the **Image source** to “Docker Hub or other registries”.
4. In the **Container** tab, click **yubicofidopreregcontainer** in the **Container Image** section.
5. On the **Properties** tab, for **Image source** select “Docker Hub or Other Registries”.
6. Click **Environment Variables**.
7. Set **SEND_PIN_URL** as follows:
    a. Go to your **Resource Group**.
    b. Open the logic app resource **Send_shipment_pin**.
    c. Copy the value “Workflow URL”.
    d. Paste it into the **SEND_PIN_URL** value field.
    e. Click **Save**.
8. Click **Create**.
9. Wait for your application to instantiate.

6.2.4 Granting Container App Permissions

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

To configure the managed identity for the Container app, do the following:

1. In your **Container App** resource, go to **Settings** > **Identity**.
2. Click **Azure role assignments**.
3. Click **Add role assignment** and apply the following values:
    a. **Scope:** Key Vault.
    b. **Subscription:** Your subscription.
    c. **Resource:** The Key Vault deployed by this project.
    d. **Role:** Key Vault Administrator.
4. Click **Save**.
5. Click **Add role assignment** and configure as follows:
    a. **Scope:** Storage.
    b. **Subscription:** Your subscription.
    c. **Resource:** The Storage Account deployed by this project.
    d. **Role:** Storage Table Data Contributor.
6. Click **Save**.

6.2.5 Granting Logic App Permissions

Note: These configuration steps require either the *Privileged Role Administrator* or *Global Administrator* roles.

To add authorization for the Send_shipment_pin Logic App to call the Microsoft Graph API, do the following:

1. In the **Send_shipment_pin** Logic App, go to the resource group where the **Send_shipment_pin** Logic App was deployed.
2. Select the “Send_shipment_pin” Logic App.
3. Go to **Settings** > **Identity**.
4. Copy the value for **Object (principal) ID**.
5. Go to **Entra ID** in the **Azure portal**.
6. Go to **Manage** > **Role and administrators**.
7. Select the role “Directory Readers”.
8. Click **+ Add assignments**.
9. Under **Select members**, select “No member selected”.
10. In the search field, paste the “Object (principal) ID” copied from step 4.
11. Select the Enterprise Application displayed and click **Select**.

12. Click **Next**.
13. Ensure the **Assignment type** is selected as “Active”.
14. Enter a justification and click **Assign**.

6.2.6 Authorizing Office 365 Usage

To authorize the Send_shipment_pin Logic App to use the Office 365 connector, do the following:

1. In the **Send_shipment_pin** Logic App, go to the resource group where the **Send_shipment_pin** Logic App was deployed.
2. Select the “Send_shipment_pin” Logic App.
3. Go to **Development tools** > **API connections**.
4. Select the “office365” connection.
5. Go to **General** > **Edit API connection**.
6. Click **Authorize**.
7. Log in with the account that will be used as sender of Yubico FIDO Pre-reg PIN emails.
8. When logged in, click **Save**.

CHAPTER

SEVEN

TESTING THE DEPLOYMENT

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

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

1. Go to the previously created *Yubico 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.

7.1 Troubleshooting

The following provides basic troubleshooting steps for common deployment issues.

7.1.1 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 YubiEnterprise Console, see Generating API Tokens.
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.

7.1.2 Verifying Shipment Status in Storage Browser

1. Log 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*. You can find more details to understand the error in *Checking FIDO Connector Logs*.

7.1.3 Verifying Delivery Status of YubiKey PIN

By default Yubico 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. Log 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.

7.1.4 Verifying YubiKey Registration in Microsoft Entra ID

1. Log in to the Microsoft Entra admin center.
2. Go to **Users** > **All users**.
3. Search for the desired **User**.
4. Go to **Authentication method**.
5. Verify that the new YubiKey is listed.

7.1.5 Checking Microsoft Entra ID Audit Log History

1. Log in to the Microsoft Entra admin center.
2. Go to **Users** > **All users**.
3. Search for the desired **User**.
4. Go to **Audit logs**.
5. Filter the **Activity** column for each of the following:
   a. "Get passkey creation options".
   b. "Admin registered security info".
   c. "User registered security info".
6. Check if any of the events indicate that an error occurred.

Note: If an error related to Microsoft Entra ID is encountered by the FIDO Connector App, the `error_message` in the **fprshipments** table, or error entry in FIDO Connector App logs, will contain a `client-request-id` which is related to the "Correlation ID" in Microsoft Entra ID Audit Logs.

