

Integrate Azure DevOps with Secrets Safe

Azure DevOps is a comprehensive set of development tools and services provided by Microsoft that supports collaboration and streamlines the software development lifecycle. It offers a range of features and capabilities to facilitate communication, planning, coding, testing, and deployment of software projects.

The Azure DevOps extension allows for the retrieval of ASCII secrets from an instance of Secrets Safe.



Note: The Secrets Safe Azure DevOps extension supports retrieval of secrets from BeyondInsight/Password Safe versions 23.1 or greater.

For this extension to retrieve a secret for use in each Azure DevOps pipeline, the Secrets Safe instance must be preconfigured with the secret in question and an account must be authorized to read it.

Configure Password Safe to Allow Azure DevOps to Retrieve Secrets

The following sections outline how to set up Password Safe to allow Azure DevOps to retrieve ASCII secrets.



Note: The following setup is meant to be a quick start guide to help with Password Safe setup. For more information, please see the [Password Safe Administration Guide](https://www.beyondtrust.com/docs/beyondinsight-password-safe/ps/admin/index.htm) at <https://www.beyondtrust.com/docs/beyondinsight-password-safe/ps/admin/index.htm>.

Set Up API Key and User Account


1. Create an API registration in BeyondInsight (does not require a user password).
2. Create or use an existing Secrets Safe group.
3. Create or use an existing BeyondInsight user.
4. Add the API registration to the group.
5. Add the user to the group.
6. Add the Secrets Safe feature to the group.

Managed Accounts Setup

1. Create or use an existing access policy that has the **View Password Auto Approve** option set.
2. Add the **All Managed Accounts Smart Group** to the **BeyondInsight** group.
3. Add the Access Policy to the **All Managed Accounts Smart Group** role, and ensure that both requestor and approver are set.
4. Create or use an existing managed system.
5. Create or use an existing managed account associated with the managed system.
6. Configure the managed account with the **API Enabled** and **Max Concurrent Requests Unlimited** options checked.

Configure Azure DevOps

The following sections outline how to set up Azure DevOps to retrieve ASCII secrets.

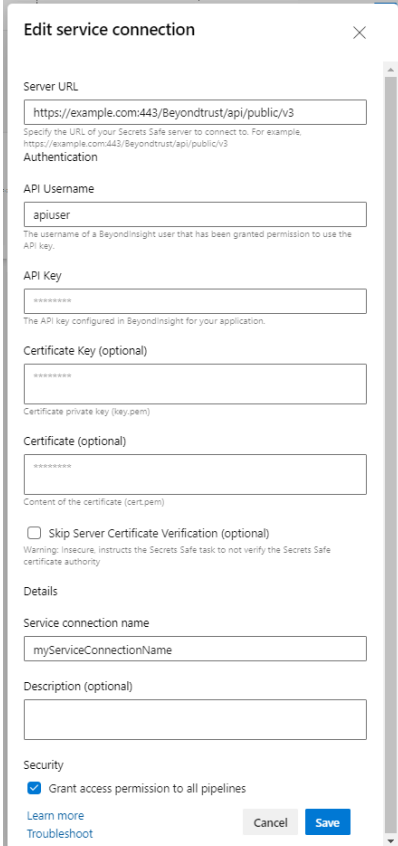
 **Note:** The following setup is meant to be a quick start guide to help with Azure DevOps setup. For more information, please see [Manage Service Connections at https://learn.microsoft.com/en-us/azure/devops/pipelines/library/service-endpoints?view=azure-devops&tabs=yaml](https://learn.microsoft.com/en-us/azure/devops/pipelines/library/service-endpoints?view=azure-devops&tabs=yaml).

Configure Secrets Safe Service Connection

Configure a service connection for the Secrets Safe pipeline task.

Service Connection Fields:

- **Server URL:** The URL for the Secrets Safe instance from which to request a secret.
- **API Username:** The username of a BeyondInsight user that has been granted permission to use the API key for the API request to the Secrets Safe instance.
- **API Key:** The API key configured in BeyondInsight for your application. For use when authenticating to Secrets Safe.
- **Certificate Key (optional):** Certificate private key (**key.pem**). For use when authenticating with an API key using a client certificate. See ["Extract Client Certificate and Key"](#) on page 2 section below.
- **Certificate (optional):** Content of the certificate (**cert.pem**) for use when authenticating with an API key using a client certificate.
- **Skip Server Certificate Verification (optional):** Indicates whether to verify the certificate authority on the Secrets Safe instance. For use when authenticating to Secrets Safe.



