Turbine User Guide

Modified: Sep 29, 2026

CONTENTS

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.

1.1.1. What's in Quickstart?

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

1.1.1.1. 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

1.1.1.2. Turbine Cloud

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

1.1.1.3. Try a Solution

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

1.1.2. What's Next?

After completing Quickstart, explore:

- **Daily Operations Overview** — Learn how to work with workspaces, dashboards, records, and reports
- **Orchestration** — Build automation with playbooks, connectors, and components
- **Hero AI** — Companion chat, Text to Playbook, component building, and native AI actions
- **Examples and Troubleshooting** — Learn from use cases and solve problems

---

1.1.1. Best Way to Start

1.1.1.1. Try a Solution (Recommended for New Users)

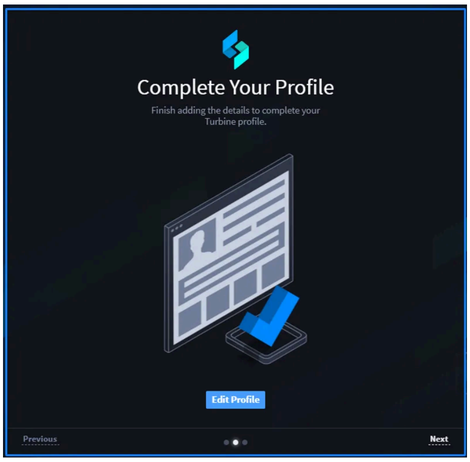
The fastest way to see Turbine in action is to install and try a complete solution. The **AI SOC Solution**

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.

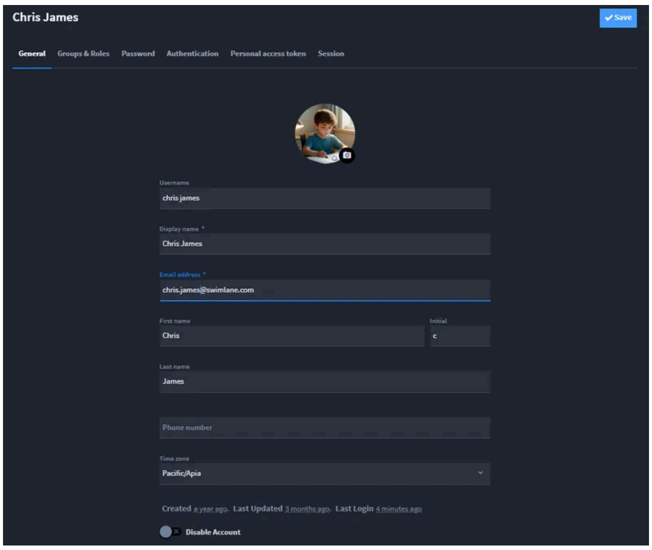
1.1.4.1.1. Completing Your Profile

Upon your first login, you will 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).



1.1.4.1.2. Changing Your Password

2.2. Daily Operations Overview

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

2.2.1. What's in Daily Operations?

2.2.1.1. Workspaces

Organize your workspace and navigate between different views and dashboards.

Key Topics:

- Navigate Workspaces and Dashboards
- Workspace management

2.2.1.2. 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

2.2.1.3. 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
- **Advanced** - Lookup references, manage server conflicts
- Import data, customize views, sharing options

2.2.1.4. Reports

Create, edit, and share reports to analyze and present your data.

Key Topics:

- Create and save reports
- Edit reports

2.2.3.5. Deleting or Editing a User Dashboard

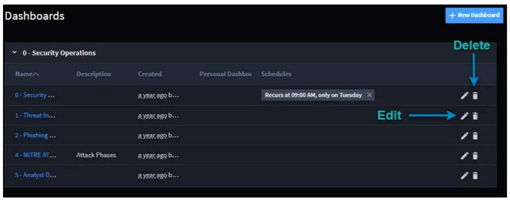
To delete a dashboard:

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.



2.2.3.6. 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 the dashboard can delete a personal dashboard.

2.2.4. Application Records

Records are the data from your Swimlane Turbine applications. They are made up of various fields which are customizable, configurable, and built within Turbine's Application Builder.

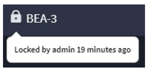
The fields and options available in the record vary based upon how the application is configured. Turbine administrators control tasks related to records for administrators and users, and can grant additional access to users at the application, record, and field level. The listing of records in Turbine are considered Reports.

2.2.4.2.1. Record Lock

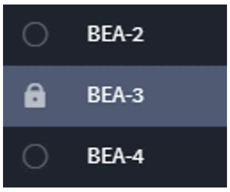
You can lock records while editing and modifying them in order to prevent your changes being overwritten by another user. In addition, locked records can not be bulk modified or bulk deleted.

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.**

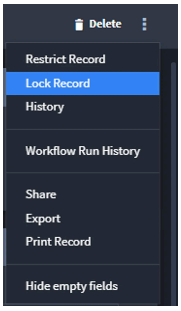


Locked records are also indicated in the report-view-of-records (Default Report) interface.



To lock a record:

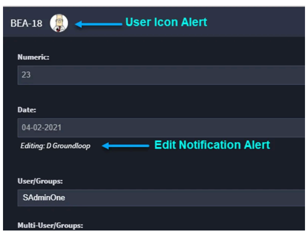
1. Open a record. From the record taskbar, access the record menu.



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 record.

2.2.4.2.2. Coedit Records

Multiple users can edit a record simultaneously. When you access a record that someone else is also accessing, you will see their user avatar immediately following the tracking ID of the record.
As fields are modified, you will also receive immediate notification below the modified field that data has changed, along with an alert of who is making the changes.



2.2.4.5. Advanced

2.2.4.5.1. Lookup and Create References within Records

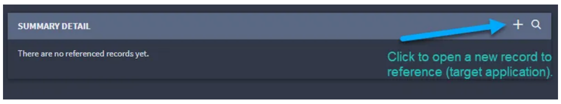
You can select which field(s) to reference from within a record. A reference field on a record, whether single-select, multi-select, or grid, is accompanied by a New and Lookup button adjacent to the reference field.

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

2.2.4.5.1.1. 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.

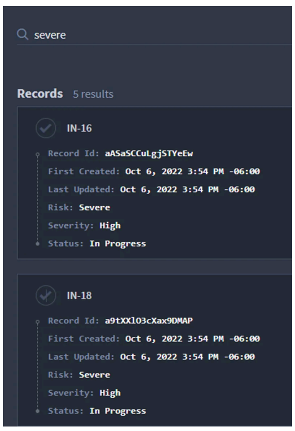


2. Fill out the target record data.

2.2.4.5.1.2. 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.



   The reference lookup searches for records within the referenced target application.
   Note: If your application contains many records, this search can take some time.

2. Once the search has completed, review the results and select the record(s) you want to display.

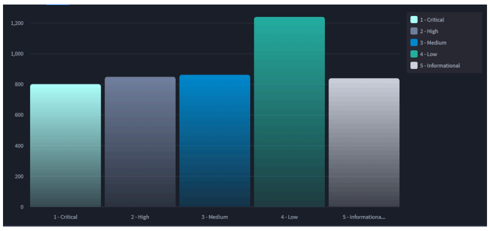
2.2.5.4.2. Chart Types

This topic contains chart type examples.