7.1.6 Checking FIDO Connector Logs

1. Log 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
```

CHAPTER

EIGHT

CREATING SHIPMENT REQUESTS

The method for creating shipment requests for pre-registered YubiKeys depends on how your Yubico 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-registered YubiKey through your front-end system, for example ServiceNow. 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 YubiEnterprise Delivery service. For more information, see *Process Flow* and *API Reference*.

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

CHAPTER

NINE

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.

9.1 Additional 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 Container App*.

Variable	Default value (if not changed by Environment Variable)	Possible values
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.

The effective values of the various runtime environment variables can be obtained by the API *Check Deployment Component Status*.

The Process Shipment job can also be run outside of the scheduled time to process shipments on demand by using the API *Process Shipments*.

9.2 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. Log 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. Log 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.

9.3 Rotating API Tokens

YubiEnterprise API tokens expire one year after generation. 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. Log 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 retrieve API tokens, see Generating API Tokens.

9.4 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. Log 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.

9.5 Changing Microsoft 365 Email Account

The Microsoft 365 email account is used for example to send the Yubico 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. Log 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. Log in with the account that will be used as the sender of emails for Yubico FIDO Pre-reg PINs.
9. When logged in, click **Save**.

9.6 Resending PIN Email

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

CHAPTER

TEN

API REFERENCE

Each deployment of the Yubico FIDO Connector will have its own instance of the API described in the following.

Base URL: `URL provided by the Container App` This URL will be dependent on the URL provided by your container app service, and will be unique for each deployment.

10.1 Check Deployment Component Status

`GET /v1/status`

Provides the status of deployment components. As part of the testing, you can first do a call to `/v1/status` to verify that the API is operational and the client can connect to it. It is also a way to ensure that some of the key properties provided during deployment are set.

Response: On success HTTP 200. Response body:

```
{
 "AZURE_TABLES_ENDPOINT": "string",
 "AZURE_TABLES_SHIPMENTS_TABLE_NAME": "string",
 "AZURE_KEY_VAULT_ENDPOINT": "string",
 "AZURE_TENANT_ID": "string",
 "CRON_PROCESS_SHIPMENT_SCHEDULE": "string",
 "CRON_DATA_CLEANUP_SCHEDULE": "string",
 "CRON_DATA_CLEANUP_COMPLETED_DAYS": "string",
 "EMAIL_API_SEND_ENDPOINT": "string",
 "ENTRA_FIDO_API_CLIENT_ID": "string",
 "ENTRA_FIDO_API_VERSION": "string",
 "ENTRA_FIDO_API_CHALLENGE_TIMEOUT_MINUTES": "string",
 "FIDO_CONNECTOR_VERSION": "string"
 "LOGGING_LEVEL_COM_YUBICO": "string",
 "YE_API_BASE_URL": "string",
 "YE_JWKS_SIGN_ENDPOINT": "string",
 "YE_JWKS_TRANSPORT_ENDPOINT": "string"
}
```

10.2 Create Shipment Request

`POST /v1/fpr/shipments`

Provides the ability to place a request for shipment of pre-registered YubiKeys.

`user_id` can be provided either as Object ID, for example `"user_id": "123456-abc-123456-xyz"`, or as UPN (User Principal Name), for example `"user_id": "username@yubico123.sample.com"`.

`reseller_organization_id` is the optional Base58 Organization ID for the reseller that a shipment's inventory was purchased through. It can be omitted or sent as an empty string if no value needs to be provided. All other fields in request are required.

Input value references:

- Character limits for yubicoShipmentRequest
- Values for product_id and inventory_product_id
- Finding a customization_id in the Enterprise Console
- YubiEnterprise API Reference

Request:

```
{
 "user_id": "Either User Principal Name (UPN) or Object ID",
 "pin_request": {
     "type": "generate",
     "length": 8, //value can be between 4 and 63, inclusive
 },
 "yubico_shipment_request": {
     "reseller_organization_id": "Optional ID for the reseller",
     "delivery_type": 1,
     "address_validation_bypass": false,
     "recipient": {
         "recipient_company": "Company name",
         "recipient_email": "Email address", //Should be email to receive PIN, not␣
↪principle object name
         "recipient_firstname": "First name",
         "recipient_lastname": "Last name",
         "recipient_telephone": "5555555555"
     },
     "mailing_address": {
         "street_line1": "Street address",
         "street_line2": "Apt / unit #",
         "city": "City",
         "region": "2 char state",
         "postal_code": "Postal code",
         "country_code_2": "2 char country code"
     },
     "shipment_items": [
         {
             "product_id": 1, //YubiKey model ID
             "inventory_product_id": 18, //Subscription ID
             "product_quantity": 1, //# of keys to include
             "customization_id": "CUSTID" //Customization ID
```

(continues on next page)

(continued from previous page)

