

Swimlane Installation and Deployment Guide

CyberArk Integration

Table of Contents

- Downloading the Integration File
- Create a New Task Input/Output
- Upload and Manage Integrations.

Downloading the Integration File

From your internet browser, navigate to the Swimlane integration repository at <https://apphub.swimlane.com/>. From there, you will login, find the search bar, and type in CyberArk. This will pull up the CyberArk integration along with an option in the top right to download it. Click this and save it to a location you will remember.

Appendix: Swimlane Resources

Swimlane Bundle Repository: <https://apphub.swimlane.com/>

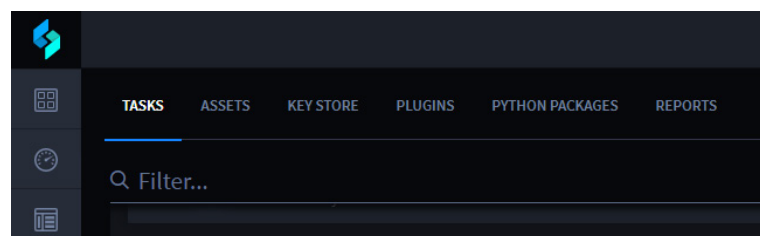
Swimlane Support Portal - <https://support.swimlane.com/support/login>

Swimlane Community: <https://www.secopshub.com/>

Swimlane Website: <https://swimlane.com/>

Integrations

Returning back to the Swimlane platform, integrations is where you create, view and manage your tasks, assets, key store, and plugins. Integration activities are for administrators of Swimlane.



From the Integrations page, you can:

- Create, edit, and delete tasks.
- Create, edit, and delete assets.
- Create, edit, and delete keys.
- Install plugins that extend Swimlane with new task and asset options.
- Upload existing scripts or tasks.
- View reports.

Create or Edit a Task

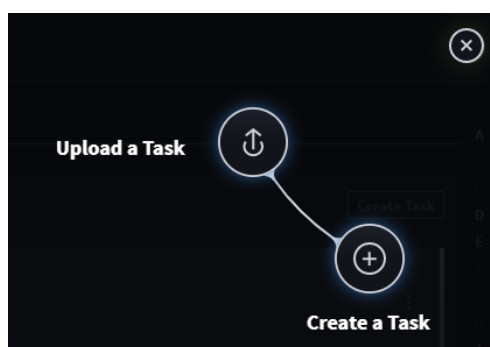
A Swimlane task is an action or series of actions that consist of input, configuration details, output, and triggers. Tasks in Swimlane can be very complex and can contain multiple input and output variables.

Use Swimlane's task creation process to fine-tune the tasks that you want an application to run.

Create a New Task

To create a new task:

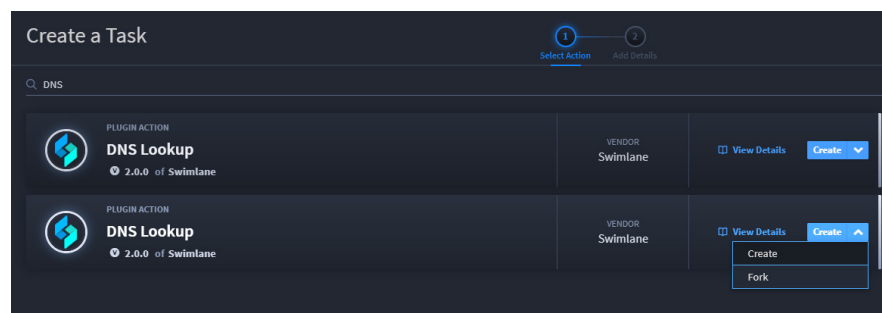
1. From the Integrations, Tasks page, click **+**, or **New Task**, and then select **Create a Task**.



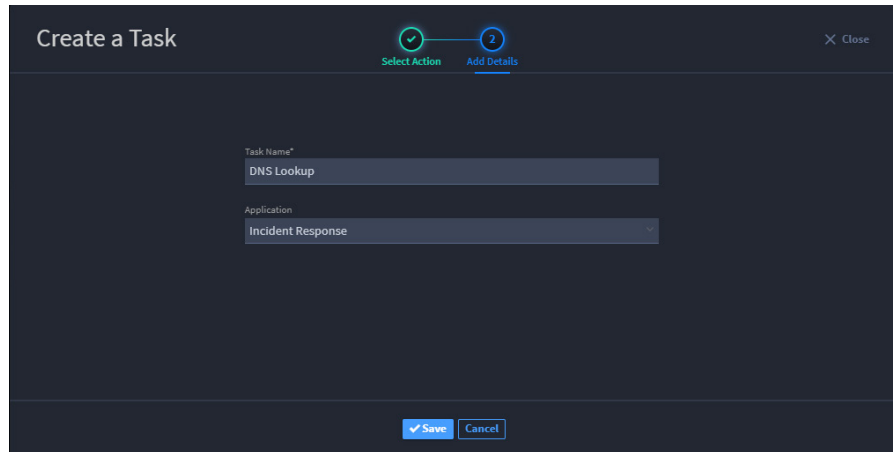
Alternatively, scroll down the page until you see your application and then click **Create Task**.

2. On Create a Task, Swimlane lists plugin actions. You can filter by vendor (on the right-side of the page), and/or type a task, plugin action, or vendor into the filter field. Once you find your desired task type, click **Create**.

If your task is eligible to be forked, you can also select **Fork** when creating your new task. A forked task is copied as an editable Python script.

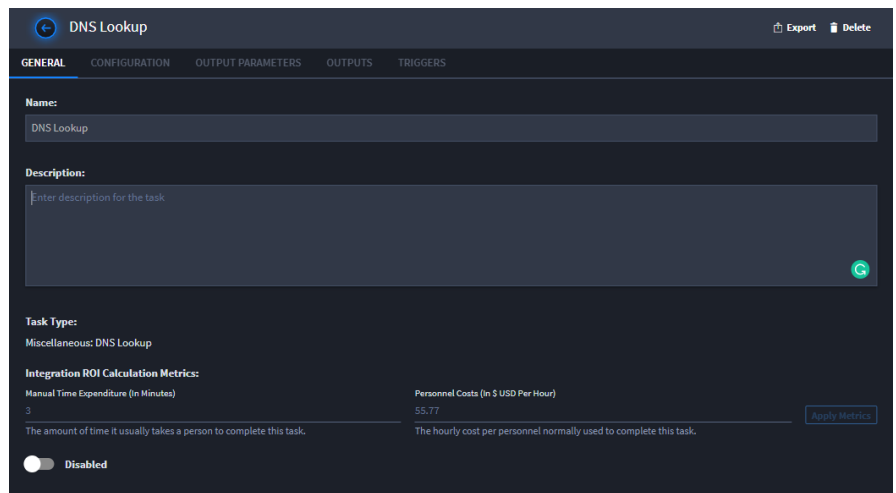


- Next, Swimlane prompts you to name your task. You can also associate it with a related application.



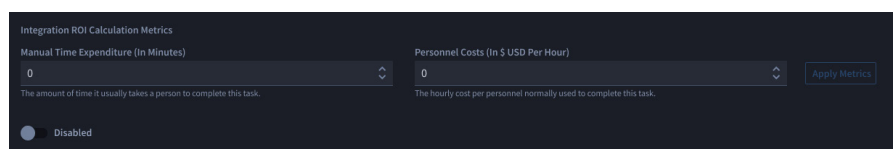
You must provide a unique name for your task in order to save it.

