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

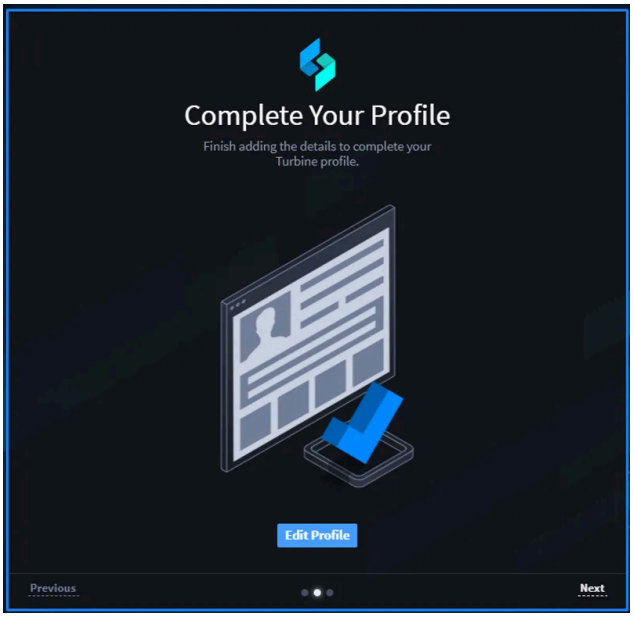
What's Next?

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.

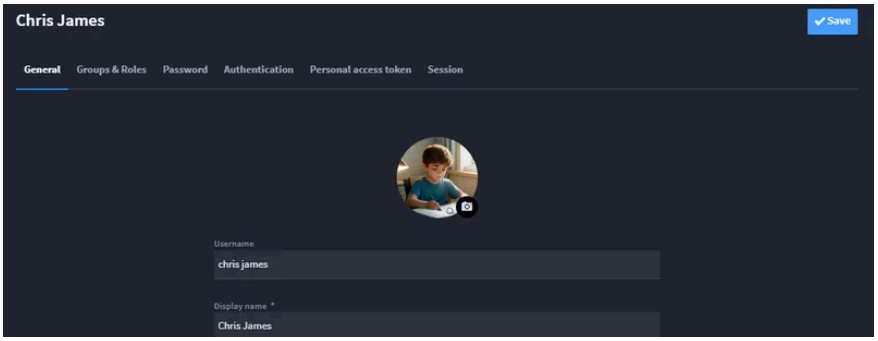
Completing Your Profile

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).



2.2. 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

2.2.3.5. Deleting or Editing a User Dashboard

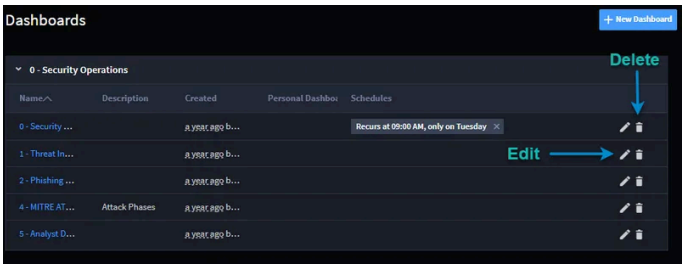
To delete a dashboard:

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.

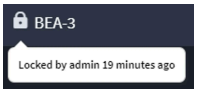
If you have permission, you can also set up private dashboards. A private dashboard can only be viewed or modified by whoever created the dashboard. Administrators or the creator of

2.2.4.2.1. Record Lock

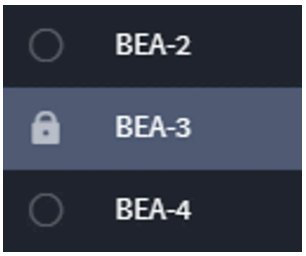
You can lock records while editing and modifying them in order to prevent your changes being overwritten by another user. In addition, locked records can not be bulk modified or bulk deleted.

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.**

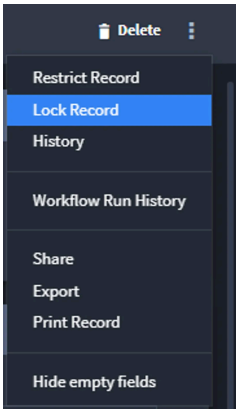


Locked records are also indicated in the report-view-of-records (Default Report) interface.