```
        }
      ]
   }
}
```

Response:

- On success:
  - HTTP 201
  - Response body: Created `shipment_id` from YubiEnterprise Delivery service

    `{"data":{"shipment_id":"String"}}`
- On error:
  - HTTP 401 Unauthorized
  - HTTP 400 Bad Request
  - Bad request, response body examples:

    `{"error_code":"ye_error","error_message":"Validation error when creating YED shipment","error_data":{"code":"validation_error","message":"Input for Last Name exceeded limit of 20 characters"}}`

    ``{"error_code":"api_error","error_message":"PIN `length` must be between 4 and 63","error_data":{"error_type":"validation"}}``

    `{"error_code":"idp_error","error_message":"Could not find user:  7dc95e2f-53-... "}`

10.3 Get Shipment Request Status

`GET /v1/fpr/shipments/{shipment_id}`

Provides the ability to get the processing state of a `shipment_id` created through the Create Shipment Request API.

The request in the FIDO Connector App has two distinct states: “ongoing” and “complete”. The states are described in more detail in the following.

shipment_state	Description
ongoing	The request has been created in YubiEnterprise with a Shipment ID. Fulfillment operations and credential creation are in progress with MS Entra ID.
complete	Response from YubiEnterprise has been received, the credential has been created on the YubiKey and successfully registered with MS Entra ID.

If a processing error has been encountered it will be saved in the **fprshipments** table and returned by the API. Details about the encountered error are provided in `error_kind` and `error_message` as described in the following.

Error	Description
error_kind	This field will contain a string value "GENERAL" if an error has been encountered during processing.
error_message	This field will contain a string value that has the detailed error message returned by YubiEnterprise or MS Entra ID.

Response:

- On success:
  - HTTP 200
  - Response body for a shipment request *without errors*:

```
{
 "shipment_id": "string",
 "shipment_state": "string"
}
```

  - Response body for a shipment request *with processing errors*:

```
{
 "shipment_id": "string",
```

(continues on next page)

(continued from previous page)

```
 "shipment_state": "string",
 "error_kind": "string",
 "error_message": "string"
}
```

10.4 Resend PIN

`GET /v1/operations/resend-pin/{shipment_id}`

Provides the ability to resend the PIN email for a `shipment_id`. For a shipment request in `complete` status, this operation retrieves the PIN response from the YubiEnterprise Delivery service and decrypts it to resend the PIN email.

Response: On success HTTP 204, no content body.

10.5 Process Shipments

`GET /v1/operations/process-shipments`

Provides the ability to trigger on-demand the process of retrieving shipment responses from the YubiEnterprise Delivery service and process them. This API is useful if shipment processing needs to be run right away instead of waiting for the scheduled job.

Response: On success HTTP 204, no content body.

CHAPTER

ELEVEN

RELEASE NOTES - FIDO CONNECTOR

The following lists public releases of new features, resolved issues, and known limitations for new versions of the *Yubico FIDO Connector App*.

11.1 Release 1.1.4 (22 August 2025)

New Features & Enhancements

- **Get user using UPN (User Principal Name):** Previously, in the JSON request the `user_id` was provided as Object ID. To enhance the user experience, the input now allows `user_id` to be provided either as Object ID, for example `"user_id": "123456-abc-123456-xyz"`, or as UPN, for example `"user_id": "username@yubico123.sample.com"`.
- **Connector container app version:** Functionality has been updated to show the application version. `/v1/status` now returns the `"FIDO_CONNECTOR_VERSION": "string"` field displaying the version of the FIDO Connector application software.

Resolved Issues

- **Connector API response:** Previously the API was returning a “500 Internal Server Error” message, for example when an invalid YubiEnterprise API token was used. To provide more guidance to the user, the error message has been changed to show a “401 Unauthorized” error message instead.

11.2 Release 1.1.1 (25 June 2025)

New Features & Enhancements

- **PIN length changed to 4-63.** The validation in the API accepts a PIN length value between 4 to 63, inclusive. This means you can enter any number from 4 up to 63 (including either 4 or 63) as PIN length. See *API Reference*.
- **Support for address validation override.** An `address_validation_bypass` flag (true/false) has been added to the API. If set to “true”, the API will accept the provided address without further validation. See Address Validation.

11.3 Release 1.0.0 (3 April 2025)

First release of the Yubico FIDO Connector App. The application is deployed to a Microsoft Azure subscription and handles most of the Customer Orchestration complexities. For more information about included features, see *Yubico FIDO Connector App*.

CHAPTER

TWELVE

COPYRIGHT

12.1 Trademarks

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

12.2 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.

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.

12.3 Contact Information

Yubico AB
Kungsgatan 44
111 35 Stockholm
Sweden

More options for getting touch with us are available on the Contact page of Yubico's website.

12.4 Document Updated

2025-09-08 10:31:53 UTC