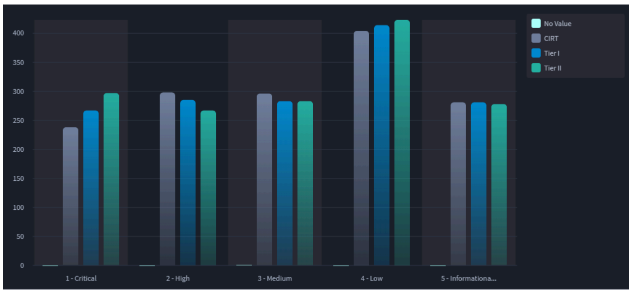
2.2.5.4.2.1. Bar Charts

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

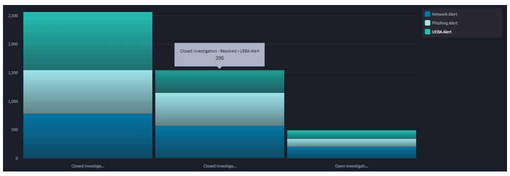
Vertical Bar



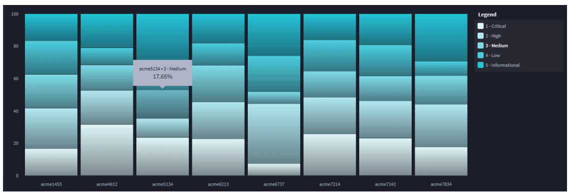
Grouped Vertical Bar



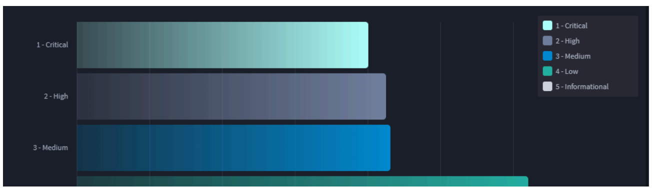
Stacked Vertical Bar



100% Vertical Bar



Horizontal Bar



2.3.1. Playbooks

2.3.1.1. Getting Started

2.3.1.1.1. Playbooks Overview

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.

> To create or change flows with natural language, use **Playbook Building Mode** (Text to Playbook). See **Create and Modify Playbooks with Hero AI** or the **Hero AI** overview.

2.3.1.1.1.1. Playbook Architecture

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

2.3.1.1.1.2. Playbook Data Model

On the canvas, Turbine stores playbook information in **two layers**. Understanding both layers explains what **Save** updates and why some changes appear only after you save from the canvas editor.

Layer	What you see in the UI	What is stored	What runs
Playbook (container)	Playbook name, description, canvas layout, flow order, annotations	A **builder playbook** record that lists which flows belong to this playbook	Does not run by itself; groups flows
Flow	One trigger and its actions on the canvas	A **playbook** (flow) document: automation definition plus **YAML source** (`meta.src`)	This is what executes when the flow is triggered

2.3.1.1.1.2.1. How the Two Layers Relate

- One **playbook (container)** can reference **one or more flows**.
- Each **flow** is a separate automation document linked from the container.
- When you add a new flow on the canvas, the product creates a new flow document and adds its ID to the container.

2.3.1.3.1.1. Flow Event Trigger

Flow Event Triggers enable playbooks to listen for and respond to events emitted by other playbooks using the **Emit Event** native action. This creates a powerful mechanism for inter-playbook communication, allowing you to build complex, distributed workflows where one playbook can trigger another based on specific conditions or outcomes.

2.3.1.3.1.1.1. Overview

Flow Event Triggers work in conjunction with the **Emit Event** native action to create event-driven workflows:

- **Emit Event Action**: Generates events during playbook execution with custom data
- **Flow Event Trigger**: Listens for specific events and initiates playbooks when those events are emitted

This pattern enables you to:

- Decouple workflows into smaller, focused playbooks
- Create reusable playbook components
- Build event-driven architectures
- Trigger multiple playbooks from a single event
- Process events asynchronously without blocking the emitting playbook

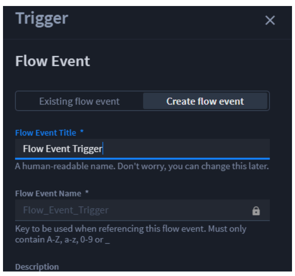
2.3.1.3.1.1.2. Key Benefits

- **Inter-Playbook Communication**: Enable playbooks to communicate and trigger each other
- **Asynchronous Processing**: Emitted events are processed asynchronously, allowing the emitting playbook to continue execution
- **Event-Driven Architecture**: Build complex workflows based on event patterns
- **Reusability**: Create reusable playbook components that can be triggered by multiple sources
- **Scalability**: Events are distributed across scaled services for parallel processing

2.3.1.3.1.1.3. Creating a Flow Event Trigger

To create a Flow Event Trigger:

1. From the **Add** section, drag and drop **Flow Event** to the playbook canvas.
2. Click on the added Flow Event trigger to view the **Trigger** configuration panel.



2.3.1.4.2. Conditions Native Action

Overview

The **Conditions Native Action** enables you to introduce if/else logic into your playbooks, executing different paths based on the evaluation of specific criteria. This action supports the customization and automation of playbooks by visually representing decision-making flows.

2.3.1.4.2.0.1. Key Concepts

- **Condition**: A set of criteria that, when evaluated, determines which path the workflow follows
- **True Path (If)**: The path executed when a condition evaluates to `true`
- **False Path (Else)**: The optional path executed when no conditions evaluate to `true`
- **Condition Builder**: The visual interface for creating and configuring conditions

---

2.3.1.4.2.1. Getting Started

2.3.1.4.2.1.2. Prerequisites

Before using the Conditions Native Action, ensure you have:

- Access to playbook creation and editing
- Understanding of your data structure and available properties
- Basic knowledge of the playbook workflow

2.3.1.4.2.1.3. Steps to Add and Configure Condition

1. From the Add Panel, drag and drop the **Condition** action onto the playbook canvas. The TRUE and FALSE branches will automatically appear.
2. Click **Edit Condition** to open the window that allows you to enter the logic for your conditional statement.
3. If your data has any sensitive information, you can mark it as sensitive by clicking the **Contains sensitive data** checkbox. The data is not shown in the UI or in the logs.
4. Click **Apply** to save changes.

2.3.1.4.2.1.4. TRUE and FALSE Logic

When you drag and drop a Condition native action onto the playbook canvas, the TRUE and FALSE flows automatically display:

- **TRUE (IF):** If the condition is met, the workflow follows the TRUE path
- **FALSE (ELSE):** If the condition is not met, the workflow follows the FALSE path

2.3.1.4.2.1.5. Using the Condition Builder

The Condition Builder provides a visual interface for creating conditions. Follow these steps to

2.3.1.4.7. Using Variables in Playbooks

The **Variables** feature in Swimlane Turbine allows orchestrators to collect, store, and use data across various actions within playbooks. By leveraging the **Create Variables** and **Update Variables** native actions, orchestrators can define and modify variables without writing any code. This enables greater flexibility and ease of use in complex workflows.

2.3.1.4.7.1. Overview

Variables in Swimlane Turbine serve as temporary data storage that can be dynamically manipulated throughout the execution of a playbook. This allows playbooks to make decisions, manage data, and pass information between actions seamlessly.

2.3.1.4.7.2. Key Benefits

- **Dynamic Data Handling:** Easily collect, modify, and utilize data throughout the workflow.
- **Code-Free Configuration:** Create and update variables without the need for scripting.
- **Improved Flexibility:** Enable complex logic and decision-making capabilities by dynamically adjusting the flow of actions.
- **Seamless Integration:** Variables can be used across different actions, ensuring smooth data flow and continuity within playbooks.
- **Promotable Variables:** Promote variables for accessibility across multiple playbooks, enhancing reusability and reducing redundancy.

2.3.1.4.7.3. Creating Variables

The **Create Variables** native action enables orchestrators to generate new variables that can be applied across downstream actions. **Important:** You cannot create a variable that already exists. If you attempt to create a variable with a name that already exists, the action will fail. Use the **Update Variables** action to modify existing variables instead.

Here is how to use it:

1. Start by adding a trigger or existing playbook action to ensure upstream data is available for the variables.
2. From the Add Panel, drag and drop the **Create Variables** action onto the playbook canvas.
3. Click **Configure**, then select **Add variable**. A list of available property types will appear. The following types are currently supported: *String, Number, Boolean, Object, Array, or Attachment*.

> **Attachment** variables can be created and used in downstream actions, but they **cannot** be evaluated in the playbook **Conditions** native action until attachment support is added to the Condition Builder. See **Conditions Native Action** → **Limitations**.

4. If you want to add more than one variable, click **Add variable** again and select another property type.
5. In the variable box, enter a value directly or click **Select a property** to open the property

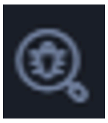
2.3.1.5.1. Test Console

The Turbine Canvas includes a Test Console within the playbook canvas, allowing end-to-end testing of entire playbooks and their trigger flows. This console displays each automation step to indicate whether a run succeeds or fails. You can test using real data or custom test data, offering a flexible environment for playbook verification.

> The Test Console is used for testing entire playbooks. For testing individual actions, use the **Test** tab within the action configuration dialog.

