Turbine User Guide

1. Quickstart

1.1. Quickstart Overview

Welcome to the Turbine User Guide Quickstart section! This section helps you get started with Turbine quickly and efficiently.

What's in Quickstart?

The Quickstart section provides everything you need to begin using Turbine:

Getting Started

- **Getting Started Basics** - Supported browsers, login methods, navigation, and key terms
- **Set Up Your Profile** - Configure your user profile and preferences
- **Best Way to Start**- Recommended learning path and best practices

Turbine Cloud

- **Turbine Cloud** - Cloud-specific features, security, and tenant management
- **Security-Specific Features** - Database security and account security guidance

Try a Solution

- **AI SOC Solution** - A complete, ready-to-use security operations workflow that demonstrates Turbine's capabilities

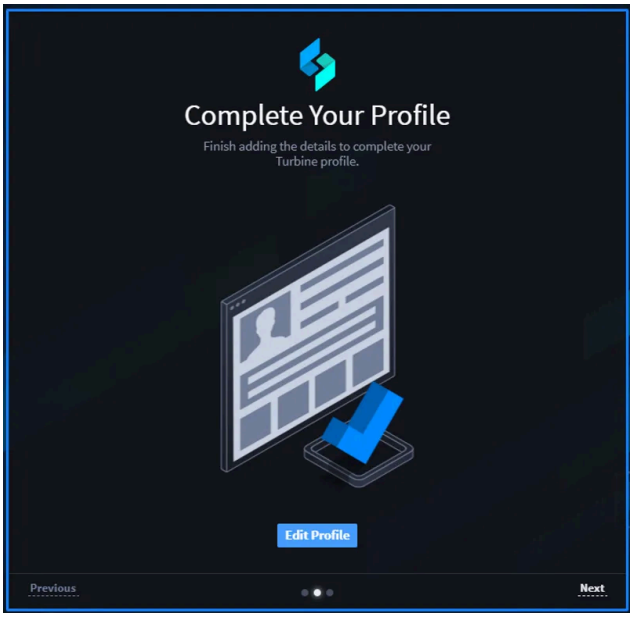
What's Next?

1.1.4.1. Customize Your User Profile

Swimlane Turbine provides flexibility in managing your user profile, allowing you to customize personal settings and manage access tokens, roles, and groups.

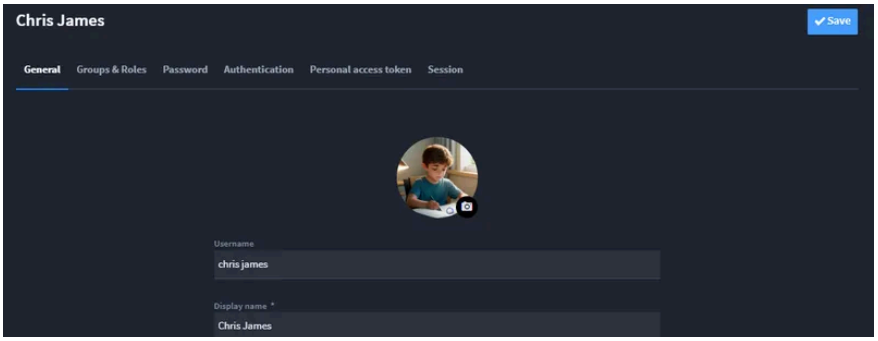
Completing Your Profile

Upon your first login, you'll see the **Complete Your Profile** screen. Follow these steps to finalize your profile:



This action opens the **User Profile Editor**, allowing you to:

- Upload a profile picture to personalize your account.
- View the account's most recent activity.
- Update general details such as your display name, email, and time zone.
- Assign groups & roles to control permissions (Admin only).
- Enable or disable user accounts (Admin only).



2.1.1. Daily Operations Overview

Daily Operations covers all the tasks you perform regularly in Turbine to view, manage, and work with your data.

What's in Daily Operations?

Workspaces

Organize your workspace and navigate between different views and dashboards.

Key Topics:

- Navigate Workspaces and Dashboards
- Workspace management

Dashboards

Create and manage dashboards with charts, visualizations, and interactive elements.

Key Topics:

- Creating and managing dashboards
- Dashboard features (charts, colors, sorting, date ranges)
- Dashboard permissions and sharing
- Dashboard filtering options

Application Records

Work with records in your applications - search, filter, edit, and manage data.

Key Topics:

- **Working with Records** - Search, filter, color code, lock, restrict records
- **Bulk Operations** - Bulk modify, bulk restrict and lock

2.1.1.3.5. Deleting or Editing a User Dashboard

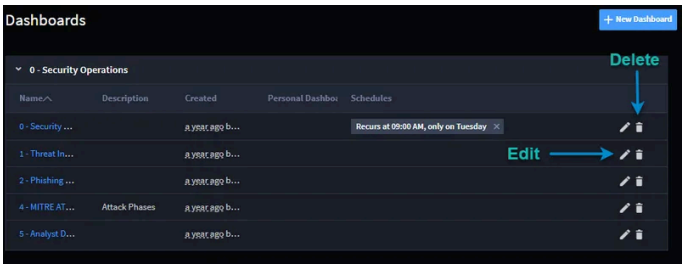
To delete a dashboard:

From the Dashboards taskbar menu, select **Delete**.

You can change the settings of the dashboard by clicking on **Settings and Schedules**. The Dashboard Settings and Schedules pages allows you to edit the following:

- **General Settings**: Edit the **Name**, **Description**, and **Workspaces**.
- **Advanced Settings:** Change the timeline filter duration.
- **Timeline Filters:** Select date fields to apply global date range filters across applications.
- **Schedules:** Create or edit the schedules.
- **Permissions:** Edit the permissions.

You can also edit or delete a dashboard from the Dashboards page. Click the pencil (edit) or the trash can (delete) icon.



2.1.1.3.6. Set Dashboard Permissions

To modify the dashboard permissions, from the Create or Edit Dashboard dialog, click the PERMISSIONS tab. Dashboards can be accessible personally or through role-based access control.

You can also set up private dashboards. Private dashboards allow end-users to create personal dashboards that only they can view and manage.

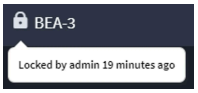
If you have permission, you can also set up private dashboards. A private dashboard can only be viewed or modified by whoever created the dashboard. Administrators or the creator of

2.1.1.4.2.1. Record Lock

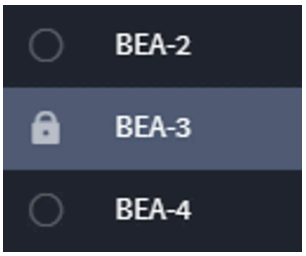
You can lock records while editing and modifying them in order to prevent your changes being overwritten by another user. In addition, locked records can not be bulk modified or bulk deleted.