- On the Integration, **General** tab, you can rename the task, provide a description, set the appropriate ROI usage metrics, and enable or disable the task.



 **Note:** By default, a task is enabled upon creation.

- You can also set up Usage Metrics for this task. Usage Metrics track the time and cost you are saving by automating this task. To set Usage Metrics, under Integration ROI Calculation Metrics fill out the **Manual Time Expenditure** and **Personnel Costs** fields so that the value of the automation is calculated in Usage Metrics.



Now you are ready to begin configuring your task.



Configure a Task

Click the Integration, **Configuration** tab to configure the task.

The details that you configure for your task vary depending upon the type of task that you create. Most tasks have a set of inputs that need to be configured in a form but others, for example Python or Powershell scripts, require a text editor.

In addition, you configure the outputs as well. Swimlane's automapping helps you step through the process of output mapping.

For detailed information about input and output configuration see:

- [Configure Task Input](#)
- [Configure Task Output](#)

Set Triggers

Click the Integrations, **Triggers** tab to specify the conditions that will cause this task to run.

Tasks can be triggered by the following conditions:

- **Scheduled** - Task runs by a specified time frame.
- **Email** - Task runs when a specified email is received.
- **Record Save** - Task runs whenever a specified record is saved in the associated application.
- **Integration** - Task runs when an integration runs.

You can specify to trigger scheduled tasks yearly, monthly, weekly, daily, hourly, or minutely.

You can also schedule a custom trigger with a valid cron expression.

The screenshot shows the 'Triggers' configuration tab in the Swimlane interface. It contains four rows of trigger settings. Each row has a 'Type' dropdown menu set to 'Scheduled'. To the right of each dropdown is a 'Triggered' section with a set of frequency buttons: 'Minutely', 'Hourly', 'Daily', 'Weekly', 'Monthly', 'Yearly', and 'Custom'. The first row has 'Monthly' selected and shows a date picker set to 'At 12:00 AM, on day 1 of the month, only in January'. The second row has 'Hourly' selected and shows 'At 0 minutes past the hour'. The third row has 'Daily' selected and shows 'At 12:00 AM'. The fourth row has 'Monthly' selected and shows 'At 12:00 AM, on day 12 of the month'. Below each date/time picker is a small note: 'Scheduled triggers use UTC time'. At the bottom left of the configuration area is a blue button labeled 'Save Triggers'.

Tasks can also be referenced from workflow or from an integration button in Swimlane's Application Builder. The Workflow tab shows any workflow actions that are associated with the current task. Click **Add/Edit Workflow** to associate a task with a new workflow action. The Manual tab displays any layout objects associated to the task in Application Builder. Click **Open Application Builder** to edit the application.



Note: Tasks that are triggered by an integration to update a specific value in a record are set to only affect the value once. This was designed so that integration record value changes do not run in a continual loop.



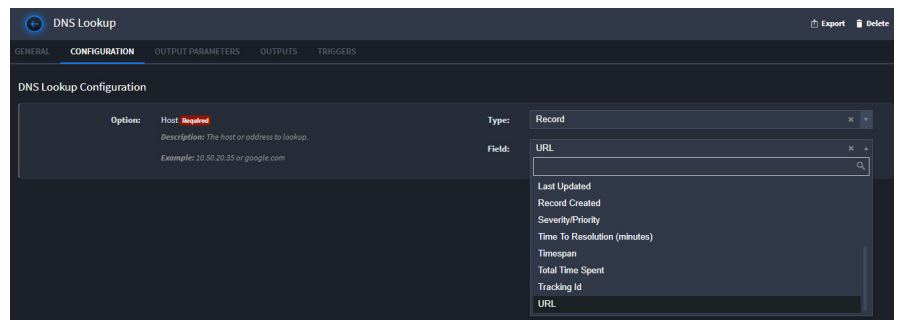
Configure Task Input

Task input types and parameters contain the information that you want your task to ingest. You configure task input parameters differently depending upon the types of input. For example, script and API input mapping differs from mapping input parameters for plugin or email fields.

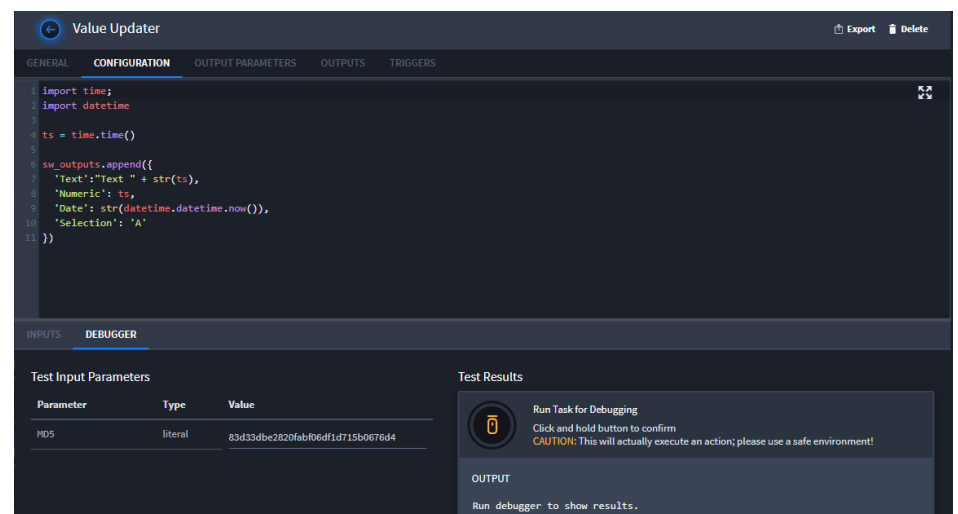
Map Plugin or Email Input Field Parameters

When you are configuring plugin or email fields, you are prompted to configure with specified values. To specify the type of value, select from the following input types:

- **Trigger** - The task runs based on specified triggering events.
When you select this option, the data from the trigger, for example, the Date Time Executed or Trigger Type, can be used as a parameter for the task.
- **Record** - The task runs with fields from a specified record used to pass parameters to the task.
- **Key Store** - The task runs with values from your key store. The values are used to pass parameters to the task. You specify the keys that will pass the values.
- **Asset Library** - The task runs with values from an identified asset's library. The values are used to pass parameters to the task. You specify keys that will pass the values.
- **Static Value** - The task runs with a hard-coded, static value passed to the task.
- **Link** - The task runs with Swimlane links as parameters.



For email fields, you can also indicate the state you want the emails to be in after the task runs. You can select between read, unread, or delete.



You can set conditions for when email will import for the task. The conditions check for text, specified in regex, in the email subject, body, or the email sender.



Note: When you add more than one condition, Swimlane processes them using AND logic.

Map Script or API Input Field Parameters

Specify script and third-party or API inputs on the Configuration tab. The configuration of the tab varies depending up on the type of input you are working with.

For Windows Swimlane installations, Powershell scripting allows further configuration where you can specify Windows-based arguments and a working directory.

Python scripting allows you to include multiple input variables.

You can also script based on the assets available through a third party such as Splunk, ThreatConnect, etc.

For more information about Swimlane-specific coding, see [Python Task API](#).

Test Scripts with Debugger

You can test the code in your script by running it through the Debugger. Click the **Debugger** subtab, then click **Run** to check your script.

You can also assign input types and variables here.

Results of your script are written to the results window of the debugger.