2.3.1.5.1.1. Accessing the Test Console

To open the Test Console, click the **Test** icon on the playbook canvas.



2.3.1.5.1.2. Understanding the Test Console Layout

After opening the Test Console, you'll see the flow being tested on the left side. This layout provides a clear view of each automation step. To the right, there are three main tabs:

- **Runs**: Displays the date and timestamp for each test run, including runs that are **Queued**, **Active**, **Successful**, or **Failed**. As a playbook executes, the Test Console updates in real time so you can monitor progress without waiting for the run to complete. You can expand each run to view detailed information about trigger data, action data, job execution details, and published results.
- **Test Data**: Allows input modification for custom testing. The available options depend on the trigger type:
  - **Webhook Triggers**: Select from past events or enter custom JSON
  - **Record Event Triggers**: Select from historical records
  - **Flow Event Triggers**: Enter custom JSON
  - **Schedule Triggers**: No test data required (test runs immediately)
- **YAML**: (Available when feature flag is enabled) Displays the playbook definition in YAML format.

Icons next to each run provide a quick status:

Status	Description
Queued	The playbook run has been submitted and is waiting to start.
Active	The playbook is currently running. A spinning indicator appears while execution is in progress.
Success 	The playbook completed successfully.
Fail	The playbook failed or was canceled before completion.

2.3.2.2. Components

Components are reusable workflow building blocks that combine actions, inputs, outputs, and interfaces so they can be reused across playbooks.
Components, also known as vendor interaction components (VICs), standardize vendor-specific actions and data for use across playbooks.

Components support two common use cases:

- Ingestion: Retrieve and transform third-party data into OCSF/TEDS objects for use throughout a playbook.
- Enrichment: Retrieve and normalize additional context from external systems to support investigations, threat hunting, and response workflows.

2.3.2.2.1. Choose Your Path

I want to...	Go to...
Create a new component from the Components library	Components Listing Page
Create a reusable component from actions on a playbook canvas	**Create a Component from Playbook Actions** (use **Group**)
Work with canvas tools on a playbook (select, cut, paste, Group)	**Organizing Playbook in the Canvas**
Install a Swimlane Content component	Swimlane Content Components
Configure inputs, outputs, and interfaces	Data and Interfaces
Export, duplicate, or delete a component	Components Listing Page
Mark a component as an AI Agent or use the AI Agents library	**AI Agents in Orchestration**
Build or modify a component with Hero AI	**Create and Modify Components with Hero AI**
Run a component from Hero AI chat	**Hero AI Companion**
Understand how Hero AI executes components	**How Hero AI Executes Components**
Learn naming and interface best practices for AI-enabled components	**AI-Friendly Component Best Practices**

2.3.2.2.2. Components Listing Page

The **Components Listing** page provides a centralized view of all available components in your environment. From this page, you can search for components, apply filters, review component details, and perform common management actions.

To access the Components Listing page:

1. Log in to Turbine

2.3.3.1.2. Events

Within Turbine, orchestrators and practitioners can view event logs for the last 24 hours. You can see the type of event, the name of the agent/host, the date and time stamp that it was received, and how many playbooks that event triggered. You can also click on events to drill into additional details, including skipped playbooks and event-specific data.

Events play a crucial role in triggering automation workflows, helping organizations manage security incidents and other operational tasks efficiently.

2.3.3.1.2.1. Accessing Events

To access events, follow these steps:

1. Log in to **Turbine**.
2. From the left-hand navigation pane, click **ORCHESTRATION, Openrational Health**, and click **Events**.
3. Click on an event to view additional details, such as skipped playbooks and event-specific data.

2.3.3.1.2.2. Benefits of Event Management in Turbine

- **Real-Time Event Tracking:** Immediate visibility into critical incidents and updates.
- **Automation Triggers:** Events automatically trigger playbooks for streamlined workflows.
- **Detailed Logs:** Comprehensive information on event source, time, triggered actions, and skipped playbook reasons.

2.3.3.1.2.3. Types of Events

Events in Turbine can be classified into different types, as shown in the image below:

- **turbine.record.update:** Events triggered when records are updated in the system.
- **turbine.record.create:** Events triggered when new records are created.
- **turbine.custom_webhook.request:** Events generated by webhook requests from third-party tools.
- **turbine.schedule.cron:** Events triggered by scheduled cron jobs or tasks.

2.3.3.1.2.4. Graphical Representation

The graph shows event activity over the past 24 hours, broken down by event type:

- **Blue Line (turbine.record.update):** Indicates record update events, with frequent peaks throughout the 24-hour window.
- **Green Line (turbine.record.create):** Represents record creation events, showing consistent activity with occasional spikes.
- **Gray Line (turbine.custom_webhook.request):** Displays webhook request events, maintaining relatively steady occurrences.
- **Cyan Line (turbine.schedule.cron):** Reflects scheduled cron tasks, which exhibit periodic

2.3.4.1.5.2. Promote Playbook Inputs to Outputs

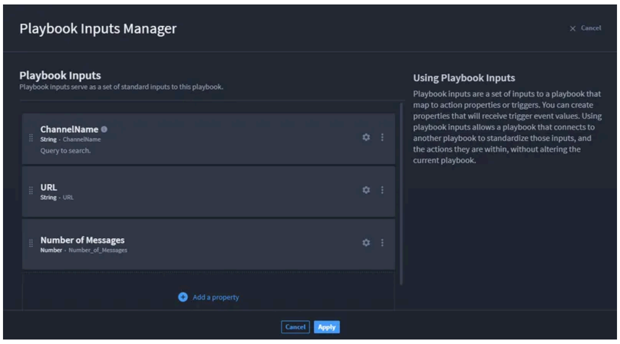
Once you create your playbook inputs or configure action inputs, you can promote the inputs to outputs, then map the outputs to applications. See **Promote Action Outputs** for more information about promoting from the action-level.

> To promote and map playbook inputs to outputs, ensure that you complete the steps to create playbook inputs and/or configure action inputs. See **Create and Edit Playbook Inputs** and/or **Create Actions and Configure Inputs** for details.

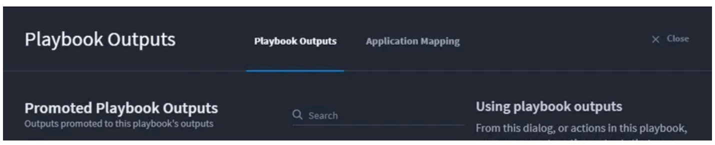
2.3.4.1.5.2.1. Promote Playbook Inputs to Outputs Example

This example has two parts. First, let's add a new property in the Playbook Inputs Manager. This playbook example has a Slack and a VirusTotal action. The inputs have been configured. To add a new property to inputs:

1. From your playbook, click **Playbook Inputs**.
   The **Playbook Inputs Manager** opens and existing input values display. Let's add another input, then promote.
2. On **Playbook Inputs Manager**, click **Add a new property**.
3. From the drop-down menu, select a property.
   Let's add a number value.
4. Click **Number**.
5. On **Property Configuration**, configure the property.
   This example uses **Number of Messages**.



6. Click **Apply**.
7. Now, complete the second part by promoting the playbook inputs to outputs.
8. To promote the example playbook inputs, return to the playbook and follow the steps below.
9. Click **Playbook Outputs** to open the outputs window.



2.3.4.1.9.3. Classic Discovered Outputs and Testing

For every action, there are outputs. Basically, when an action executes (and does not fail), the output data returns a minimum base of property types, which you can apply as inputs for other actions within your playbook, or promote to use outside of the current playbook.

2.3.4.1.9.3.1. Discovered Outputs

Also! There are discovered outputs. Based on what action or connector you use, the results may vary. So in addition to the base of property types, and depending on the inputs, action outputs may return additional properties. These are the discovered outputs.

What if you want to test your action to see the possible outputs? This can be done at the action-level. Let's take a look at the how to test the actions/connectors that Turbine supports.

2.3.4.1.9.3.1.1. Action Testing

Let's test the action that you created! Testing a single action enables better debugging, helps you detect errors in code earlier, saves time, and lets you test inputs while playbook building.

To test connectors/actions, follow these steps:

1. Select your action and click **Configure**.
   The Request, Outputs, and Test tabs are available to configure.
2. From the Test tab, enter your inputs and configure the rest of the action as needed.
3. After entering configuring your request, click **Test** to the right.