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

2.2.4.5. Advanced

2.2.4.5.1. Lookup and Create References within Records

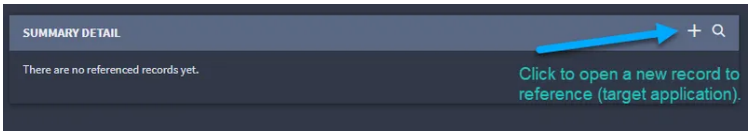
You can select which field(s) to reference from within a record. A reference field on a record, whether single-select, multi-select, or grid, is accompanied by a New and Lookup button adjacent to the reference field.

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

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.

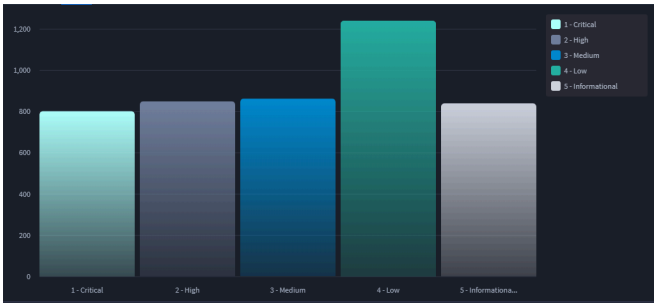
2.2.5.4.2. Chart Types

This topic contains chart type examples.

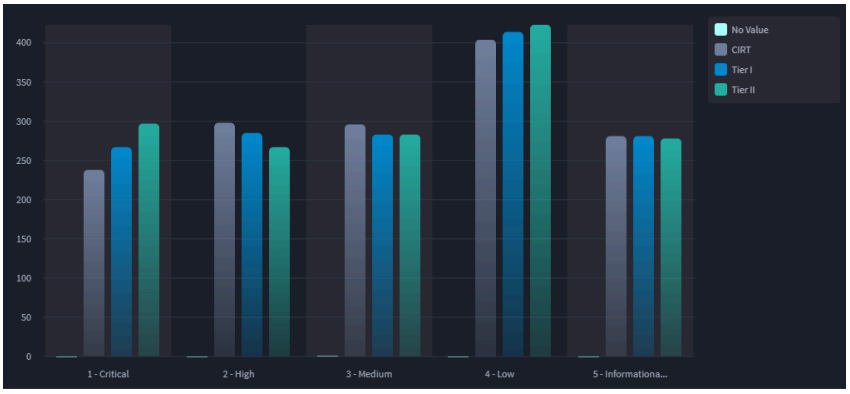
Bar Charts

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

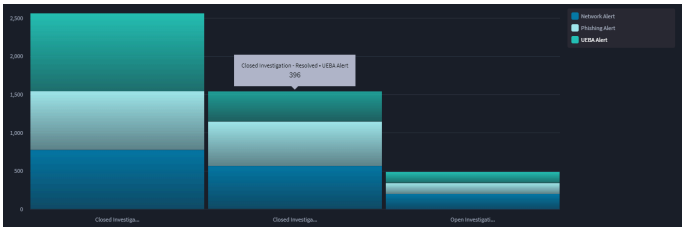
Vertical Bar



Grouped Vertical Bar



Stacked Vertical Bar



100% Vertical Bar

2.3.1.1. 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.

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**](../03-Using Actions/03-Flows/02-02-69-Flows.md).

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	Playbook name, description, canvas	A **builder playbook** record that lists which flows belong	Does not run by itself; groups flows

2.3.1.3.1.3. Record Event Triggers

Record event triggers monitor changes to records in a specified application and automatically execute playbooks when those changes occur. You can configure triggers to fire on record creation, record updates, or correlation search completion, with optional conditions to filter when the playbook should run.