The screenshot displays the Swimlane Debugger interface. The top section shows a Python script with the following code:

```
1 import random
2 import re
3 import time
4
5 time.sleep(3)
6 good_threat_score = 0
7 bad_threat_score = [95, 75, 60, 85, 90, 10, 15, 20, 25, 30]
8
9 md5 = sw_context.inputs["MD5"]
10
11 if re.search('9BD8E415B23C164D581361B3B21D6CA8', md5):
12     result = random.choice(bad_threat_score)
13
14 else:
15     result = good_threat_score
16
17 sw_outputs.append({"score": result})
18
```

The bottom section is divided into two tabs: "Inputs" and "Debugger". The "Debugger" tab is active, showing a table of parameters and a results window.

Name	Type	Value	Run
MD5	Record: Attachment MD5s		

The "Results" window displays the following traceback (most recent call last):

```
File "C:\Program Files\Swimlane\PythonQueue\58941c8fb39b0a0f08bb
if re.search('9BD8E415B23C164D581361B3B21D6CA8', md5):
File "C:\Python27\lib\re.py", line 146, in search
return _compile(pattern, flags).search(string)
TypeError: expected string or buffer
```



Note: This utility will run your actual code so be aware of what you've scripted when running through the debugger.



Review Task Configuration Examples

The following screen shows an example of the Configuration tab for a third-party plugin (or bundle):

The screenshot shows the 'Configuration' tab for a 'Slack Broadcast Configuration' task. It features three configuration sections, each with an 'Option', 'Description', 'Example', 'Type', and 'Literal' field.

Option	Description	Example	Type	Literal
Channel	The channel to broadcast the message	#incidents	Static Value	incidents
Title	The title of the message	New Incident	Static Value	Malicious URL Hit
Message	The message body	IR-23 was created on 1/22/25	Link	Record

You configure the fields according to how the plugin is set up. Some plugins will have many fields and options for configuration.

This screen shows an example of the **Configuration** tab for a Data Creation task type:

The screenshot shows the 'Configuration' tab for an 'Email Import Configuration' task. It includes an 'Email Asset' dropdown set to 'IMAP Demo', a 'State After Import' section with radio buttons for 'Read', 'Unread', and 'Delete', and a 'Conditions' section with a note: 'No condition(s) added. Conditions enable you to conditionally run the import process.'

This example deals with Email, which means that you will need to specify how to handle the records after you've imported an email as part of the task. You specify whether the email will appear read, unread, or whether you want the task to delete the email. (You will need to setup the task further to have the task delete the email.) You can also assign conditions, which allow you to run import processes.

The following screen shows an example of the **Configuration** tab for a script task type:

The screenshot shows the 'Configuration' tab for a script task. It features a code editor with Python code for sending an email via SMTP. Below the code editor is the 'Inputs' section, which includes a 'Script Inputs' area with a list of parameters.

```
1 import smtplib
2 from email.mime.text import MIMEText
3
4 recipients = sw_context.inputs["RECIPIENT"]
5
6 sender = 'customer.support@swimlane.com'
7 sender_pwd = 'UR4SwimlaneCustomerSupport!'
8
9 smtpserver = smtplib.SMTP("smtp.gmail.com", 587)
10 smtpserver.ehlo()
11 smtpserver.starttls()
12 smtpserver.ehlo()
13 smtpserver.login(sender, sender_pwd)
14
15 body = "The Acme CIRT has confirmed that the suspicious e-mail you forwarded is NOT malicious.\n\nThank you for reporting this."
16
17
```

Script Inputs

Variable	Type	Field
RECIPIENT	Record	Submitted By





The tab opens with a script editor. See [Python Task API](#) and [Rest API](#) in the Swimlane Developers Guide for more information.

You can expand the script editor to a full screen view. Click the  icon to expand the editor. To return to the Configuration tab view, with the cursor in the editor, press Esc.



Note: The Swimlane script editor uses a variation of Python editors like PyCharm. Swimlane’s editor uses 2

Review, Discover, and Add Output Parameters

If your task relies on a plugin or asset, you can review the parameters, names, and descriptions that go along with that plugin or asset on the Output Parameters tab.

General Configuration Output Parameters Outputs Triggers		
39 Parameters Defined		
Retrieve a ip address report from a scan		
Parameter	Name	Description
country	Country	Origin Country of IP
as_owner	Owner	Owner of IP
resolutions_hostname	Resolved Hostnames	List of all the resolved hostnames
resolutions_last_resolved	Last Resolved Dates	List of all the dates for the last resolved hostnames
detected_communicating_samples_date	Detected Communicating Samples Dates	Latest 100 files submitted to VirusTotal detected dates
detected_communicating_samples_positives	Detected Communicating Samples Positives	Latest 100 files submitted to VirusTotal that are detected by one or more antivirus solutions and communicate with the IP address provided when executed in a sandboxed environment.
detected_communicating_samples_sha256	Detected Communicating Samples SHA256	Latest 100 files submitted to VirusTotal detected SHA256s
detected_communicating_samples_total	Detected Communicating Samples Total	Latest 100 files submitted to VirusTotal totals
undetected_communicating_samples_date	Undetected Communicating Samples Dates	Latest 100 files submitted to VirusTotal undetected dates

Discover and Add Parameters for Scripted Tasks

The Output Parameters tab is also where you review and add parameters for scripted tasks.



Important! Your script must define outputs in order for them to be discovered.

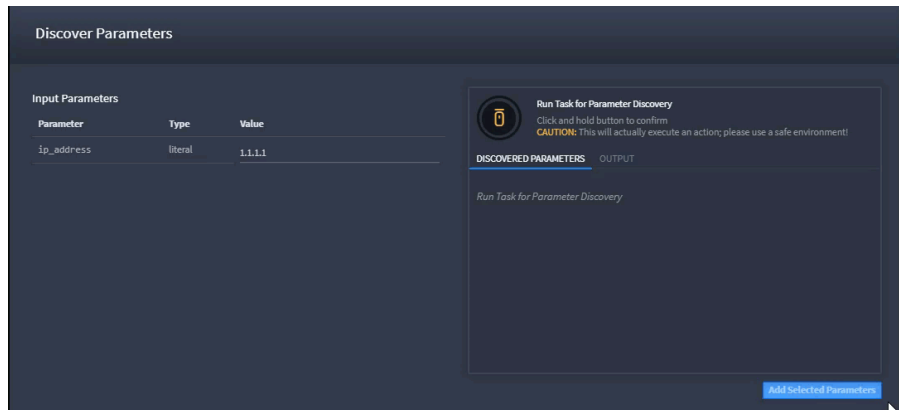
To discover and add parameters for scripted tasks:

1. Create or open a task that is dependent upon a script, then navigate to the Output Parameters tab. On the Output Parameters tab, click **Discover Parameters**.

General Configuration Output Parameters Outputs Triggers		
No Output Parameters Defined		
Execute Python scripts		
Discover Parameters		
Parameter	Name	Description
No parameters defined		
Add new parameter with name Output Parameter		



- Next, you will run the task and discover its parameters. Click and hold the **Run Task for Parameter Discovery** button.