The Result pane below provides the raw JSON data that you don't need to worry about! Turbine turns that difficult-to-read JSONata into an easy-to-read format under the **Discovered Outputs** tab. The discovered outputs are combined with the base (original) outputs. All of the known outputs display with check marks. The remaining outputs from the request will be unchecked, and you can select one or all of them and click Add those selected to outputs.

To view the added outputs, navigate to the Outputs tab. All of the outputs that were custom (that you just added) display with the option to delete. You'll notice that you cannot delete any outputs that are part of the original outputs.

2.3.4.1.9.3.1.2. Error Outputs

What happens if your action fails? You need to know more information about why the action failed. Turbine now provides a base-level Error property type for action outputs. The Error property type gives you access to any output properties when an error occurs.

Currently, the HTTP native action, Script native action, and connectors have this property type. This makes for a better user experience when testing actions and handling errors.

Let's walk through an example.

Example:

2.3.4.1.10.4. Classic Record Actions

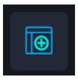
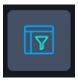
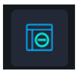
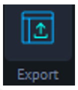
The Create Record, Update/Create Record, Search Records, and Delete Record native actions describe the functions for creating and maintaining data used in Turbine application records. These actions together are also commonly referred to as CRUD.

2.3.4.1.10.4.1. Record Actions in Playbooks

To access these native actions, follow these steps:

From the playbook builder, **ACTION** panel, and drag and drop one of the desired Record actions.

The following table shows the action and associated icon.

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

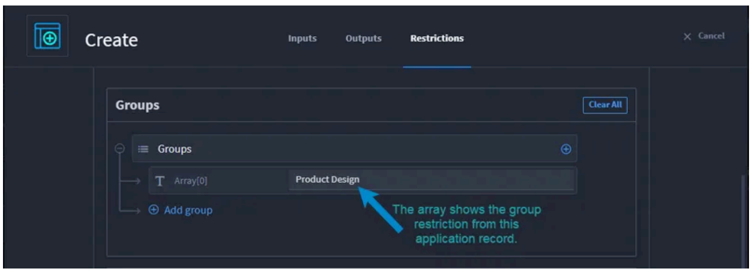
The Create and Update/Create Record actions have the ability to configure inputs, outputs, and restrictions to your record.

Note: Restrictions can be modified on the record as well.

2.3.4.1.10.4.1.1. 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 restrictions, there is a status indicating that no restrictions have been applied. Values you can add when adding a user are: email address, username, display name, or id. Under either the **Groups** or **Users** sections, click **Clear All** to remove all that section's restrictions. If you want to delete just a single group or user, click on the ellipsis to the right of the field, then click **Delete**.



2.3.4.1.11.1. Nested Playbook Triggers

You can create parent playbooks as triggers in nested playbooks. When an action is configured to call a nested playbook, the nested playbook will show that parent playbook as a trigger.

Review **Nested Playbooks** to understand the relationship of a parent and nested playbook before continuing.

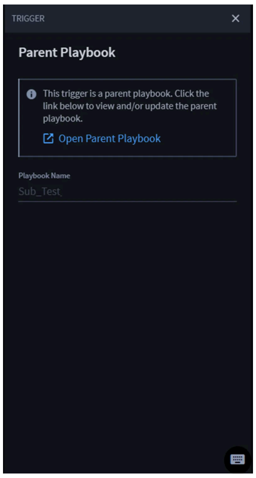
You do not need to select a trigger before adding and configuring actions and conditions.

2.3.4.1.11.1.1. Playbook Button Triggers

If you have more than one playbook as a trigger, click the desired playbook trigger, which will be visible with a blue highlight box around the trigger.

The TRIGGER panel displays. To view the list of parent playbooks and triggers:

1. From TRIGGER, click **Open Parent Playbook**.



The parent playbooks and triggers open in a new tab. From here, view and/or make updates to the parent playbooks.

When calling nested playbooks, Turbine's default configuration is for a maximum depth of 10 playbook levels.

2.3.5.1.1. Export Swimlane Solution Packages

Export Swimlane Solution Packages (SSPs, file extension `.ssp`) to transfer playbooks, applications, applets, builder components, builder solutions, and their dependencies to another Turbine instance or to the Content Library.

> Exporting SSPs requires the **System Administrator** role.

2.3.5.1.1.1. What Is Included in an Export

Included	Excluded
Root application, applet, playbook, builder component, or builder solution	Application history and record history
Associated applications and applets	Records (except when exporting supported application records; see *Export Application Records*)
Workflows for root and associated entities	Secure credentials and key store values (add after import)
Reports, workspaces, dashboards, and export templates	Secure values in asset configurations (reconfigure after import)
Playbooks, including parent and nested playbooks	Connectors (maintain separately)
Orchestration tasks, ingestion rules, sensors, and correlation rules	
Assets, plugins, and Dynamic Orchestration assets	
Builder components, builder solutions, and builder intents	
Mapped inputs to key store values (keys removed; add back after import)	

2.3.5.1.1.2. Export an Application or Applet SSP

You can start an export from the **Applications and Applets** main page or from within the builder.

2.3.5.1.1.2.1. Export from the Applications and Applets Page

1. Open the **Applications and Applets** page.
2. Click the plus icon and select **Import or Export a package**.
3. Select the **Export** radio button.

2.4.2.3. Create the Layout

Now that you have defined your application's foundation, you build out the format of the application. You organize your application in sections, tabs, by playbook, or with HTML.

The layout of your application consists of fields and layout objects. Drag and drop the fields and layout objects to the Form Layout area of Application Builder.

Any of the layout objects can be re-sized to a percentage of the page.

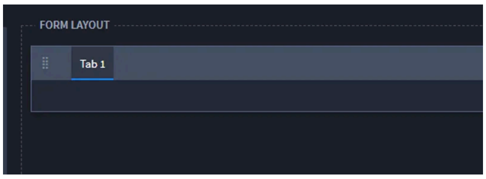
Available Layout Objects

Layout Object	Description
Tab	Groups multiple sections in the same area. Each tab can have it's own layout, which can contain additional sections and tab elements.
Section	Groups multiple fields and layout elements into a single area. Sections can contain additional sections and tab elements and can be collapsed or expanded in the record.
HTML	Displays HTML in the record.
Playbook	Adds a button to the record form that allows you to run a playbook from the records page. You must have already defined the playbook that will run in order to set up a playbook from a record.

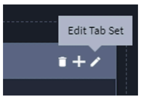
2.4.2.3.1. Creating Tab Layouts

To create tab layouts:

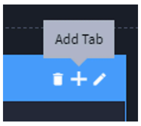
1. From the FORM LAYOUT section of the Application Builder page, select **Tabs**, and then drag and drop it into the Form Layout.



2. Select the pencil icon to edit the tab settings, or double click the default tab title text.



3. To create additional tabs on your tab layout, click the plus icon.

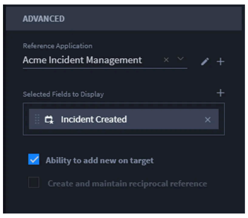


Another tab is added. Name the tab and add additional tabs as needed.

2.4.2.6.7. Reference

Use a reference field to add a reference to another record, either from the current application or from another application.

For example, consider that you have an application called Address with these fields: Address, Zip Code, City, Country; and another application called Employee with these fields: First Name, Last Name. A reference field in the Employee application called "Home Address" could link to the Address application and contain the fields Address and Zip Code. When creating a record in the Employee application, a pre-existing Address record can be referenced, or a new one can be created and referenced on the spot.



When creating a reference field, you can choose from these reference types:

Field Type	Description
Single-select	Use to reference a single record.
Multi-select	Use to reference multiple records.
Grid	Use to reference multiple records with additional, user-specified, field details.