Once a record is locked, the lock icon appears next to the record's Tracking ID in the Record header.

When a record is locked, only the user who created the lock or an administrator can unlock it. To unlock a record, from the Record header, access the record menu and select **Unlock Record.**

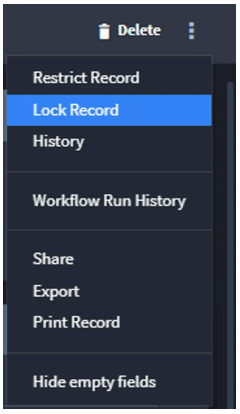


Locked records are also indicated in the report-view-of-records (Default Report) interface.



To lock a record:

1. Open a record. From the record taskbar, access the record menu.



2. From the menu, select **Lock Record**.
   The record is locked immediately. Anyone can access the individual record and open it, but only the user who created the lock or an administrator can make changes to the

2.1.1.4.5. Advanced

2.1.1.4.5.1. Lookup and Create References within Records

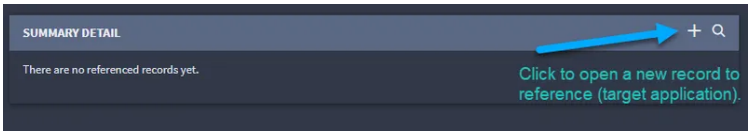
You can select which field(s) to reference from within a record. A reference field on a record, whether single-select, multi-select, or grid, is accompanied by a New and Lookup button adjacent to the reference field.

For more information about reference fields in general, see **Reference**.

Creating a New Record to Reference

To create a new record to reference:

1. From within an open record, locate the reference field, and then click **+**, or Add New. A record editor for the target application opens.



2. Fill out the target record data.

Looking Up References within Records

To lookup and create references within records:

1. Click within the field, or click the Lookup (Search) icon. Enter a keyword or search term and then press Enter to begin the search.

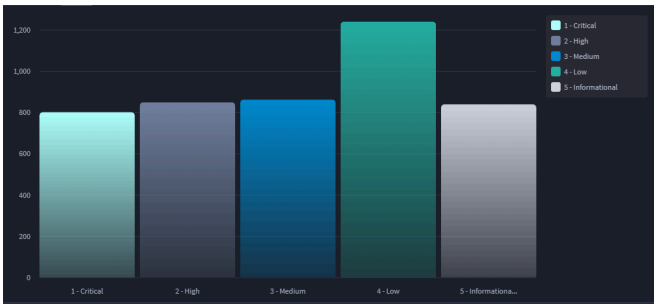
2.1.1.5.4.2. Chart Types

This topic contains chart type examples.

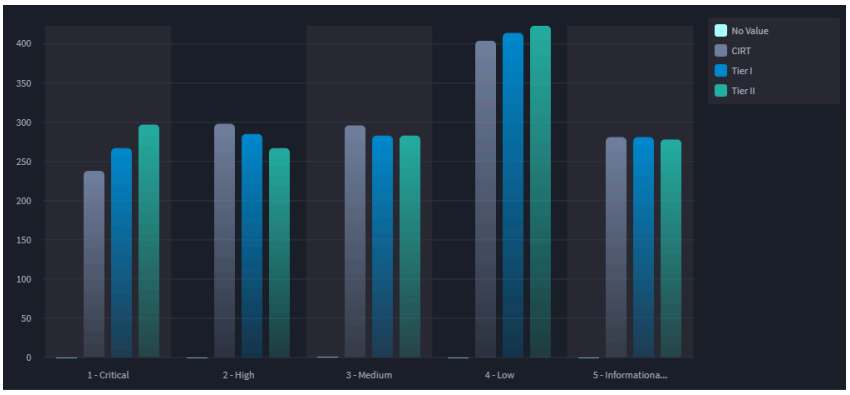
Bar Charts

Use bar charts to show comparisons between different categories of data. The bars can display either vertically or horizontally.

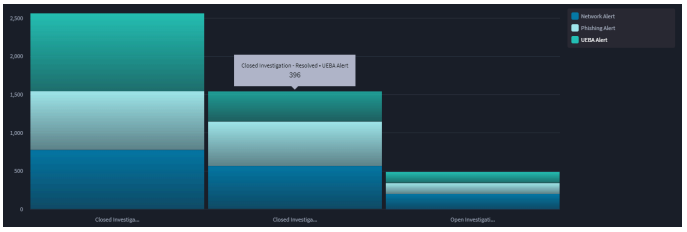
Vertical Bar



Grouped Vertical Bar



Stacked Vertical Bar



100% Vertical Bar

2.1.2.1.1. Getting Started

2.1.2.1.1.1. Playbooks

Playbooks are a series of triggers, logic, and actions that automate a workflow. A playbook can contain triggers, actions, native actions, components, assets, inputs, and outputs.

Playbook Architecture

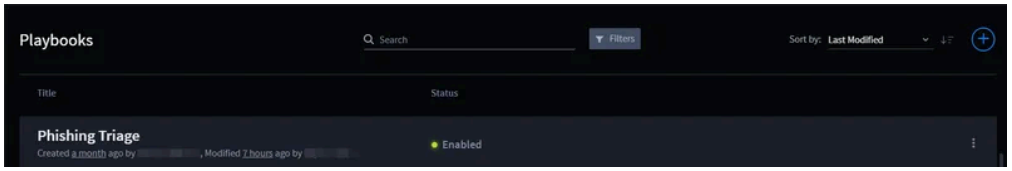
- A playbook can have one or more flows.
- Each flow has exactly one trigger.
- Flows do not communicate with each other.

For details, see **Flows**.

Playbooks Home Page

From **ORCHESTRATION > Playbooks**, you can:

- Search for playbooks by keyword
- Filter by source, interface, or creator
- Sort by last modified, last created, or alphabetical order
- Enable/disable playbooks
- Open playbook operations (export, duplicate, delete)



2.1.2.1.3.1.4. Schedule Triggers

Schedule triggers initiate a playbook at a configured time using cron expressions. They are ideal for automating repetitive tasks, improving productivity, and reducing human errors. For example, you can use a schedule trigger to automatically generate reports, eliminate the need to manually retrieve data, or perform periodic maintenance tasks.