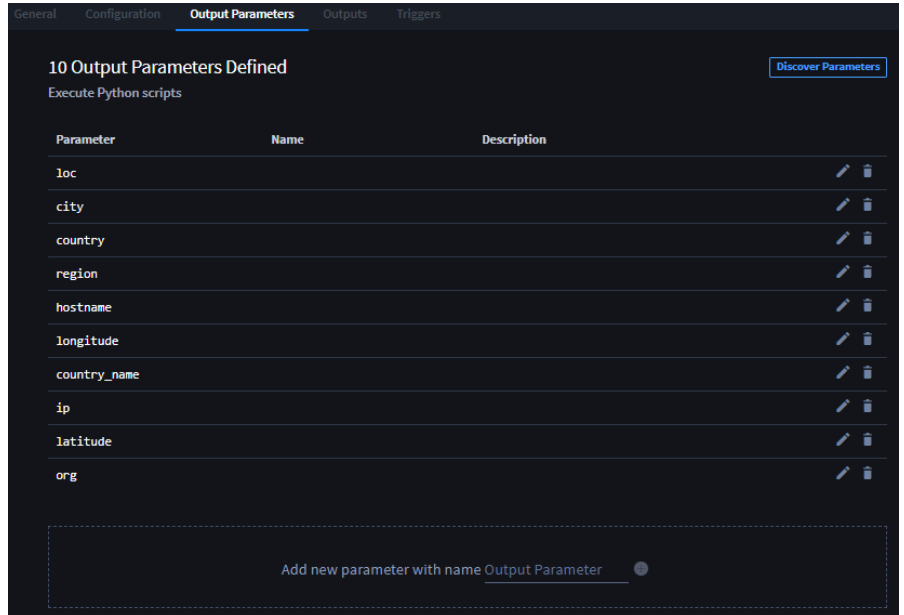


The task runs, which allows Swimlane to discover and index the available parameters for the task.



Important! The task actually executes, so make sure you're running the task in a safe environment!

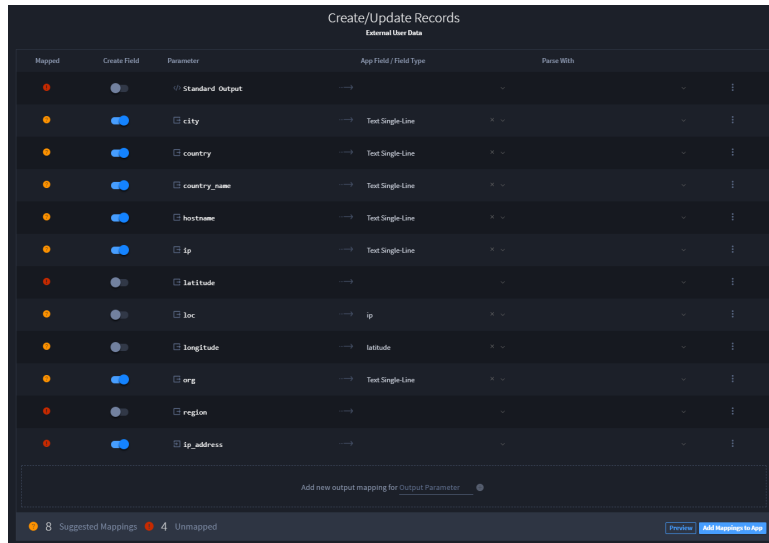
- You now have a list of defined parameters. Review the list and add new parameters as needed.



Next, you'll map the output parameters for the task.



- Click the Outputs tab. The parameters you've just discovered and added are now available to be created and mapped to in your task.



Map Output for Creating or Updating Records

Select Create/Update Records when you have a number of fields and parameters that you need to map for task output.

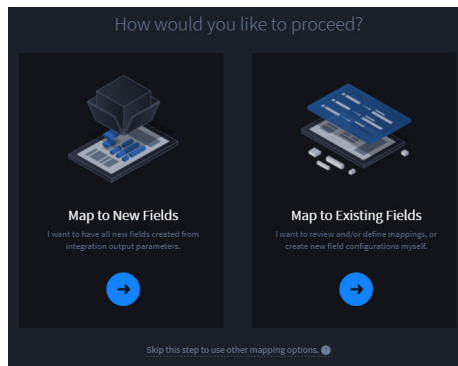
To map output for Create/Update Records or Update Current Record:

- From the Outputs tab, click **Create/Update Records** then, on the next page, specify the application for the task.



Note: You can also create an application from the Which app would you like to use? page prompt. For more information see [Create an Application for Output Mapping](#).

- The first time you create a task this way, you will see this mapping decision dialog asking you, *How would you like to proceed?*



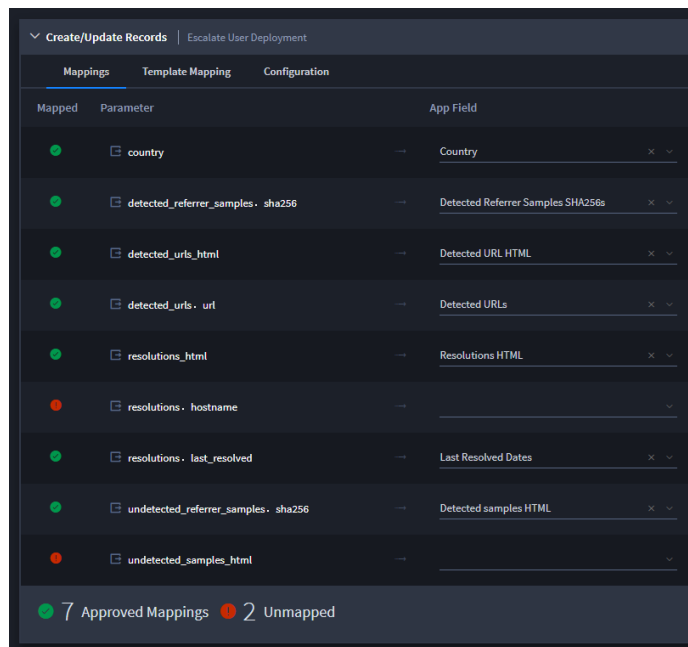
Follow the guidance on the dialog to continue.

Swimlane's automapping feature works to create a list of matching parameters and fields that you can review, configure, and approve on the next screen.

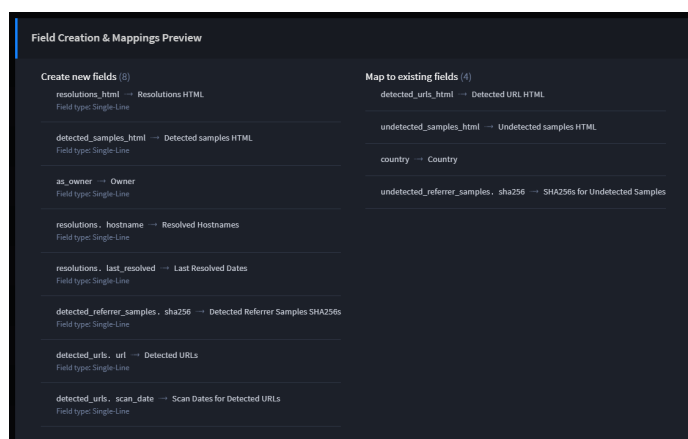
- On Create/Update Records, review the suggested mappings.



- If you want Swimlane to create a field with the output parameter and mapped field, select the **Create Field** switch.
- Select the Application Field or Field Type.
- If applicable, indicate what you will use to parse the output data.
- To configure and map additional parameters, fields, or field types, click the elipsis that follows the **Parse With** field and select **Add Configuration**.
- While you're working on your mappings, you can keep track with the mapping status bar at the bottom of the window.