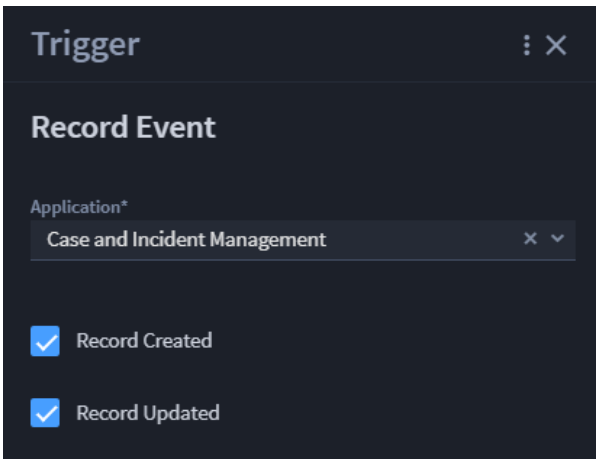
Key Benefits

- **Real-time Automation:** Respond immediately to record changes without polling or manual intervention
- **Flexible Triggering:** Choose to trigger on record creation, updates, or both
- **Conditional Execution:** Use conditions to filter when playbooks run based on field values
- **Access to Record Data:** Full access to current and previous record values in your playbook
- **Correlation Support:** Trigger on correlation search completion for advanced use cases

Creating a Record Event Trigger

To create a record event trigger:

1. In a playbook, from the Add panel, click and drag **Record Event** to the canvas.
2. Hover over the plus icon to add it to the canvas. The Trigger panel displays to the right of the canvas, where you can configure your Record Event trigger.



2.3.1.4.4. Record Actions

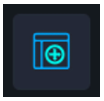
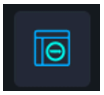
The Create Record, Update/Create Record, Search Records, and Delete Record native actions describe the functions for creating and maintaining data used in Turbine application records. These actions together are also commonly referred to as CRUD.

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

2.3.1.4.9. 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

2.3.1.7. Action Configuration

2.3.1.7.1. 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

2.3.2.3. Assets

Term	Definition	Characteristics
Assets	Saved credentials and key/value pairs that help you connect to technologies, and also serve as a key store for commonly used sets of keys and values	• Can be for a specific connector or customized, which can be applied to any action inputs. • Most useful for standardizing and securing configurations.

Assets can be for a specific connector or customized, which can be applied to any action inputs, and are most useful for standardizing and securing configurations. In Turbine Canvas, assets can be configured in your playbook allowing you to configure a connector and its asset without leaving the playbook canvas.

When to Use Assets

- You need to reuse credentials or common settings across playbooks.
- You want to avoid storing secrets directly in playbook inputs.
- You want consistent configuration for connector actions.

Security Tips

- Use least-privileged credentials for assets.
- Rotate secrets on a regular schedule.
- Review which playbooks depend on an asset before changing it.

Key Benefits of Assets

- **Standardization**: Assets ensure that sensitive data, such as API keys, credentials, and other key-value pairs, are consistently applied across different playbooks, avoiding manual input errors.
- **Security**: By centralizing credentials and keys in assets, they can be managed and

2.3.4.1. 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.

Triggers

One or more triggers automate actions based on the criteria set within a playbook. Swimlane Turbine currently uses four types of triggers to retrieve and/or ingest data.

1. Webhooks
2. Record events
3. Schedules
4. Record action button

Once a playbook has been triggered, a series of actions within that playbook are executed