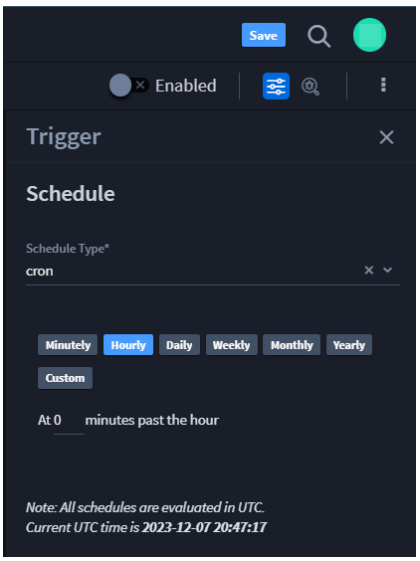
Key Benefits

- **Automated Execution:** Run playbooks automatically at specified intervals without manual intervention
- **Flexible Scheduling:** Support for various schedule frequencies (minutely, hourly, daily, weekly, monthly, yearly, or custom cron expressions)
- **UTC Timezone:** All schedules are evaluated in UTC for consistency across different server locations
- **Reliable:** Uses Hangfire job scheduler for reliable execution of scheduled tasks

Creating a Schedule Trigger

To set a cron job for your playbook, follow these steps:

1. In a playbook, from the Add panel, click and drag **Schedule** to the canvas.
2. Hover over the plus icon to add it to the canvas. The Trigger panel will display on the right side of the canvas, where you can configure your schedule trigger.



2.1.2.1.4.5. Record Actions

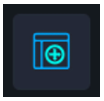
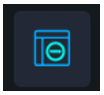
The Create Record, Update/Create Record, Search Records, and Delete Record native actions describe the functions for creating and maintaining data used in Turbine application records. These actions together are also commonly referred to as CRUD.

CRUD Actions in Playbooks

To access these native actions, follow these steps:

1. From the playbook builder, click **Add an action**.
2. From the ACTION panel, in the Action drop-down menu, select one of the desired CRUD options.

The following table shows the action and associated icon.

Icon	Record Action
	Create Record
	Update/Create Record aka Upsert
	Search Record
	Delete Record

The Create and Update/Create Record actions have the ability to configure inputs, outputs, and restrictions to your record.

> Restrictions can be modified on the record as well.

Restrictions and Outputs

Create and Update/Create Record actions have a Restrictions tab where you can restrict application records to specific groups and/or users.

If you navigate to the Restrictions tab on a Create action, and you have not configured any

2.1.2.1.4.10. Using the Loop Native Action

The **Loop** native action in Swimlane Turbine enables orchestrators to apply one or more actions to each item in an array or object. This makes processing collections more efficient by iterating over each element and applying the necessary actions. Loops provide orchestrators with flexibility while maintaining clarity and control.

Overview

The Loop action provides a streamlined way to process arrays and objects by iterating through each element. This allows users to apply actions systematically, ensuring consistency and efficiency in handling repetitive tasks within workflows. Loops can iterate over arrays, objects, or strings, and provide access to iteration context (index, key, value) within the loop body.

Key Benefits of the Loop Action

- **Native Action:** Easily accessible within the playbook builder.
- **Efficient Array Processing:** Simplifies operations on arrays by automating repetitive tasks.
- **Flexible Integration:** Works seamlessly with other actions and connectors in the playbook.
- **Supports Nested Arrays:** Capable of handling complex data structures through nested loops.
- **Parallel and Sequential Processing:** Offers asynchronous and synchronous processing options for *forEach* loops to suit different workflow requirements.
- **Conditional Iteration:** Allows for conditional processing using *while* loops.

Limitations

- **Nested loop depth:** You can nest up to five loops inside a single playbook (one loop plus four nested levels).

Using the Loop Action

2.1.2.1.7. Action Configuration

2.1.2.1.7.1. Inputs

Actions are individual capabilities of connectors that interact with external technologies by passing in data via action inputs. Results are available from action outputs.

Actions can:

- Call another playbook
- Interact with external technologies
- Contain inputs and outputs (outputs can be promoted through the **Playbook Outputs** dialog)

> You do **not** need a trigger to add an action to your playbook.

Create Actions

1. From your current playbook, open the **Add** panel on the left side of the screen. The **Add** panel displays available actions, connectors, and components organized by category. The panel has three tabs: **Triggers**, **Actions**, and **Components**.
2. Browse or search for the desired action from the available connectors and native actions.
    - Native actions (such as Create Variables, Update Variables, HTTP Request, Script, Transform Data, Condition, Loop, Parallel, Create Record, Delete Record, Search Records) appear at the bottom of the **Add** panel
    - Connector actions are organized by vendor/connector above the native actions

2.1.2.2.2.2. Publish components

Publish Components

To publish a component to the **User Content** library:

Before You Publish

- Confirm the component name and description are accurate.
- Verify inputs/outputs are configured as intended.
- Test the component in a playbook to validate behavior.

Publish as New Content (V1)

Steps:

1. You can publish a component by selecting **Publish** from the ellipsis icon of **Components** screen or navigate to the desired playbook.
2. Open the component settings menu and select **Publish**
3. In the **Publish Content** window:
    - Select **Publish as New Content (V1)**.
    - Provide a name and description for the content.
    - Click **Next**.
4. Select Content to Publish:
    - All the playbook components are listed.
    - Click **Next**.
5. Review Content:
    - Review the selected content and check for potential issues.
    - If everything looks good, click **Next**.
6. Summary:
    - Confirm the details: version (1.0.0), content name, description, and publish message.
    - Click **Publish**.

2.1.2.4. Classic

2.1.2.4.1. Classic Playbooks

Playbooks are where automation is built quickly and easily and enriches data processing. With Swimlane Turbine's playbooks, anyone can create modular, repeatable automations that process real-time data.

What are Playbooks?

Playbooks are self-contained modular entities that help you quickly and easily build and/or enrich automation processes.

Playbooks contain:

- Triggers
- Actions
- Conditions
- Inputs and Outputs
- Repeats
- forEach Loops

Playbooks are initiated by triggers and used to achieve desired outcomes.

Playbook Inputs

Playbook inputs pass and normalize data into a playbook from a trigger, application, or event.