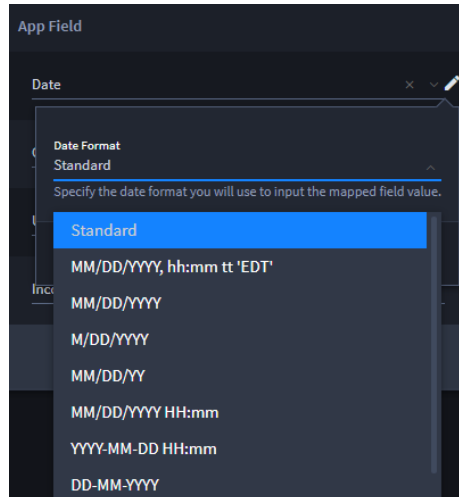
- Review and save your work as you continue to create approved mappings. Click Preview to get an at-a-glance view of the currently updated mappings.



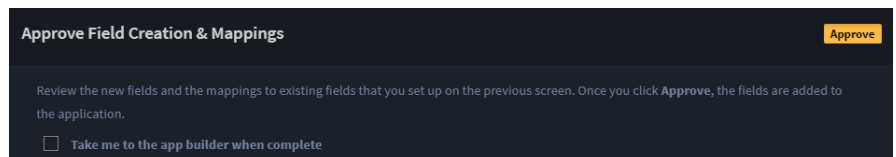
Note: As you set parameters for date or user/group fields, you can specify the format you'd prefer that the date or name be mapped in the record.



For example, you have the following options for dates/times:



10. Click **Add Mappings to App** to go to Approve Field Creation & Mappings. Review, and then click **Approve**.



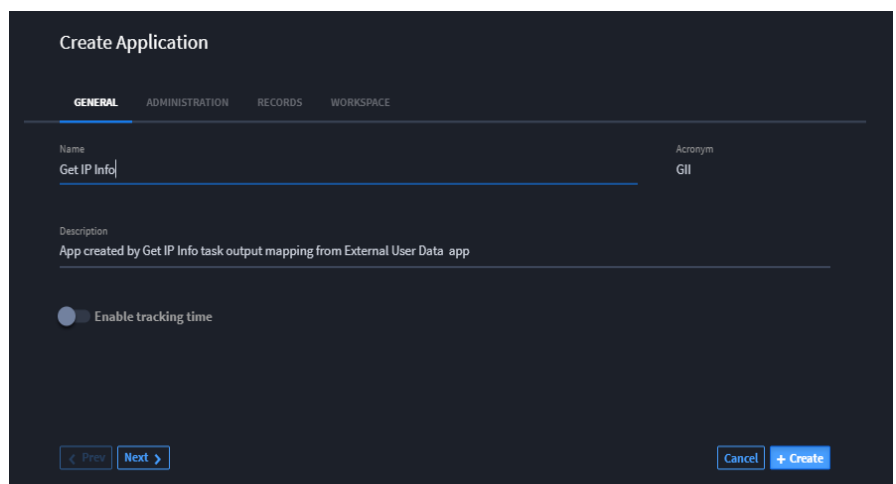
11. Once you've approved your mappings, you return to the task view. Click **Save** to save the task.

Create an Application for Output Mapping

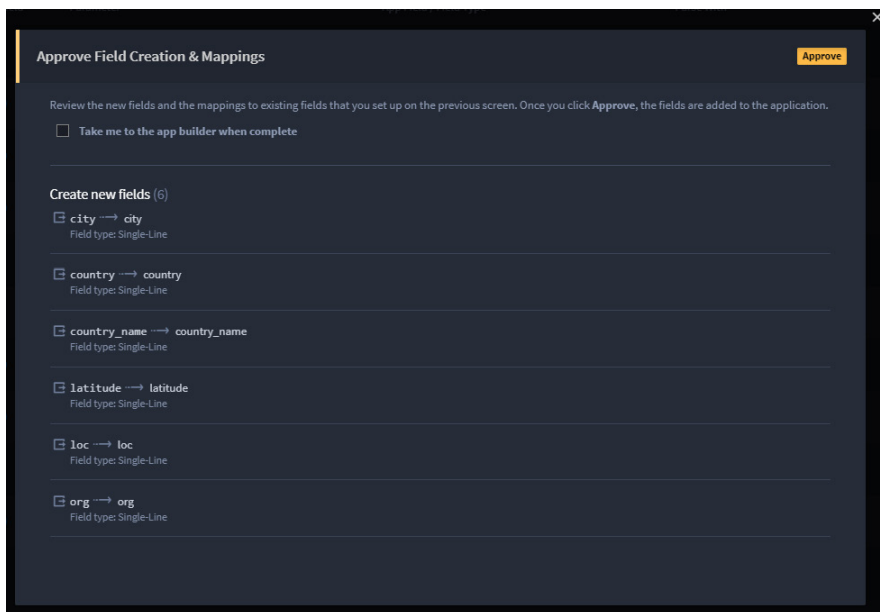
When mapping to create or update records, you can also create an application on the fly and create fields to map to.

To create an application for output mapping:

1. From the Outputs tab, click **Create/Update Records** then, on the next page, click **Create New Application**.



2. Assign your application a name and other details as necessary, and then click **+Create**.
3. On Create/Update Records, select the **Create Field** switch to create fields for your application. When done, click **Add Mappings to App**.



Save the task.

Template Mapping Sub-Tab

On the Template Mapping sub-tab, you can also specify a template at this time, depending upon what you want the task to do. Template Mapping uses [Mustache templating language syntax](#) that can combine multiple output values into one composite-string of text.

Here is a template example:

```
Malicious Scores: static analysis result: {{static_score}}, dynamic analysis result: {{dynamic_score}}
```

Use `{{#newline}}` to add extra lines and/or line breaks into your template.

Configuration Sub-Tab

Go to the Configuration sub-tab to make further decisions about how to update records with output from tasks. You control how the records coming out of the task relate to the records within Swimlane by setting the Creation Type, Error Handling, Key Field, and Reference. Hover over the help icons on this subtab for more information, or consult this table:



Task Outputs Configuration Options

Configuration Setting	Option	Definition
Creation Type	Insert Only	Only inserts new records. Does not update existing records.
	Insert and/or Replace	Creates a new record, or replaces an existing record.
	Insert and/or Update	Creates a new record, or updates an existing record.
Error Handling	Set as empty	If an error occurs during updating this field, set the field to an empty value.
	Skip Row	If an error occurs during updating this field, skip the entire record.
	Stop Import	If an error occurs during updating this field, stop the import process.
Key Field		The field which uniquely identifies each record. Existing records will be matched based on the value of this field.
Reference		If your task uses references, you make additional specifications here.
	Add reference on source record	Click to update the source reference field with the created or updated records from this mapping.
	Source Reference Field	The source record's reference field. This field contains any eligible reference fields from the source application that you can select.

Here's an example of the Output Mapping Configuration tab:

The screenshot shows a configuration interface for a task named 'Create/Update Records' under 'Acme Security Operations'. The 'Configuration' tab is selected, showing options for 'CREATION TYPE' (radio buttons for 'Insert only', 'Insert and/or Replace', and 'Insert and/or Update'), 'ERROR HANDLING' (radio buttons for 'Set as empty', 'Skip Row', and 'Stop Import'), a 'Key Field' dropdown, and a 'REFERENCE' section with a 'Source Reference Field' dropdown and a 'Scanned Values' list.

Map Other Types of Output

You can also map the output of a task to update a record in process, send an email, save data to file, or to execute another task.