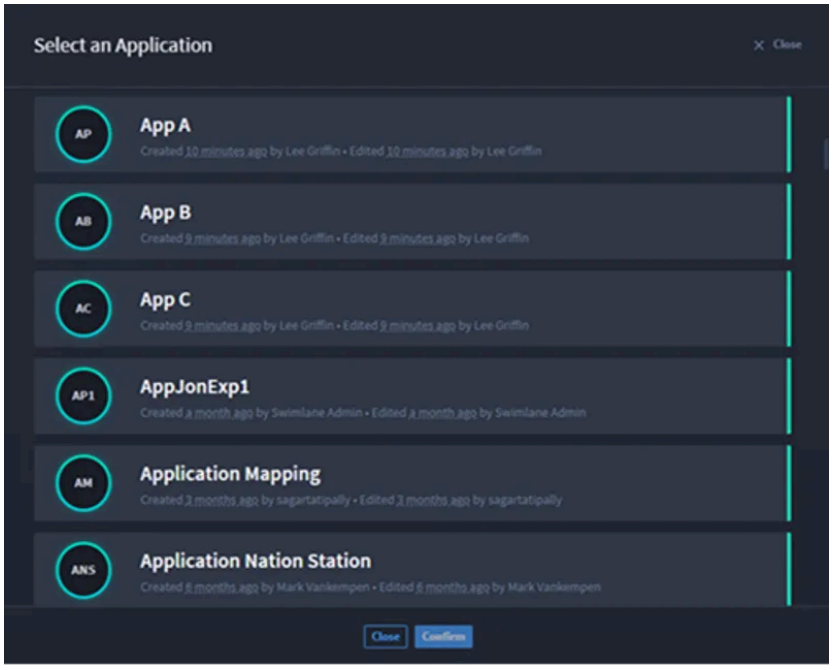
2.3.4.1.5.2. Map Playbook Outputs to Applications

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**

2.3.4.1.8.7. Classic Retries

Playbooks allow you to set action conditions that automate retrying actions based on conditions, time limits, and retry count attempts. Retries allow you to set up the following criteria for an action:

- Interval
- Frequency
- Number of attempts/tries

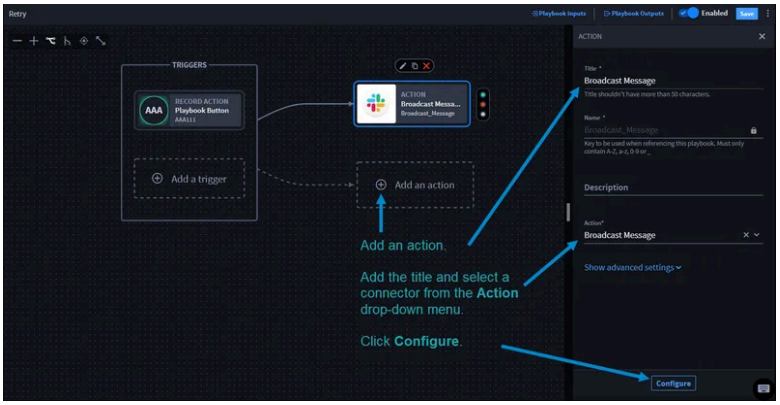
After you set the criteria, you can manage the parameters of that retry by adding conditions. The retry condition settings are similar to how you **Create Action Flows and Conditions**.

> Conditions are **not** required.

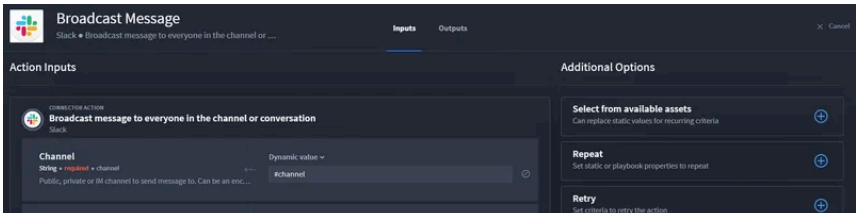
Apply Action Retries

To retry an action:

1. Add and configure an action.



2. From ACTION, click **Configure**.
   Here is where you will configure the action inputs, and then set up the action retry by entering appropriate data in the input fields to set your parameters.



2.3.4.1.9.5. Scripts

As analysts and security professionals, you have knowledge on writing simple scripts in Python. And while Turbine allows you to write more complex code with JSONata, you can also use a native action for simple functions.

> For Boolean and Null data types, we recommend importing JSON and using **json.loads(<my ref>)** to ensure the data loads correctly. Since all playbook data is JSON and Python does not support all JSON types natively.

Use the controlled Script native action and write with Python to:

- manipulate data and edge cases.
- reduce complexity used with JSONata.
- use the most common programming language in security today to do simple tasks.

Turbine uses Python 3.11 and accesses all the standard libraries that come with Python, including the **Python Standard Library**. You can also use **Numpy** version 1.25.2, and **Pendulum** version 2.1.2.