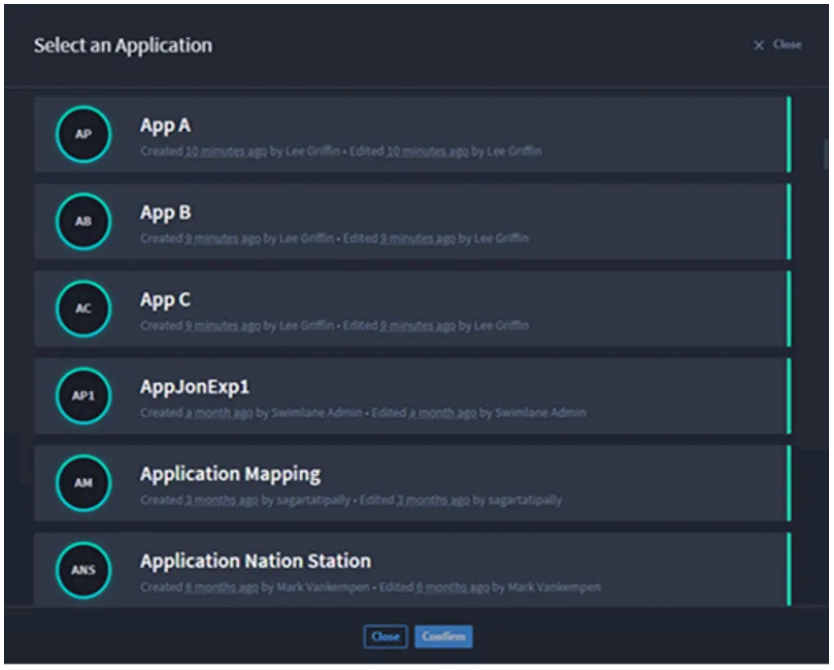
2.1.2.4.1.4.2. Map Playbook Outputs to Applications

When you map playbook outputs to an application, you allow the playbook runs to update records for various things, such as performing manual security actions, collecting information for charts and dashboards, and increasing visibility for auditing.

Map Playbook Outputs to Applications

Create a new playbook or upload an existing playbook, then follow the steps below:

1. Click **Playbook Outputs**.
2. On **Playbook Outputs**, click the **Application Mapping** tab.
3. To map the promoted playbook outputs to an application, click **Select Existing Application**.
4. On **Select an Application**, click the application to which you want to map the playbook outputs and click **Confirm**.



Update or Create New Records

Select how you want to handle distribution of the playbook outputs to the application's records.

1. Click **Update or create new records**.

2.1.2.4.1.8.6. Repeats

Playbooks allow you to repeat actions. Arrays are the only repeatable property types.

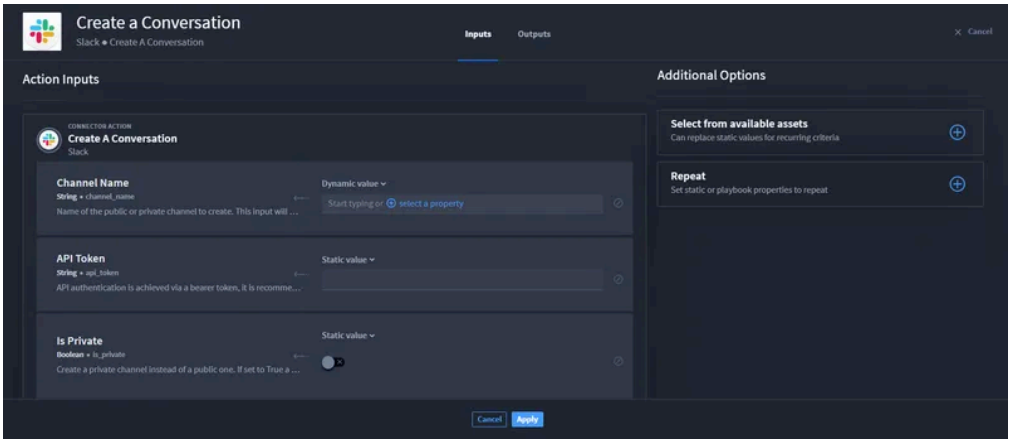
Add Repeats to an Action

Repeat actions can be static or dynamic.

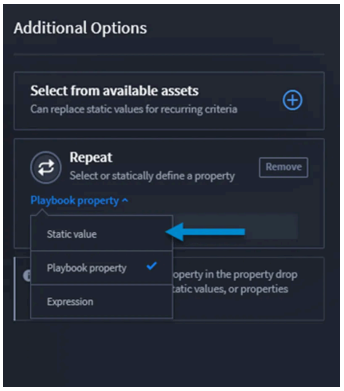
> Dynamic actions are reserved for string inputs only.

To add a repeat to an action:

1. Add an action.
2. From **ACTION**, click **Configure**.
   From here, decide which fields you need to complete.
3. If you want to repeat a static value, click the **Repeat** plus icon.



4. Click the drop-down menu and select *Static value*.



2. Click **Configure Static Values**.

2.1.2.4.1.9.4.3. Linking Records

This use case shows how to link records successfully when the **No record linking applied** button is enabled.

Scenario

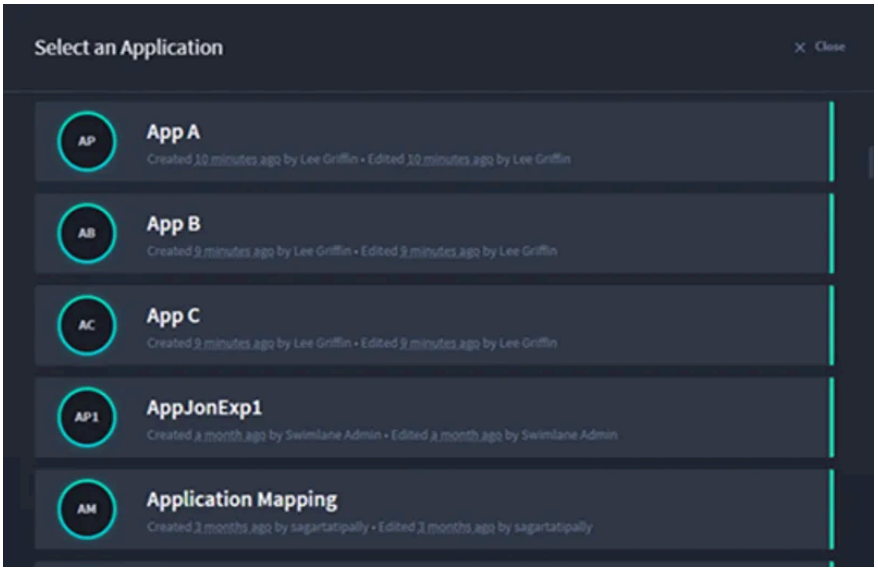
Max is in the Linking Records playbook. Max wants to map an output record to the input record that triggers the playbook.

Max creates a playbook and names it **Linking Records**.

> This playbook name is for the sake of the use case example. You can name the playbook according to whatever outcome/action you are trying to create.

1. In the Linking Records playbook, click **Playbook Outputs**.
2. On Playbook Outputs, click the **Application Mapping** tab.
   If Max has not promoted any playbook outputs, he sees the **There are no mappings yet** message.
3. To map the promoted playbook outputs to an application, click **Select Existing Application**.
4. On Select an Application, click the application for which you want to map the playbook outputs, and then click **Confirm**.

Max selects App A.



2.1.2.4.1.10.4. Schedule Triggers

A schedule trigger initiates a playbook at the configured schedule.