Edit service connection

Server URL

Specify the URL of your Secrets Safe server to connect to. For example, https://example.com:443/Beyondtrust/api/public/v3

Authentication

API Username

The username of a BeyondInsight user that has been granted permission to use the API key.

API Key

The API key configured in BeyondInsight for your application.

Certificate Key (optional)

Certificate private key (key.pem)

Certificate (optional)

Content of the certificate (cert.pem)

Skip Server Certificate Verification (optional)
Warning: Insecure. Instructs the Secrets Safe task to not verify the Secrets Safe certificate authority.

Details

Service connection name

Description (optional)

Security

Grant access permission to all pipelines

[Learn more](#) [Troubleshoot](#)

Extract Client Certificate and Key

Download the PFX certificate from Secrets Safe and extract the certificate and the key to be pasted into the service connection.


Example: Extract Certificate and Key

```
openssl pkcs12 -in client_certificate.pfx -nocerts -out ps_key.pem -nodes
openssl pkcs12 -in client_certificate.pfx -clcerts -nokeys -out ps_cert.pem
```

Copy all text from the **ps_key.pem** file to the service connection **Certificate Key** field. Copy all text from the **ps_cert.pem** to the service connection **Certificate** field.

Configure Secrets Safe Secret Task

A task must be configured to retrieve secrets from Secrets Safe.

Pick the service connection, and then enter the path and title of the requested secret. Specify the name of the pipeline variable to populate. The pipeline variable is created and set at runtime by the task.


IMPORTANT!

Secret retrieval types are base64 encoded only. Reuse of the variable name in multiple tasks overwrites the existing secret.

When using the variable in the pipeline the variable must be decoded.


Example: Decode Variable

```
$(variableFromPipeLine) | base64 --decode
```

Task Fields:

- **Retrieval Type:**
 - **Secret:** Secrets Safe types (credential, text, file) base64 encoded.
 - **Secret Path:** Path to the Secrets Safe secret. For example, *folder1/folder2*.
 - **Secret Title:** Title of the Secrets Safe secret found at the path specified above.
- **Secrets Safe service connection:** Select the Secrets Safe service connection.
- **Pipeline Variable Name:** A pipeline variable created and set at runtime to contain your retrieved secret. Reuse of the variable name in multiple tasks overwrites the existing secret.

← Secrets Safe Secret ⓘ

Retrieval Type *

Secret (base64 encoded)

Managed Account

Secrets Safe service connection * ⓘ

Secret Path * ⓘ

Secret Title * ⓘ

Pipeline Variable Name * ⓘ

**IMPORTANT!**

Do not log the secret in the pipeline log after you base64 decode the secret. Security-minded engineers should review pipeline composition before changes are run with access to secrets.



Note: Binary files are not supported.

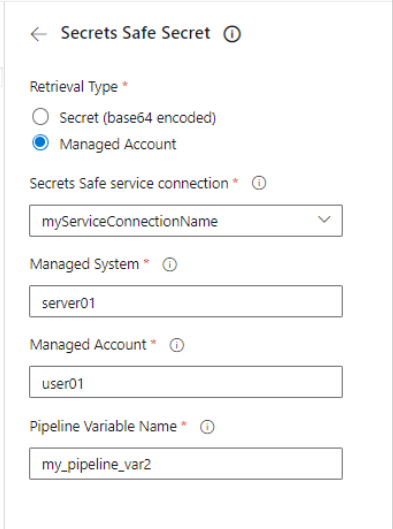
Configure Secrets Safe Secret Managed Account

A task must be configured to retrieve managed account secrets from Secrets Safe.

Pick the service connection, and then enter the managed system and account for the requested secret. Specify the name of the pipeline variable to populate. The pipeline variable is created and set at runtime by the task. It contains your retrieved secret. Reuse of the variable name in multiple tasks overwrites the existing secret.

Task Fields:

- **Retrieval Type:**
 - **Managed Account:** Password Safe type account associated with a system.
 - **Managed System:** System managed by Password Safe.
 - **Managed Account:** Account associated with the managed system.
- **Secrets Safe service connection:** Select the Secrets Safe service connection.
- **Pipeline Variable Name:** A pipeline variable created and set at runtime to contain your retrieved secret (not encoded). Reuse of the variable name in multiple tasks overwrites the existing secret.



← Secrets Safe Secret ⓘ

Retrieval Type *

Secret (base64 encoded)

Managed Account

Secrets Safe service connection * ⓘ

myServiceConnectionName ▾

Managed System * ⓘ

server01

Managed Account * ⓘ

user01

Pipeline Variable Name * ⓘ

my_pipeline_var2