Map Output to Update Current Record

Select Update Current Record when you want a task to update a record in process, while the task runs.

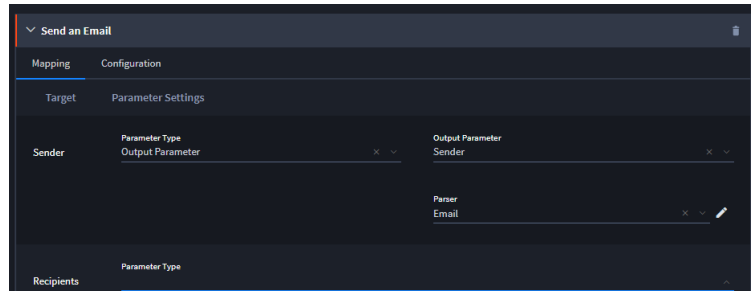
The process for mapping output to update a record in process is the same as mapping output for creating or updating records. See [Map Output for Creating or Updating Records](#) for details.



Map Output to Send in an Email

To map output for Send an Email:

1. From the Outputs tab, click **Send an Email**.

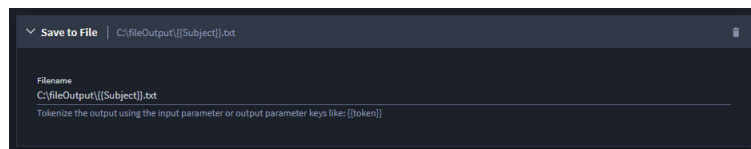


2. Review the Email Target fields, and specify the parameters from these options. The available parameters depend upon the email asset you select.
3. If applicable, indicate what you will use to parse the output data.
4. Continue with all email target fields and parameters, then click **Save**.

Map Output to Save to File

To map output for Save to File:

1. From the Outputs tab, click **Save to File**.



2. Specify the location where you want Swimlane to save the data, then click **Save**.

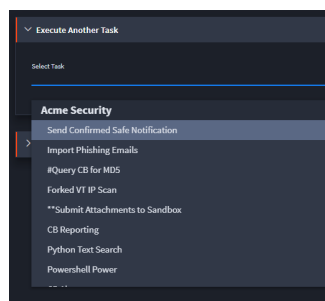


Note: Swimlane strips invalid characters from Save to File output. Invalid characters include / * ? < > " |.

Map Output to Execute Another Task

To map output for **Execute Another Task**:

1. From the Outputs tab, click **Execute Another Task**.



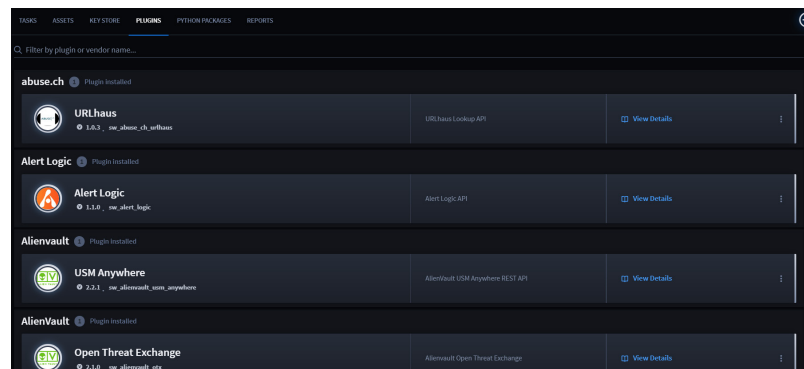
2. From the Select Task drop-down, select the task that will receive the current task's output as input, then click **Save**.



Upload and Manage Plugins

Plugins enable you to perform out-of-the-box integrations with third-party vendors.

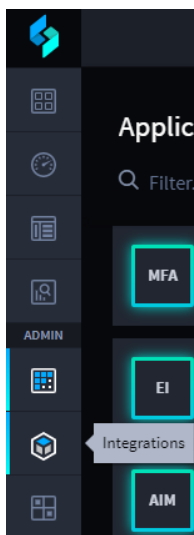
To view installed plugins, from the Integrations page, click the **Plugins** tab.



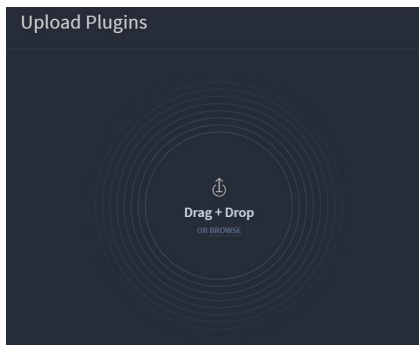
Upload Plugins

To upload a new plugin:

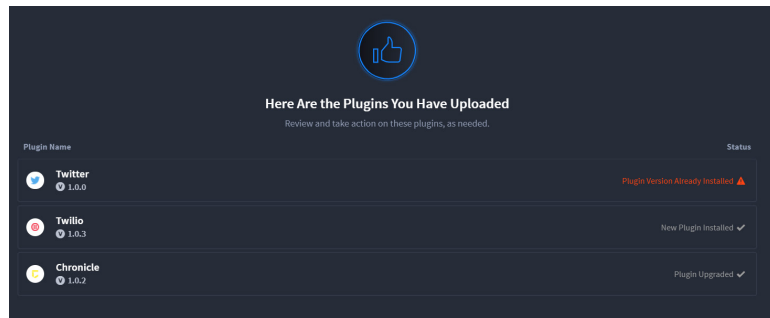
1. From the global navigation menu in Swimlane, select **Integrations**.



2. From the Integrations taskbar, click the **Plugins** tab. Click , or Upload Plugin.
3. In the Upload Plugins dialog, drag and drop the plugin files or click **Browse** to locate and select the plugin files.



On the Upload Plugins dialog, a list shows the uploaded plugins and the install status.

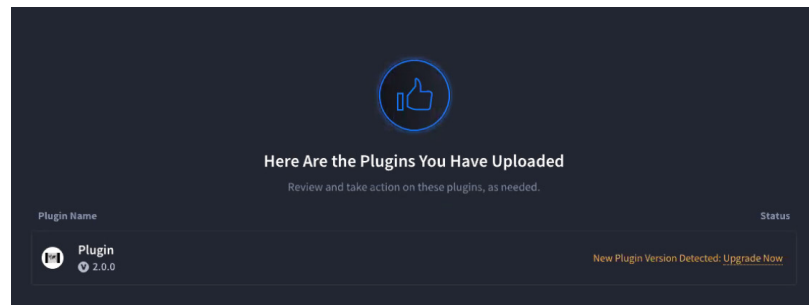


Now your plugins are uploaded to Swimlane.

When you upload a plugin, you introduce the library or utilities for interacting with the third-party software into Swimlane. For most plugins, your next step is to create an asset with credentials from the plugin library.