2.4.2.6.7.1. Create Reference Fields

To create reference fields:

From Application Builder's Field Types, click **Reference** and then select which Reference Type you want to create. Drag the reference field to the Form Layout and drop it 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.	*User Reference*
Help Text	Enter contextual help text. You will first need to specify whether the help text will appear above or below the field in the record form, and then you can enter the text.	*Review referenced fields.*

Next, consider the following advanced options in the FIELD PROPERTIES tab:

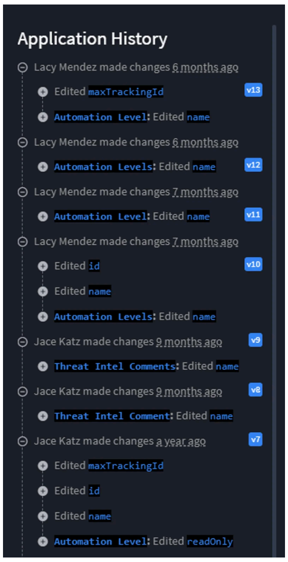
Field	Step	Example
Create New	Select to open Application Builder, from which you	"New Application"

2.4.3.5. View Revision History - Applet

Every modification to the application's layout is stored in the revision history.

- Adding/Removing fields
- Adding/Removing layout elements
- Field configuration changes

To view revision history, access the pull-down menu from the Application Builder taskbar and select **History**



Revisions are listed by version number and show which user made the change and which fields were affected. Expand the revision to see additional detail.

2.4.4. Widgets

Widgets are javascript components that you can use to enrich the User Interface (UI) on records and dashboards.

The flexibility of Swimlane Turbine allows it to cover an unlimited amount of use cases, many of which benefit from enhancements to the UI. Widgets allow you to easily create your own UI components that interact with Turbine or third-party endpoints and enrich your custom use cases, as well as give you access to a growing set of widgets that cover common use cases.

There are currently two types of widgets in Turbine:

- **Record Widgets** — widgets on application and applet record layouts
- **Report Widgets** — widgets created from Charts on reports (under **Reports** > **Chart Configuration**)

To edit widgets with natural language, see **Build Widgets with Hero AI**.

Turbine's widgets, and their supporting code, are hosted here: **Swimlane Platform Custom Widgets**

From the custom widgets repository, you download applets that contain the widgets in order to

2.4.5.4. Getting Started with Workflow

2.4.5.4.1. Accessing the Workflow Builder

2.4.5.4.1.1. For Applications

1. Navigate to **ADMINISTRATOR > APPLICATIONS & APPLETS**
2. Select the application you want to configure
3. Click the **Workflow** icon on the application builder toolbar

2.4.5.4.1.2. For Applets

1. Navigate to **ADMINISTRATOR > APPLICATIONS & APPLETS**
2. Select the application containing the applet
3. Open the applet you want to configure
4. Click the **Workflow** icon on the applet builder toolbar

2.4.5.4.1.3. Workflow Builder Interface

The workflow builder provides:

- **Visual canvas** – Tree view of your workflow structure
- **Zoom controls** – Zoom in/out to focus on specific sections
- **Collapse/Expand** – Collapse or expand branches to focus on specific sections
- **Search functionality** – Search for specific conditions or actions
- **Node editor** – Configure conditions and actions in the right panel

2.4.5.4.2. Navigating the Workflow Builder

2.4.5.4.2.4. Searching

To search for a condition or action:

1. Place your cursor in the search field
2. Enter a term from the name of the condition or action
3. Turbine locates each instance, highlights it, and brings up the editable window
4. If multiple instances exist, use the arrows to navigate between them

2.4.5.4.2.5. Selecting Nodes

- **Click a node** – Select a condition, action, or stage to view/edit its configuration
- **Node editor panel** – Configuration options appear in the right panel when a node is selected

2.4.6.1.6. Statistics

2.4.6.1.6.0.1. AVEDEV

Description: Returns average deviation.

Example:

AVEDEV(number1, number2, ...)

2.4.6.1.6.0.2. AVERAGE

Description: Returns the arithmetic mean.

Example:

AVERAGE(number1, number2, ...)

2.4.6.1.6.0.3. AVERAGEA

Description: Returns the arithmetic mean.

Example:

AVERAGEA(value1, value2,...)

2.4.6.1.6.0.4. AVERAGEIF

Description: Finds the average of the values in a given array that satisfy a given criteria, and returns the average value of the corresponding values in a second given array

Example:

AVERAGEIF(range, criteria, average_range)

2.4.6.1.6.0.5. AVERAGEIFS

Description: Finds the average of the values in a given array that satisfy a set of given criteria.

Example:

AVERAGEIFS(average_range, criteria_range1, criteria1, [criteria_range2, criteria2], ...)

2.4.6.1.6.0.6. BINOMDIST

Description: Returns the cumulative beta probability density function.

Example:

BINOMDIST(number_s, trials, probability_s, cumulative)

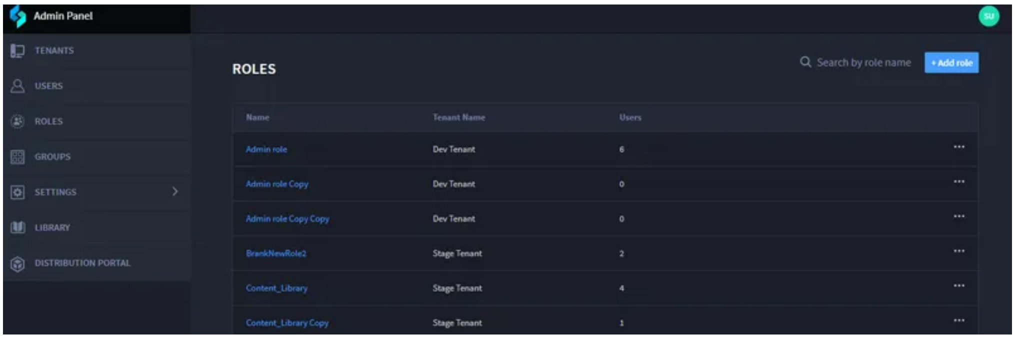
2.4.6.1.6.0.7. CHIDIST

Description: Returns the chi-squared distribution.

Example:

2.5.3.3. Legacy Role-Based Access Control

Administrators manage Swimlane Turbine roles. To access the Roles page, from the Admin panel, select **ROLES**.



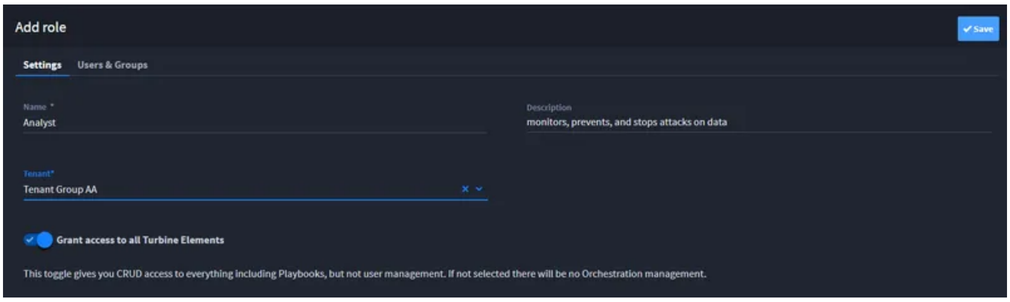
You can view the profile of each role (by clicking on the name) and Search by role name. You can also **Edit**, **Duplicate Role**, or **Delete** a role.

2.5.3.3.1. Adding a Role

Create a new role by clicking **Add Role**.

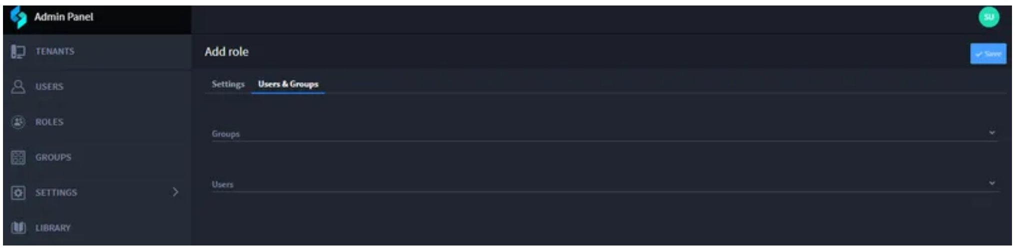
From the Settings tab:

- Name: Enter the role name
- Description: Provide the description for that role
- Tenant: Select the tenant from the drop-down to associate a tenant for that role.



From the Users and Groups tab:

- Click Groups to search for groups, select the group, and add role to that group.
- Click Users to search for users, select the group, and associate that role to that user.