> You do not need to select a trigger before adding and configuring actions and conditions.

Schedule Triggers

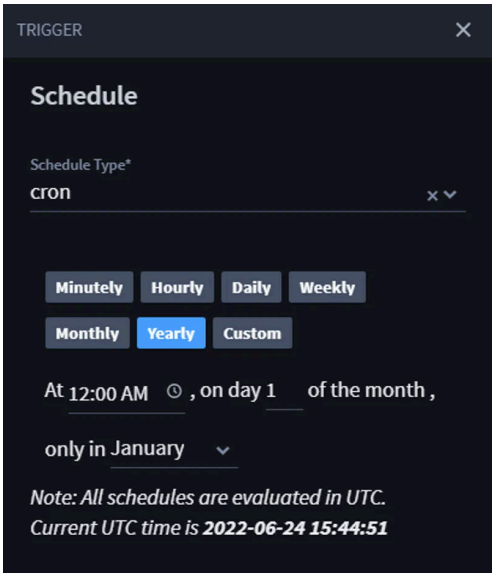
Create a new playbook or upload an existing playbook, then follow the steps below:

1. On the playbook, click **Add a trigger**.
2. Click **Schedule**.

Example

The example below shows a yearly cron job that runs at 12:00 AM on January 1st each year.

1. Once the trigger options display, select **Schedule**.
   The **TRIGGER** window displays with the **Trigger Type** as *Schedule* and the **Schedule Type** as *cron*.
2. Select the desired schedule option: **Minutely**, **Hourly**, **Daily**, **Weekly**, **Monthly**, **Yearly**, or **Custom**.
3. Depending on the selected schedule, configure the schedule in the fields that display.



2.1.3. Applications and Applets

2.1.3.1. Getting Started

2.1.3.1.1. Applications and Applets

Applications and Applets are the foundation of the Swimlane Turbine platform.

What is an application?

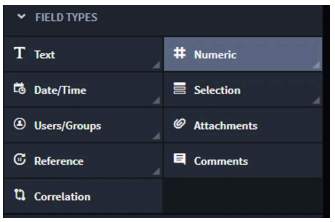
A user-defined template for collecting, storing, and organizing your data. All automated activities and decisions are driven by how your application stores data. You also manage workflow from within applications.

What is an applet?

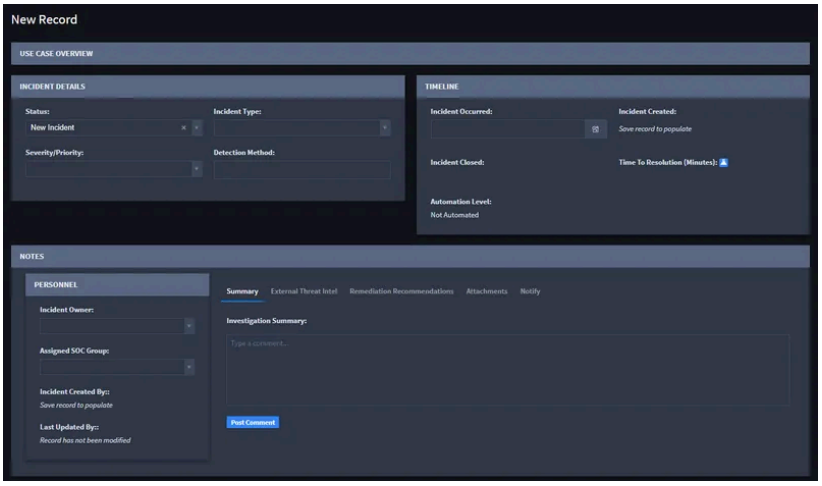
A preconfigured set of fields and layout specifications. Applets are appended to an existing application form layout and allow users to easily update and expand their existing applications.

2.1.3.2.6. Select Fields and Assign Field Properties

You select fields and assign field properties as you build your application. This section covers the Application and Applet fields and the options they provide for setup on the FIELD PROPERTIES tab.



The fields you use to build your application are the fields that users will access when they create or edit records in Swimlane. Here's a screenshot of a new Swimlane record with multiple fields available to fill out.



Field Keys

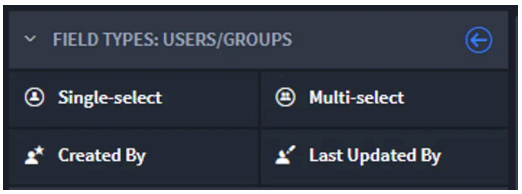
Field Keys function as aliases for fields. They allow for consistent data ingestion, product upgrades, and imports of applications across different environments.

Field Keys are validated for uniqueness against other Field Keys and field Display Names within an application (they can be the same as their corresponding Display Name). They can be a maximum of 32 characters and can only contain:

- Lower-case letters.
- Numbers.

2.1.3.2.6.10. Users / Groups

Use this field when you need to select from a list of users or groups. You can select individual users or groups, or select multiple users or groups.



When creating a Users/Groups field, you can choose from these types:

Field Type	Description	Example
Single-select	Use to select one user or group.	User: D. Groundloop
Multi-select	Use to select one or more users and/or groups.	Groups: SOC Analysts, Desktop Support
Created By	Use to have Swimlane Turbine specify the user that initiated the record.	Automatically populates with user's name.
Last Updated By	Use to have Turbine specify the user who last edited the record.	Automatically populates with user's name.

To create Users/Groups fields:

From Application Builder's Field Types, select a users/groups field and then drag and drop it to the Form Layout. Drop the field in the layout area, or within a Tab or Section layout object.

Access the field's properties and complete the following fields as needed:

Field	Step	Example
Display Name	Enter the name of the field.	*Assigned User*
Help Text	Enter contextual help text. You will first need to specify whether the help text will appear above or below the field	*Specify the users and/or groups*