Upgrade Plugins

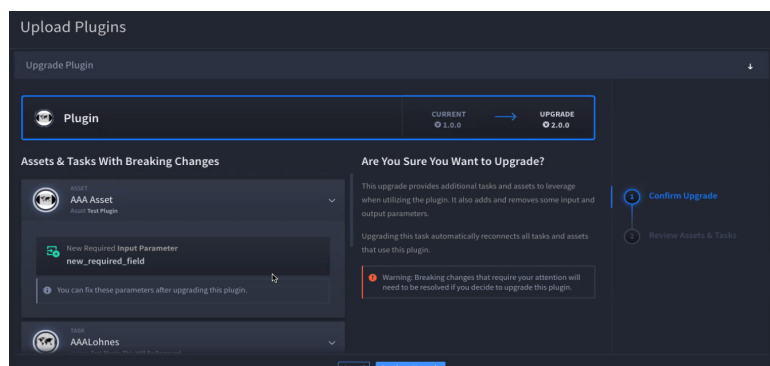
When uploading a newer version of a plugin, if a breaking issue is detected the status on the Upload Plugin dialog will indicate that a new plugin version has been detected. This allows you to upgrade the plugin to the new version without losing previously defined tasks, assets, and configurations. If no breaking change is created by the upgrade, the plugin will automatically upgrade without any additional input.



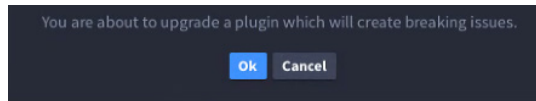
To upgrade a plugin:

1. Click **Upgrade Now** to the right of the plugin.

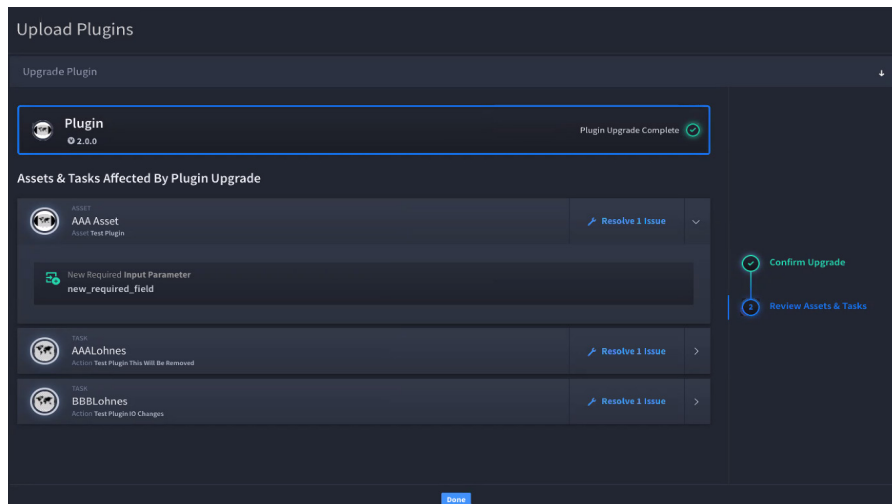
The Upload Plugins dialog will open to show the tasks and assets that upgrading the plugin will affect.



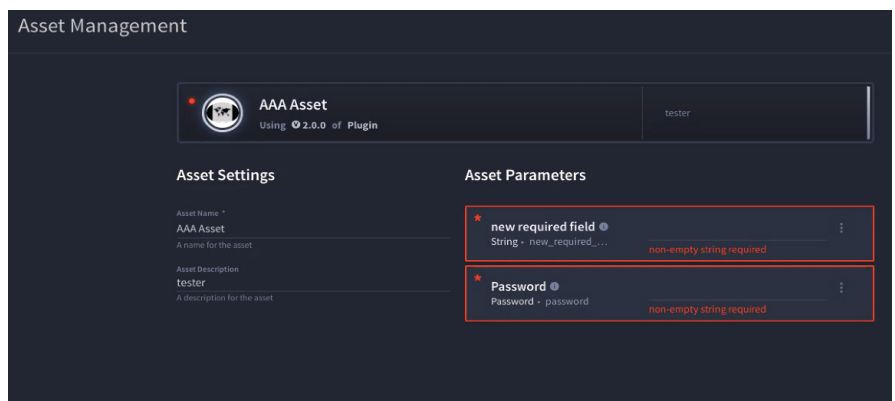
2. Click **Continue Upgrade**.
3. A dialog opens stating “You are about to upgrade a plugin which will create breaking issues.” Select **Ok**.



The Upgrade Plugin window opens showing the tasks and assets affected by the plugin upgrade. This allows you to view and resolve issues.



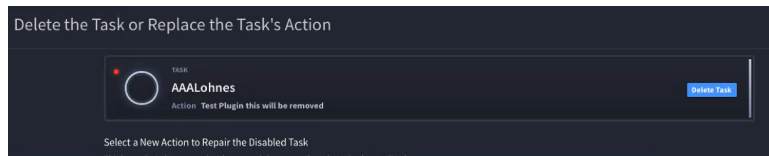
4. Click **Resolve # Issues** to the right of the item.
5. To resolve issues associated with an asset, add the required details to satisfy the asset parameters. Click **Save** to finish resolving the issue.



 **Note:** Required details are indicated by red notation.



- There are two ways to resolve issues associated with a task. If the task has breaking changes, clicking **Resolve # Issues** allows you to fix and save the task. If a task based on a removed action, clicking **Resolve Issues** will allow you to delete the task or select a new action.



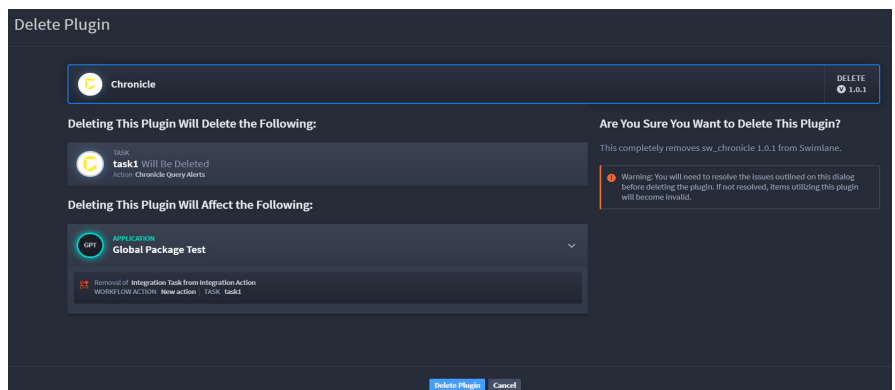
- Once a new action is selected, the task edit window opens, allowing you to review the details. Click **Save and Close** to continue.
- Select **Done** to finish resolving item issues.

Delete Plugins

To delete a plugin:

- On the plugin, click the ellipsis menu and select **Delete Plugin**.

The Delete Plugin dialog opens to show the tasks and assets deleted and the applications, applets, and forked tasks affected by removing the plugin.



- Click **Delete Plugin**.
- A dialog opens stating “You are about to delete a plugin. This will create breaking issues.” Select **Delete Plugin**.

Note: Once the plugin is deleted, the breaking issues noted by this dialog need to be fixed.

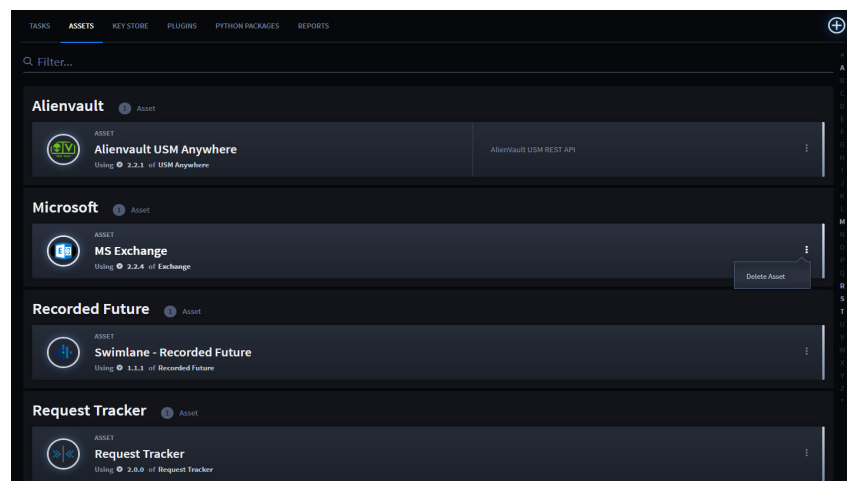
Now the plugin has been removed.