2.5.3.7.3. JumpCloud SCIM Integration

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

SCIM helps customers manage onboarding and off-boarding of users centrally in JumpCloud without logging in to Swimlane Turbine for manual user management.

Users are managed in Swimlane Turbine through their membership in SCIM-provisioned user groups.

Groups are synced automatically through SCIM. However, roles are not provisioned through SCIM and must be assigned manually to each group in Swimlane Turbine. After roles are assigned, the group can be reused for ongoing provisioning.

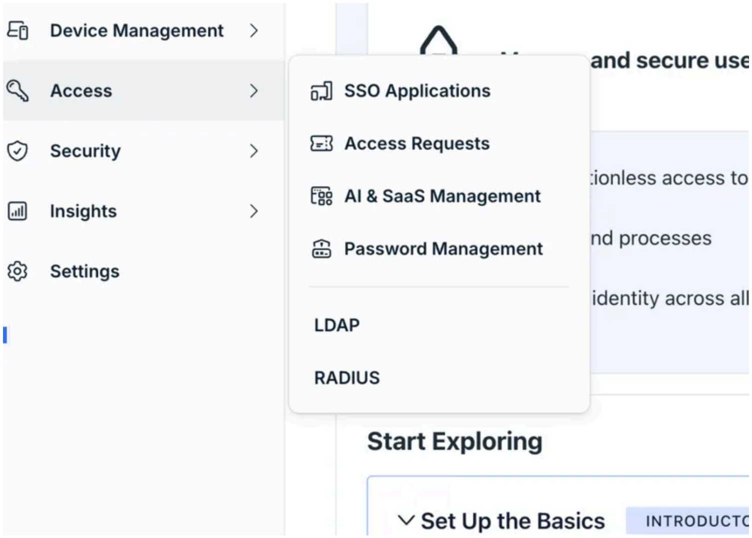
2.5.3.7.3.0.1. Choose Your Path

Goal	Go to
Configure JumpCloud SCIM for Swimlane	**How to Configure JumpCloud SCIM**
Understand attribute mapping	**Field Mapping**
Provision users and groups	**User Provisioning**
Remove users or groups	**User Deprovisioning**, **Group Deprovisioning**
SCIM limits and direction of sync	**Notes**

2.5.3.7.3.1. How to Configure JumpCloud SCIM

To create and configure a SCIM application in JumpCloud for Swimlane Turbine:

1. Sign in to the **JumpCloud Administrator Console**.
2. In the left navigation menu, hover over **Access**, and click **SSO Applications**. **OR,**
   In the left navigation menu, under **User Authentication**, click **SSO Applications**, if you are on an old user interface.



2.5.4.2.1. Connector Trust Keys

Turbine provides a list of connector trust keys. Connector trust keys are used to digitally sign custom built connectors leveraging the **Turbine SDK**.

To view existing connector trust keys:

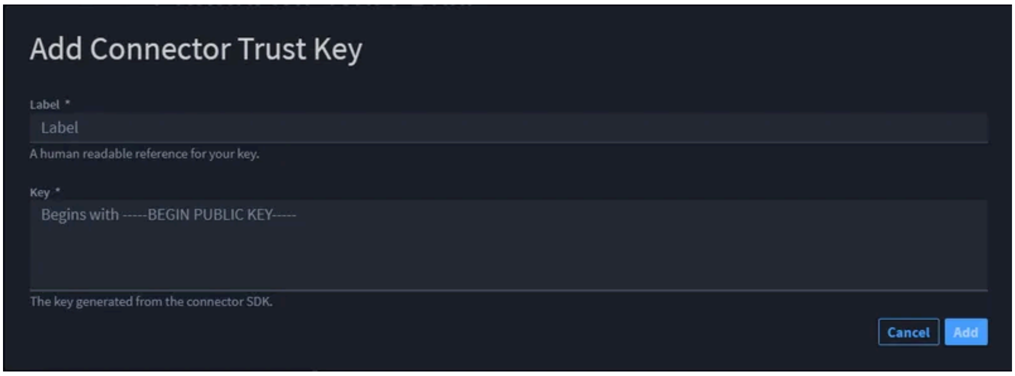
1. Navigate to **SETTINGS** > **Tenant-specific**.
2. Click **Connector Key**.
   The page loads, and Turbine makes an API call to display the current list of connector trust keys from newest to oldest.

Each connector trust key provides the, title of the key, part of the key displays if needed; however the entire key is obscured, date it was created. You can also add or delete a connector from this page.

2.5.4.2.1.1. Add Connector Trust Keys

To add a connector trust key, ensure you are on the Connector Trust Keys page, and then:

1. Click **Add Key**.
   The **Add Connector Trust Key** window opens for you to add a label and a key.
   Important: Both fields are required.



1. In **Label**, enter a title for the key.
   Note: The limit is 50 characters.
2. In **Key**, enter the key that is generated from the connector SDK.
   Warning! If the key is invalid, an error message shows.
   If the key you enter is valid, click the enabled **Add** button to continue.
   Returning to the Connector Trust Keys page, the new key is listed at the top.

2.5.6.1.2. Configuration Manager

The Configuration Manager is a centralized tool within Swimlane Turbine that streamlines asset management across multiple tenants. This feature is designed for MSSP Admins and Orchestrators, allowing users to configure and manage assets outside Canvas efficiently. With powerful filtering, search, and bulk operation capabilities, it significantly reduces manual effort and accelerates workflows for deployment and troubleshooting.

2.5.6.1.2.1. Why Use the Configuration Manager?

- **Centralised Asset Management:** Access and manage all assets across tenants in one place.
- **Seamless Multi-Tenant Management:** Simplify deployment across multiple tenants without switching contexts.
- **Error Visibility:** Quickly identify and resolve configuration errors through intuitive feedback mechanisms.

2.5.6.1.2.2. Key Features

- **Table-Based UI:** View and manage all assets across tenants in an intuitive table layout.
- **Asset Type Support:** Configure HTTP actions, authentication assets, and custom assets.
- **Create, Update, Delete Assets:** Handle individual asset configurations efficiently.
- **Bulk Edit:** Update common fields for multiple assets simultaneously.
- **Error Resolution Feedback:** Detailed error messages to assist with troubleshooting.
- **Search and Filter:** Quickly locate assets using filters and search capabilities.

2.5.6.1.2.3. Who Should Use the Configuration Manager?

- **Orchestrators/Builders:** Advanced users configuring and deploying playbooks who require precise control over asset details.
- **Account Admins/MSSP Administrators:** Users managing multi-tenant configurations with a need for streamlined, high-level asset management.

2.5.6.1.2.4. How to Access the Configuration Manager

1. Log in to Swimlane Turbine.
2. Navigate to the **Admin Panel** > **Library**, select **Configuration Manager**.

2.5.6.1.2.5. Configuration Manager Dashboard

The Configuration Manager provides centralised visibility into asset health and allows administrators to test connector assets individually or in bulk across tenants.

2.5.6.1.2.5.1. Asset Health Metrics

At the top of the Configuration Manager page, summary metrics provide visibility into asset status

2.5.7.1.7. Hero AI Prompts

The **Hero AI Prompts** dashboard tracks **prompt counts** for Hero AI usage across your account. Use it to see how many prompts ran through playbook automation or user-facing Hero AI features, filter by tenant and model, and compare trends over time.

> **Hero AI Prompts** appears under **Admin Panel > Usage**. You need **Admin Panel** read access and **Usage** read permission. For **credit** consumption (not prompt counts), see **Hero AI Credits**. For token volume, see **Hero AI Tokens**.

2.5.7.1.7.1. Choose Your Path

I want to...	Start here
Open the prompts dashboard	**Open Hero AI Prompts**
See prompts from playbooks or components	**Playbook Usage Tab**
See prompts from companion and user features	**User Prompts Tab**
Compare with other Hero AI usage reports	**Related Dashboards**

2.5.7.1.7.2. Open Hero AI Prompts

1. Open **Admin Panel**.
2. Go to **Usage**.
3. Click **Hero AI Prompts**.

The **Usage** menu also includes **Hero AI Credits** and **Hero AI Tokens**.

> Starting in **Turbine 26.2.0**, the legacy **Hero AI** usage page is removed from **Admin Panel > Usage**. Use **Hero AI Credits**, **Hero AI Prompts**, and **Hero AI Tokens** for usage reporting.