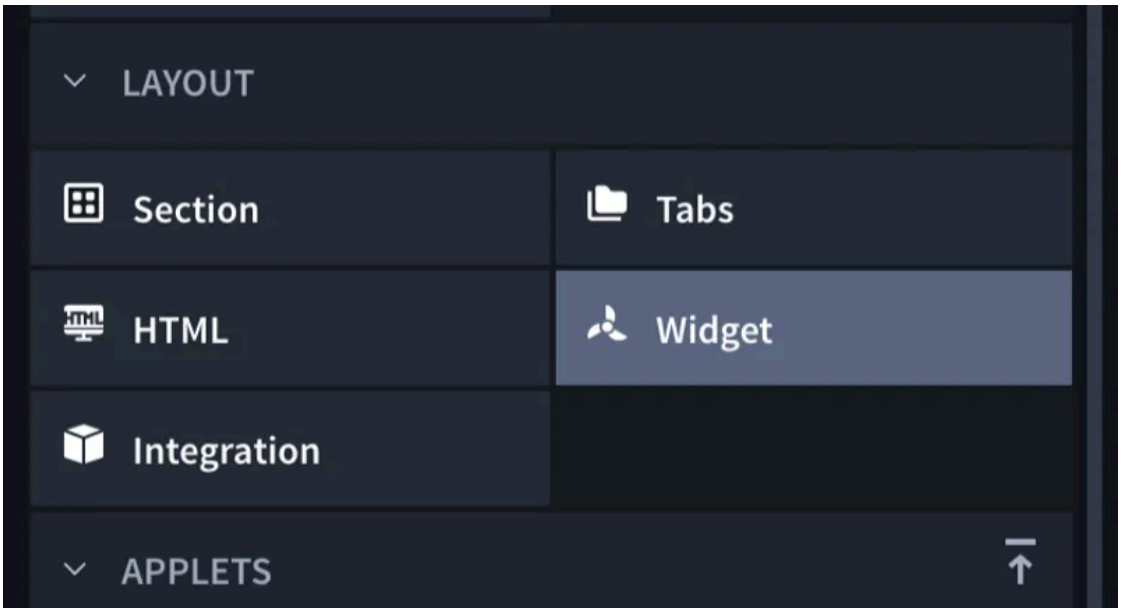
2.1.3.4.2. Record Widgets

Widgets are javascript and html components that you can use to enrich the User Interface (UI) of records. They allow you to create your own UI components that interact with Swimlane or third-party endpoints. Currently, Swimlane supports two types of widgets: Record widgets and Report widgets.

This topic covers Record widgets. For more about using widgets for reports, see **Report Widgets**

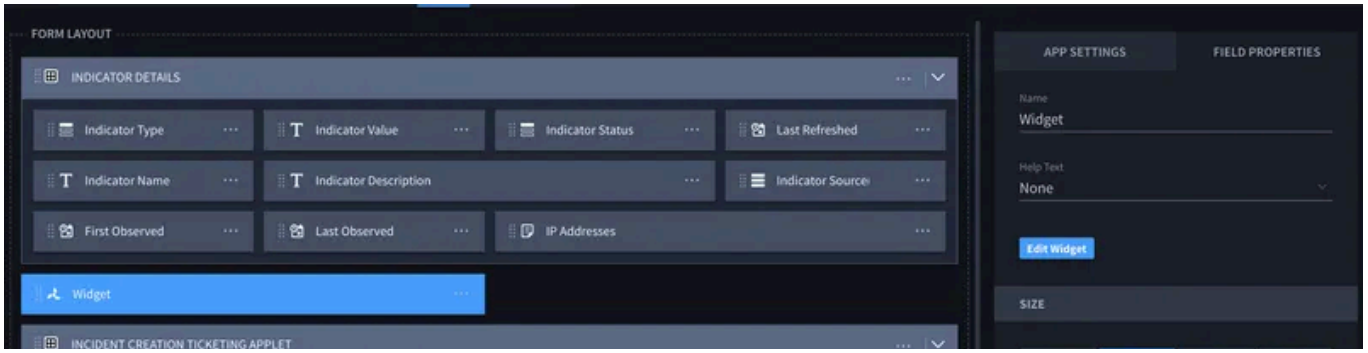
To create record widgets:

From the Layout section of the Application Builder or Applet Builder page, select **Widgets**, and then drag and drop it into the Form Layout.



You can add multiple widgets per application or applet.

To edit the record widget, select the widget in the Form Layout, and then, from the Field Properties click **Edit Widget**.



2.1.3.5.5. Workflow Actions

Configuring Actions

Using the Form Layout Dialog (Select Fields)

Several workflow actions use the **Form Layout Dialog** to select fields and layout items. This dialog provides a visual representation of your application's form structure.

Actions That Use Form Layout Dialog:

- **Set Field Read/Write** - Uses "Select Fields" button
- **Set Field Required/Optional** - Uses "Select Fields" button
- **Modify Layout (Show/Hide)** - Uses "Select Layout Items" button

How to Access:

1. Configure a workflow action (Read/Write, Required/Optional, or Show/Hide)
2. Click the **Select Fields** or **Select Layout Items** button
3. The Form Layout dialog opens in a modal window

Dialog Structure:

The Form Layout dialog displays:

- **Dialog Header**:
  - Application/applet icon and acronym
  - Application/applet name
  - **Apply** button (saves selections and closes dialog)
- **Form Layout Section**:
  - Shows all fields and sections as they appear in your application form
  - Maintains hierarchical structure (sections, tabs, nested sections)
  - Fields display with their type icons
  - Sections show their names and structure
- **Hidden Fields Section** (if applicable):

2.1.3.6.1.7. Text

Char

Description: Returns the character whose number code is defined in the parameter.

Example:

Char(number)

Clean

Description: Removes the non-printable characters from the given text, represented by number 0 to 31 of the 7-bit ASCII code.

Example:

Clean(text)

CONCATENATE

Description: Joins arguments into a single string.

Example:

CONCATENATE (text1, text2,...)

Dollar

Description: Converts a number to text, using a currency format.

Example:

Dollar (number, decimal_places)

LEFT

Description: Returns the first character or characters in a text string, based on the number of characters you specify.

Example:

LEFT(text,num_chars)

2.1.5.3. User Management

2.1.5.3.1. Groups

In Swimlane Turbine, Groups serve as collections of users with shared permissions, roles, and access levels, allowing administrators to manage access and permissions efficiently across multiple users. Each user can belong to several groups simultaneously, which is particularly useful for organizing users by department, role, or specific project needs. By managing permissions at the group level, organizations enhance security, streamline onboarding and offboarding, and ensure consistent access control across departments, projects, and workflows, ultimately saving time and improving collaboration.

How Groups Work

Each user can belong to single or multiple groups simultaneously, making it easy to organize users based on department, role, or specific project needs. This flexibility supports dynamic roles and cross-functional team structures.

Key Benefits:

- **Enhanced Security:** Permissions are managed at the group level, ensuring users have access only to relevant resources and workflows.
- **Streamlined Onboarding and Offboarding:** Adding or removing users from groups automatically applies the appropriate permissions, simplifying user lifecycle management.
- **Consistency and Efficiency:** Group-based management ensures consistent access control across departments, projects, and workflows while reducing administrative overhead.