Create or Edit an Asset

Assets are reusable, structured, and product-specific objects that contribute to the success of Swimlane tasks by handling secure authentication and configuration specifications for external systems.



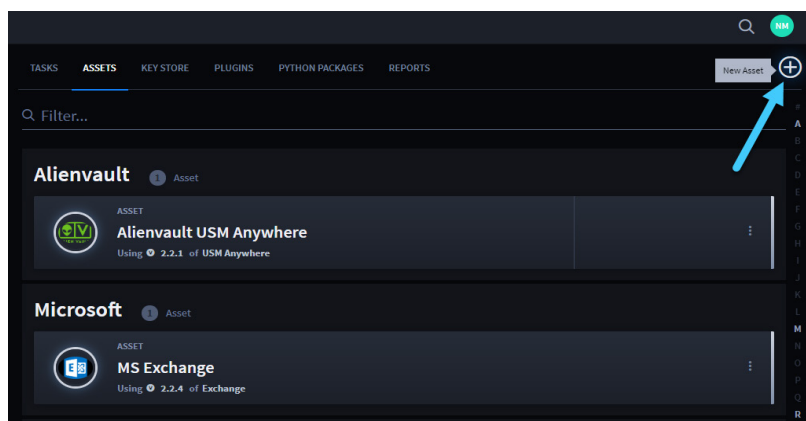
On the Integrations, **Assets** tab, you can view the assets that you have already set up. If you can't find a specific asset, consider looking through your uploaded plugins. You might need to upload a plugin that has the asset type that you are looking for.



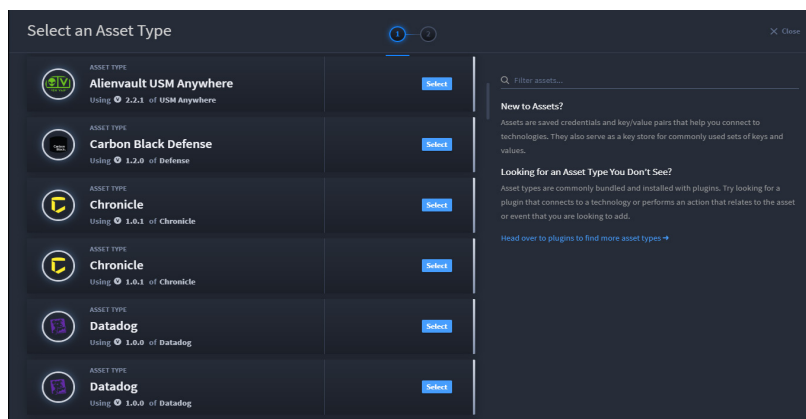
Create a New Asset

To create a new asset:

1. On the Integrations, Assets tab, click **New Asset**.



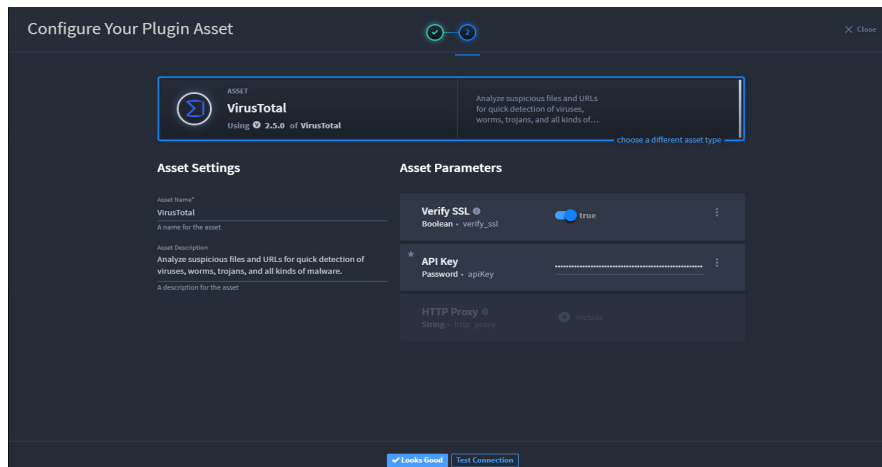
2. On Select an Asset Type, review the available asset types. Once you find the plugin asset you need, click its **Select** button.



You can filter the assets to narrow your search.



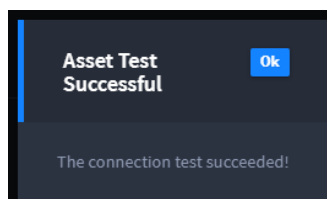
3. On **Configure Your Plugin Asset**, set the asset's parameters. You can also edit the existing **Asset Name** and **Asset Description**, as needed.



The screenshot shows the 'Configure Your Plugin Asset' window for the 'VirusTotal' asset. The asset is identified as 'VirusTotal' using version '2.5.0' of the 'VirusTotal' plugin. The description is 'Analyze suspicious files and URLs for quick detection of viruses, worms, trojans, and all kinds of malware.' The 'Asset Settings' section includes fields for 'Asset Name' (VirusTotal) and 'Asset Description'. The 'Asset Parameters' section includes a 'Verify SSL' toggle set to 'true', an 'API Key' field with a password mask, and an 'HTTP Proxy' field with a dropdown menu set to 'Include'. At the bottom, there are 'Looks Good' and 'Test Connection' buttons.

If the parameters you set on the asset include any API key or other connective details, you can test the active connection of the asset within Swimlane.

4. Click **Test Connection** and ensure you get the resulting "Asset Test Successful" message.



5. On **Configure Your Plugin Asset**, click **Looks Good**.

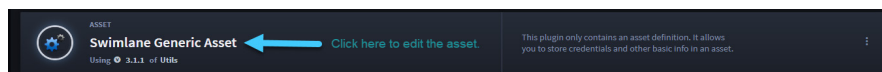
Once an asset is configured, it can be used with any new or existing task.

Note: Tasks can require a certain version of an asset. For example, a task that talks to Elasticsearch 2 would require an Elasticsearch 2 asset.

Edit an Asset

To edit an asset:

1. On the Integrations, **Assets** tab, locate the existing asset.
2. Access the Asset Management page by clicking near the name of the asset.



3. Now you can edit the parameters or asset name or description. Test the connection as necessary and then click **Looks Good**.



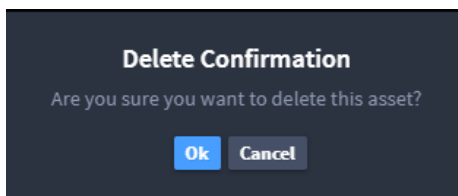
Delete an Asset

To delete an asset:

1. On the Integrations, **Assets** tab, locate the asset.
2. Click the ellipsis menu on the asset that you want to delete, and select **Delete Asset**.



3. On the Delete Confirmation dialog, click **Ok**.



For further help or support with setup, please contact your Swimplane PS Engineer or reach out to our support team at support@swimplane.com.