> **Historical data:** Filters such as **Playbook**, **Component**, and **Model Type** do not apply to prompt data from before **26.2.0**. The **Tenant** filter continues to work. Summary charts may show totals for older dates while the usage table appears empty. See **Hero AI Credits**.

2.5.7.1.7.3. Playbook Usage Tab

The **Playbook Usage** tab shows prompt counts tied to **playbook and component automation** (for example Hero AI Native Action runs and related canvas usage).

> **Test** runs from an unsaved **Hero AI Native Action** may show placeholder **Action** and **Name** values (for example **Unsaved Playbook** and **Unsaved Flow**). See **Hero AI Credits** in **Hero AI Credits**.

2.5.7.1.7.3.1. Filters

2.5.8.4. Troubleshooting

2.5.8.4.1. Troubleshooting and Diagnostics

Collecting Logs

- Use the logs command mentioned in **Reference: Validation and Commands** section to review and collect logs for support.

Network Test

Validate connectivity to Swimlane Turbine Cloud:

Text

```
docker run --rm quay.io/swimlane/turbine-agent:<version> network-test https://<region>.swimlane.
```

Replace `<version>` with your agent version and `<region>` with your Swimlane Cloud region.

> This test does not work with privately hosted Swimlane Turbine instances. For the test to work, you must allow outgoing connections to `*.amazonaws.com`.

2.5.8.4.2. Reference: Validation and Commands

Command	Action
docker ps	Lists all running docker containers.
docker logs --follow **container-id** or **name**	Gets and follows the logs of a container. Checks for connectors requesting and installing (do not be alarmed if this takes some time) CTRL + C = exit
docker restart **container-id** or **name**	Restarts a container.
docker stop **container-id** or **name**	Stops a container.
docker rm **container-id** or **name**	Removes a container.

2.6.1.2. SOC Interfaces

This document lists the interface contracts available in the SOC Solutions Bundle. For general information about what interfaces are and how to use them, see **Working with Interfaces**. For complete data model field definitions, see **Turbine Schema Reference (Classic SOC)**.

2.6.1.2.1. SOC Solutions Bundle Interfaces

The SOC Solutions Bundle includes 20 interfaces for Security Operations Center workflows. Use these interfaces for alert triage, observable enrichment, email processing, and remediation actions.

2.6.1.2.1.1. Alert to None v1.0.2

Purpose: Processes an alert object without producing output. Use this interface for alert ingestion workflows where you process alerts without transforming them.

Input schema:

Field	Type	Required	Description
`alert`	object	Yes	Alert object
`alert.alert_categories`	array of strings	No	Alert categories
`alert.alert_created_timestamp`	string	No	When the alert was created
`alert.alert_description`	string	No	Alert description
`alert.alert_end_timestamp`	string	No	Alert end time
`alert.alert_impacted_hostnames`	array of strings	No	Affected hostnames
`alert.alert_impacted_ip_addresses`	array of strings	No	Affected IP addresses
`alert.alert_impacted_usernames`	array of strings	No	Affected usernames
`alert.alert_ingested_timestamp`	string	No	When alert was ingested
`alert.alert_mitre_attack_tactic_technique`	array of objects	No	MITRE ATT&CK mappings
`alert.alert_mitre_attack_tactic_technique.tactics`	array	No	Tactic information
`alert.alert_mitre_attack_tactic_technique.technique`	object	No	Technique details with `name`, `uid`
`alert.alert_mitre_attack_tactic_technique.technique.name`	string	No	Technique name
`alert.alert_mitre_attack_tactic_techn`	string	No	Technique UID

3.4. Create and Modify Playbooks with Hero AI

Use **Playbook Building Mode** (also called **Text to Playbook**) to create or change playbook flows on the canvas with natural language. The **Playbook Generator Agent** applies your prompts to the selected flow or flows.

Starting in **Turbine 26.2.0**, **Playbook Builder v2** added **clarifying questions** before building and **progress detail** messages while generation runs.

Starting in **Turbine 26.3.0**, the agent can **edit multiple flows in one request**—select flows on the canvas or name them in natural language, then review and **Accept all** or **Reject all** proposed changes. See **Edit Multiple Flows in One Request**. You can still edit a **single flow** by selecting it on the canvas or naming it in your prompt.

For manual drag-and-drop creation, see **How to Create a Playbook**. For supported operations, limits, and trigger rules, see **Playbook Generator Agent Reference**.

3.4.1. Choose Your Path

I want to...	Start here
Create the first flow in a new playbook with AI	**Create a New Flow with Hero AI**
Change an existing flow on the canvas	**Modify an Existing Flow**
Edit several flows in one Hero AI prompt	**Edit Multiple Flows in One Request**
Accept or reject AI changes across flows	**Review and Accept Hero AI Changes**
Understand what a flow does without editing it	**Ask Questions About Existing Flows**
Answer agent questions before a build completes	**Clarifying Questions** (playbook or component canvas)
See status messages during generation	**Progress During Playbook Generation**
See what the agent supports (including multi-flow)	**Playbook Generator Agent Reference**
Confirm I am in the right Hero AI mode	**Enter Playbook Building Mode**
Understand container vs flow when saving	**Playbooks Overview**

3.4.2. Enter Playbook Building Mode

There is **no separate Building Mode toggle** in settings. The companion enters **Playbook Building Mode** when Hero AI opens **with canvas context** (the product knows which playbook and flow you are editing).

How you open Hero AI	Mode	Typical use
Hero AI from the main application	General	Questions and chat not tied to a specific

4.1.1. Accessibility Conformance Report

- **Name of Product/Version:** Swimlane Turbine Cloud 25.3.0
- **Report Date:** October 2025
- **Product Description:** Swimlane Turbine is a low-code security automation platform. With Turbine you can prioritize alerts, re-mediate threats and improve your operational performance.
- **Contact Information: info@swimlane.com**

> Turbine is a web-only application.

This report covers the degree of conformance for the following accessibility standard/guidelines:

Standard/Guideline	Included in Report
Web Content Accessibility Guidelines 2.2	Level A (Yes) Level AA (Yes) Level AAA (Yes)

4.1.1.1. Terms

The terms used in the Conformance Level information are defined as follows:

- **Supports:** The functionality of the product has at least one method that meets the criterion without known defects or meets with equivalent facilitation.
- **Partially Supports:** Some functionality of the product does not meet the criterion.
- **Does Not Support:** The majority of product functionality does not meet the criterion.
- **Not Applicable:** The criterion is not relevant to the product.

4.1.1.2. Success Criteria, Level A

Criteria	Conformance level	Remarks and explanations
1.1.1 / Non-text Content	🔶Partially supports	Some images that contain text are missing `alt` tags. Some icon button links are missing a `name` that describes its purpose.
1.2.1 / Audio-only and Video-only (Prerecorded)	〰️ Not applicable	The product does not use prerecorded media.
1.2.2 / Captions (Prerecorded)	〰️ Not applicable	The product does not use prerecorded media.
1.2.3 / Audio	〰️Not	The product does not use prerecorded media.

4.2.9. Manual Steps to Re-run Remote Agent Update Cron Job

These steps outline how to manually identify and re-run the remote agent update cron job that normally executes on schedule. This is useful if you need to trigger the update immediately without waiting for the cron schedule.

Manual steps to re-run a remote agent updade cron job

1. List all cron jobs for the current user:

   ```
   crontab -l
   ```

2. Locate the cron job entry containing your host, for example: us1.swimlane.app.

   Example entry:

   ```
   0 0 1 1 * curl -s https://<domain>/orchestration/api/account/<account-id>/tenant/<tenant-id>/v1/agents/update | bash -s -- -d "/usr/bin/docker"
   ```

2. Copy just the command portion (everything after the 5 schedule fields).
   In the example above, that would be:

   ```
   curl -s https://<domain>/orchestration/api/account/<account-id>/tenant/<tenant-id>/v1/agents/update | bash -s -- -d "/usr/bin/docker
   ```

2. Run it manually to trigger the upgrade.

   ```
   curl -s https://<domain>/orchestration/api/account/<account-id>/tenant/<tenant-id>/v1/agents/update | bash -s -- -d "/usr/bin/docker
   ```

4.2.10. Monitoring Script for Remote-Agent Host Health

This article provides a script that customers can use to collect system health metrics from the server running the Turbine remote-agent containers. The goal is to capture key data points (DNS resolution, CPU, memory, I/O, network performance, etc.) at regular intervals so that, in the event of a container restart or connectivity issue, system conditions at the time can be reviewed.