2.1.5.3.6.4. Ping Identity SCIM Integration

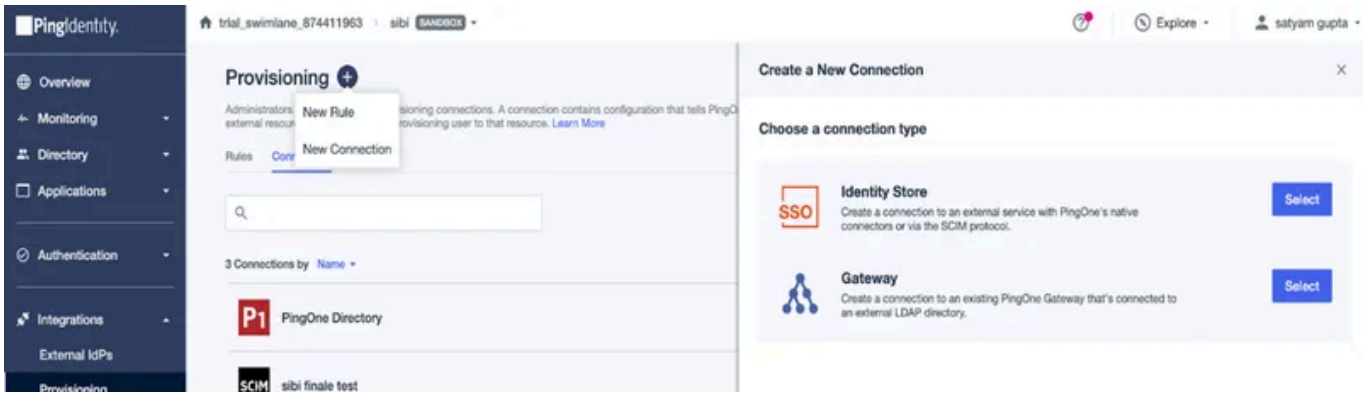
Swimlane Turbine supports SCIM 2.0 integration with **Ping Identity**. This integration enables administrators to automatically provision and de-provision users and groups from Ping to Swimlane Turbine using the SCIM standard.

SCIM helps customers manage onboarding and offboarding of users centrally in Ping without logging in to Swimlane Turbine for manual user management.

How to Configure Ping Identity SCIM

Use this section to configure a SCIM outbound provisioning connection and provisioning rules in Ping Identity.

1. Sign in to the **Ping Identity (PingOne) administrative console**.
2. In the left navigation menu, under **Integrations**, click **Provisioning**.
3. On the **Provisioning** page, click the **plus (+)** icon.
4. Click **New Connection**.
5. Under **Choose a connection type**, click **Select** next to **Identity Store**.
6. From the available options, select **Provisioning Identity Store - SCIM Outbound**.
7. Click **Next**.
8. Customize the connection details:
    - Name (required)
    - Description
    - Icon (optional)
9. Click **Next**.



2.1.5.5. Security and Authentication

2.1.5.5.1. Enable SAML for SSO

SAML (Security Assertion Markup Language) is an open standard that facilitates single sign-on (SSO) by allowing Swimlane Turbine (the service provider) to rely on an external identity provider (IdP) to authenticate users.

Using SAML, there are two possible ways to initiate login:

- **Service Provider-initiated:** The login process starts with Swimlane Turbine.
- **Identity Provider-initiated:** The login process starts with the external identity provider (for example, Okta, JumpCloud, and Azure Entra).

Benefits of SAML Authentication

SAML authentication provides numerous advantages for organizations, improving security, streamlining user access, and simplifying administration. Below are the key benefits:

- **Centralized User Management**: User authentication is managed through a single, external IdP, reducing administrative overhead.
- **Enhanced Security:** Leverages the security features of the IdP, such as Multi-Factor Authentication (MFA) and conditional access policies.
- **Seamless Login Experience:** Users can log in to Swimlane Turbine using existing credentials, improving usability.
- **Scalability:** Easily accommodates user management for large organizations or multi-

2.1.5.6.2. Enabling Git Integration in Swimlane Turbine

Swimlane Turbine offers Git integration, allowing you to sync the content in your Turbine Library with a remote GitHub/GitLab repository. This ensures proper version control, collaboration, and easy recovery of your Turbine content.

Why Git Integration?

1. **Change Management**
    - **Audit Trails and Accountability**: Git provides a clear history of all changes, including who made them and when. This is critical for organizations needing to track modifications and understand events leading to the current state of their system. It helps enforce accountability, especially in collaborative environments.
2. **Integrity and Security**
    - **Immutable History and Backup or Recovery**: Once changes are committed to Git, they become part of a permanent history that cannot be altered without evidence, protecting the integrity of your data. Git repositories can also be backed up, ensuring that you can recover data in case of any loss or mishap.
3. **Regulatory Compliance**
    - **Documentation and Evidence**: Git's logs and commit messages serve as evidence during audits and compliance assessments. It provides documentation that is useful for demonstrating an organization's adherence to change management and version control best practices.
4. **Incident Response and Forensics**
    - **Rollback Capabilities**: Git makes it easy to revert changes, allowing for quick restoration in case of harmful or accidental modifications. This minimizes the risk of disruptions from unauthorized or erroneous changes and helps in forensic investigations to understand incidents and recover from them.

Setting up the Git Repository

To setup Git Repository in Swimlane Turbine from the Admin Panel:

2.1.5.8.1.1. Requirements and Pre-Installation

User Permission Requirements

- Must be an Account Admin in Swimlane.
- Multi-Factor Authentication (MFA) must be disabled for installation, or use the "**Exempt Force SSO**" option if MFA is enabled.

The user used for installation is not saved, and may be removed after installation. Instead, machine IDs and secrets are created during installation for agent communication.

Supported Operating Systems

- For the latest supported OS list, see **Critical Prerequisites**.

Hardware Requirements

Resource	Minimum
CPU	2 vCPUs (x64)
RAM	2 GB
Disk	OS minimum + 20 GB

Additional hardware may be required dependent on workload and specific processing requirements.

Network Requirements:

- Outbound HTTPS (443) to your Swimlane Turbine instance and quay.io
- Whitelist all of **quay.io**

Remote agent installation typically downloads a bash script from your Swimlane Turbine instance. During installation, several containers will be downloaded from **quay.io**. They may also be downloaded during cron-based upgrades. Connectors are not downloaded from **quay.io** and are instead pulled from the primary Swimlane Turbine instance.

Required Software

You must install Docker (latest) or Podman 4.2+ (only one required):

2.1.5.8.4.3. Support

If you encounter issues, collect agent container logs and all error messages, and provide these to Swimlane support for fastest resolution.

2.1.5.8.5. Uninstall Remote Agent

To uninstall the remote agent:

- Delete the two agent containers in Docker Desktop / Podman:
  - `docker rm -f <agent_name>-turbine-agent / podman rm -f <agent_name>-turbine-agent`
  - `docker rm -f <agent_name>-podman / podman rm -f <agent_name>-podman`
- After the remote agent is disconnected (may take a couple of minutes after container deletion), delete the remote agent from the **Tenant-specific** settings page, list of tenants.
- `<agent_name>` is the name of the agent user input while creating the agent.

2.1.5.9. Audit and Compliance

2.1.5.9.1. Audit Logging

Audit logging in Swimlane Turbine records critical actions and events across the system, providing visibility for monitoring, security, and compliance. Audit logs can be retrieved using

2.1.6.2.1. Turbine Schema Reference (AI SOC)

> Turbine Schema was formerly known as Turbine Extensible Data Schema, or TEDS.

This document describes the Turbine Schema objects used by the **AI SOC Solution**. For general concepts, best practices, and troubleshooting, see **Working with Turbine Schema**.

For additional Turbine Schema objects used across solutions (Observable, Enrichment, File, File Hash, Header, MIME Part, Detection Rule, Attack, Tactic, Technique, Status, Error, Content, User, and Cloud Storage Query Input), see **Turbine Schema Reference (Classic SOC)**.

For AI SOC interface contracts (Alert to Alert, Alert Search Params, Email Search Params, and Alert Triage Ingestion), see **AI SOC Interfaces**.

Alert Object

The AI SOC Alert object captures security events and incidents from SIEM, XDR, and EDR systems. It includes fields for alert identification, severity and priority, host and user context, MITRE ATT&CK and D3FEND mappings, observables, and supporting evidence.

Name	Key	Type	Requirement	Description
Alert UID	`alert_uid`	String	Required	Unique identifier for the alert
Title	`alert_title`	String	Recommended	Name or title of the alert
Description	`alert_description`	String	Recommended	Brief summary of the alert, detailing the nature and significance of the event
Severity	`alert_severity`	String	Recommended	Alert severity level (for example, High, Medium, Low)

3.2.4. How to Collect MongoDB Information and Logs

This article outlines steps on how to collect the necessary MongoDB information and logs. This collection is needed by Engineering and MongoDB Support in order to better understand the issue and begin their analysis.

The following items are the most often needed and usually help troubleshoot 90% of issues:

- mongod logs files from all members
- dbPath/diagnostic.data folder content. This contains MongoDB internal metrics, which help in diagnosing i/o, cache, journal, and other issues.

We'll use a script to capture the aforementioned items along with additional information to assist with the troubleshooting.

Turbine Platform Installer (TPI)

1. Download the collect-mongo-logs-tpi.sh script available at the bottom of this page.
2. For Standalone, the MongoDB PRIMARY member will always be swimlane-sw-mongo-0. If this is HA, then let's start by looking up the MongoDB PRIMARY member. You'll need this information when running the script.

```
export NS=<SWIMLANE-NAMESPACE>kubectl -n $NS exec mongo-0 -- mongosh -u
Admin -p --authenticationDatabase admin --tls --tlsAllowInvalidHostnames
--tlsAllowInvalidCertificates admin --eval="rs.isMaster().primary;"
```

3. Execute the file

```
chmod u+x collect-mongo-logs-tpi.shcollect-mongo-logs-tpi.sh | tee
collect-mongo-logs-tpi-results.log
```

4. Update the Support ticket with the following information:
    1. Upload the resulting /tmp/mongo.logs.####.tar.gz file to our secure portal
    2. Timeframe information of the incident
    3. Is this issue recurring?
    4. Is there any workload-driven pattern?

3.2.14. What happens when a license expires?

This article explains what happens when the Swimlane application and Swimlane Platform Installer (SPI) license expires.

Swimlane application

When a Swimlane license expires:

- Non Admin users can not login
- No data is deleted (i.e. apps/applets/records/key store/assets, etc).
- No credentials are removed.

NOTE: No data is deleted or changed, but the system will essentially stop working from that point forward. Users will not be able to log in.

3.3. Use Cases

3.3.1. Condition and Logic Use Cases

3.3.2. HTTP Integration Use Cases

3.3.2.1. HTTP Requests

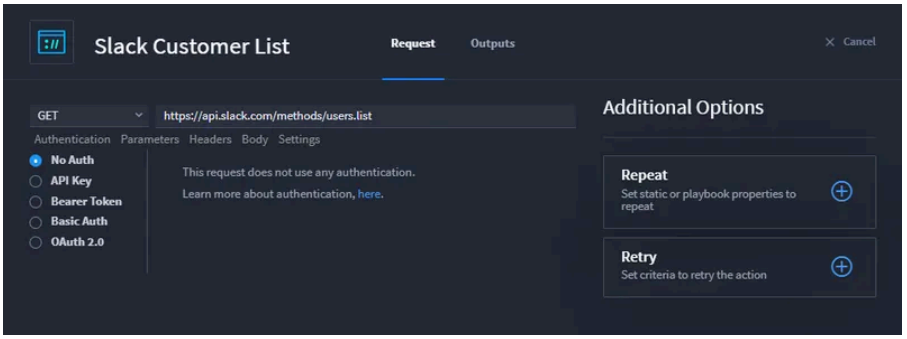
Use HTTP Request native actions to get data from or send data to an API endpoint, even if the connector does not exist for the service. This gives flexibility in integrating Turbine with third-party services.

Scenario: Get Users from Slack

Max, a Turbine Admin, needs to verify a list of Slack users via the Slack API. He uses the **HTTP Request** action to retrieve data from the API using a **GET** request.

Steps to Set Up a GET HTTP Request:

1. Select **No Authorization** under the Authentication tab.
2. Enter the URL: **https://api.slack.com/methods/users.list**.
3. On the Settings tab, ensure that the **Enable SSL certificate verification** toggle is ON.
4. Click **Apply**.



3.3.5. Script Use Cases

3.3.5.1. Script Test Use Case

Scenario: Test Script Action

Rowan is an orchestrator who wants to map static and playbook property data to the Script native action inputs so that he can reference this data in the Python script and test the inputs to see the results before continuing to build the playbook. He also wants to see if there are any discovered outputs in addition to the original outputs.

First, Rowan adds his webhook trigger, so the webhook property data is available downstream in the playbook. Let's see how he sets up the Script native action to see if there are **Discovered Outputs and Testing**.

Steps to Configure the Script Native Action:

1. Click the **Action** drop-down menu, select **Script**, and enter a title for the action.
2. Click **Configure** to open the Script configuration window. This window displays three tabs: **Script**, **Outputs**, and **Test**.
   Note: To ensure proper testing, verify that your script has valid inputs and a valid output structure. This avoids errors in the next steps.
3. From the Script tab, click **Add property** and select **String**. Enter the value "**hello**" for this property.
   Tip: You can add multiple input types (e.g., strings, numbers, objects) to test various scenarios.
4. Add another property of type **Object**. Rename the properties as **string_test** and **object_test1** for easier identification.