This is particularly useful in environments where connectivity to the Turbine Cloud instance may be impacted by DNS timeouts, network quality issues, or resource exhaustion.

This solution has been tested on commonly used Linux distributions, including:

- **RHEL 7 / 8 / 9**
- **Ubuntu 18.04 / 20.04 / 22.04**

4.2.10.1. What the script collects

4.3.1.2. Configure Numeric Array Condition Expressions

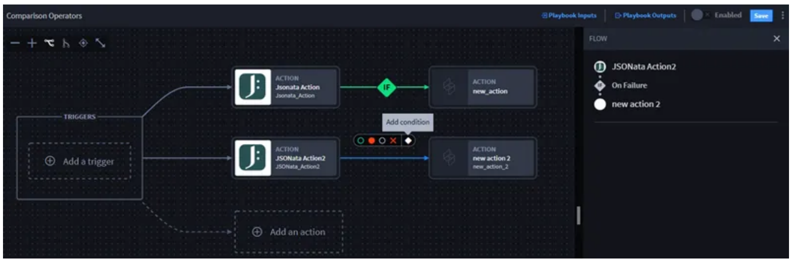
Turbine allows users to configure conditions between two actions by utilizing playbook inputs and/or action outputs.

4.3.1.2.1. Scenario

Alex wants to configure a numeric array in a conditional expression using the output of a JSONata action. Alex is ready to begin. She starts by adding and configuring a JSONata connector with an **On Success** action flow, and now she wants to add a numeric array conditional expression.

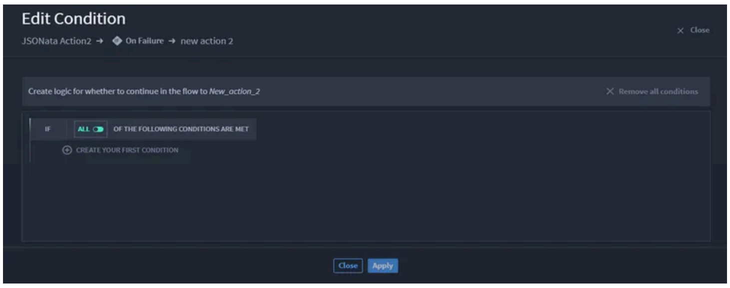
To add a condition, click the **Add condition** icon.

> Once you click the action flow, it turns blue and the FLOW panel displays to the right.



The Edit Condition window opens.

Click **CREATE YOUR FIRST CONDITION**.



The available action/playbook properties displays. Click the property to expand the available input types.

Click the **Numeric Array** property.

> If available, you can select more than one Numeric Array.

The selected array with numeric items output shows in the condition builder.

Select the comparison operator from the drop-down menu.

After the operator drop-down, enter a value.

Click **Apply**.

4.3.3.2. Case and Incident Management Application

> The Case and Incident Management (CIM) application serves as a central point of interaction for a Security Operations team. You can use this application independently or with a solution such as Phishing Triage or Alert Triage from the SOC Solutions Bundle.

> In the following example, the CIM application works as part of the SOC Solutions Bundle. For assistance with SOC Solutions Bundle installation and setup, contact your Swimlane professional services point of contact.

Let's see how the Case and Incident Management application works.

4.3.3.2.1. How it Works

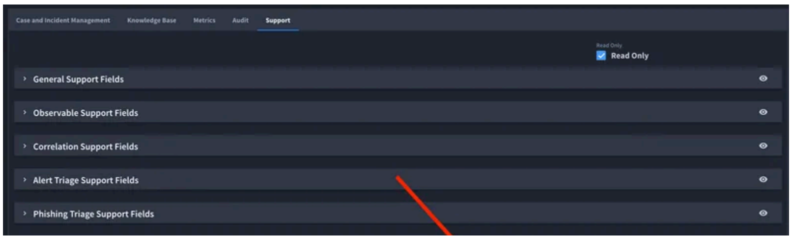
The Case and Incident Management (CIM) application, as a part of the SOC Solutions Bundle, serves as the central point of interaction for a Security Operations team. The application provides the following best practice capabilities:

- Unified signal triage from alert triage, phishing triage, and manual creation playbooks with record creation automations
- Threat Intelligence (TI) enrichment interface
- Various orchestration launch points
- Signal Triage, Case Management, Incident Management, Investigation details, Knowledge Base Articles, Remediation, Correlation, and After Actions Reports
- Dedicated spaces for customizations
- Automatic metric collection
- Advanced mode for troubleshooting and fine tuning

4.3.3.2.2. Correlation Configuration

Turbine can **Correlate Records**, which allows Turbine to compare a new record to a previous record that has correlation keys. In the Case and Incident Management (CIM) application, there is Correlation information on the CIM record in a designated section and on the Support tab.

By default, correlation of CIM records happens when records match at least three observables with other records. This correlation happens at record creation time or when the **Correlate CIM Records** Button is pressed in a CIM record.

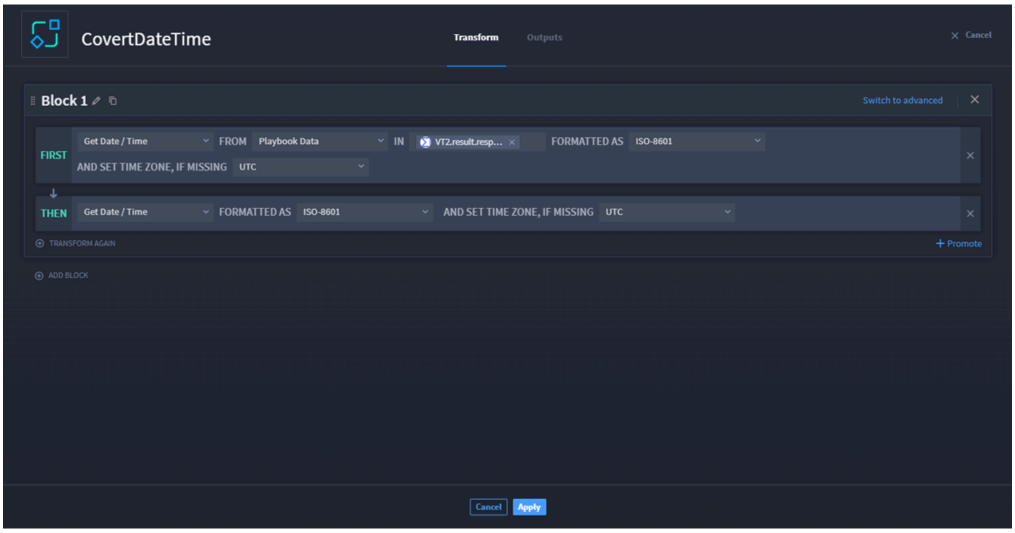


4.3.6.1. Transform Data Action – Use Cases

The Turbine Transform Data native action can create a myriad of transformations. Making changes to date and time in a playbook is often used, and this page provides use cases for different actions for transforming date and time using the Transform Data action. JSONata is used for these actions.

4.3.6.1.1. Get Date and Time

If you configure the FIRST row, click **TRANSFORM AGAIN**, and select **Get Date/Time**, there are new configuration options available.



4.3.6.1.1.1. Transform Block Options – Get Date/Time

Note: When FROM is set to "Current Time", the IN field is not required and will be hidden.

4.3.6.1.1.2. Scenario

After identifying malicious URLs, you want to scan your environment for any other IOCs that may have been seen 90 days before the first sighting. With the Transform Data action, you can automate this task to get a date/time and subtract 90 days.

1. On the **FIRST** row, select **Get Date/Time**.
2. From the **FROM** drop-down, select **Playbook Data** or **Current Time**.
3. If using Playbook Data, select the property containing the date/time in the **IN** field.
4. Configure the format and timezone as needed.
5. Click **TRANSFORM AGAIN**.
6. On the **THEN** row, select **Adjust Date/Time**.
7. From the **BY** drop-down menu, select **Subtracting**.
8. Enter **90** in the amount field.
9. From the unit of measure drop-down menu, select **Days**.