When configuring inputs, consider using the Swimlane Python Chatbot, which uses ChatGPT's Open AI to help you formulate transformations and customized Python code. See **Swimlane Python Chatbot** for information.

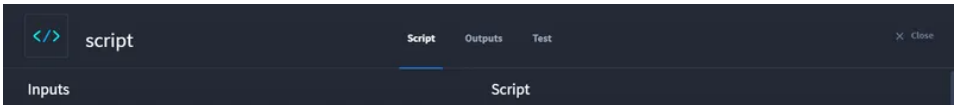
Script Native Action Set Up

Time to start basic set-up for the Script native action.

You have already created a playbook, and you are ready to manipulate data from a property.

1. From your playbook, click **Add an Action**.
2. From the ACTION panel, click the **Action** drop-down menu.
3. Select **Script**, then click **Configure**.

The Script window opens.



2.3.4.1.10.5. Webhook Triggers

Webhooks are a type of event stream that enables products, vendors, or services to push real-time communication in Turbine. They process and enrich data to send from third-party services and platforms to Turbine records.

> Click **Save** or **Save and Close** after making changes.

View Existing Webhooks

To view existing webhooks:

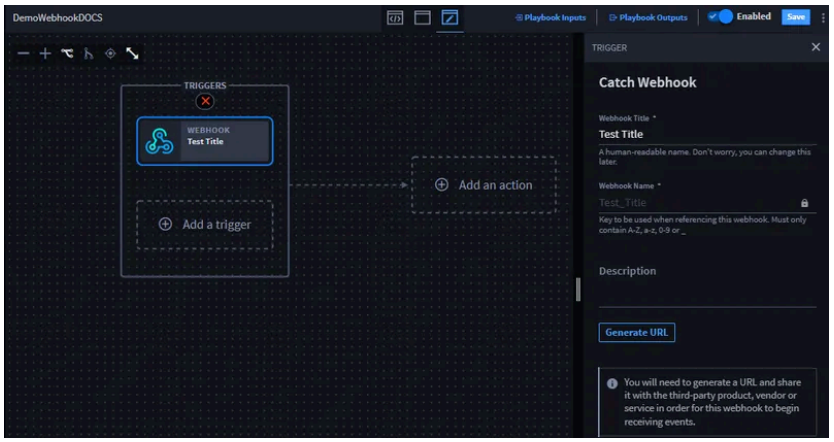
1. Navigate to **ORCHESTRATION**.
2. Click **Webhooks**.

From here you can toggle the **Enabled** button to enable or disable the webhook, or click the desired webhook to view settings and log information.

Create Webhook Triggers

To create a webhook trigger, create a new playbook, upload a playbook, or search for an existing playbook.

1. On the playbook, click **Add a trigger**.
2. Click **Catch Webhook**.
3. On TRIGGER, enter a webhook title and click **Generate URL** or the webhook will not be created.



2.4.1. Getting Started

2.4.1.1. 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.

This section lists all the Applications and Applets that have been created and are available on this Turbine instance. You can filter the list by type (Application and Applets), or by a search string that you enter.

Each Application and Applet provides a management menu with viewing options, copy and export options, and a delete option.

To review or reconfigure the application, click **Builder**.

2.4.1.2. Manage Applications and Applets

The Applications and Applets home page is where you manage your applications and

2.4.2.6.1. Attachments

Use this field to store files in a record. The list of preview options are: .csv, .doc, .docx, .gif, .jpeg, .jpg, .pdf, .png, .ppt, .pptx, .tsv, .txt, .xls, and .xlsx. There are no document-type limitations for attachments.

To create an attachments field:

From Application Builder's Field Types, select **Attachments** 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.	*Related Files*
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.	*Attach related files here.*
Read-only	Click to indicate that the field is read-only for the record. The field will not be editable.	*checkmark*

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

Field	Step	Example
Max Size	Use to specify the maximum allowed file size for the attachment.	*100,000 kb*
Delete Attachments	Select this field to indicate that the attachment be deleted.	*select*
Time to live (days)	Use to indicate the number of days before the attachment is deleted.	*30 (days)*

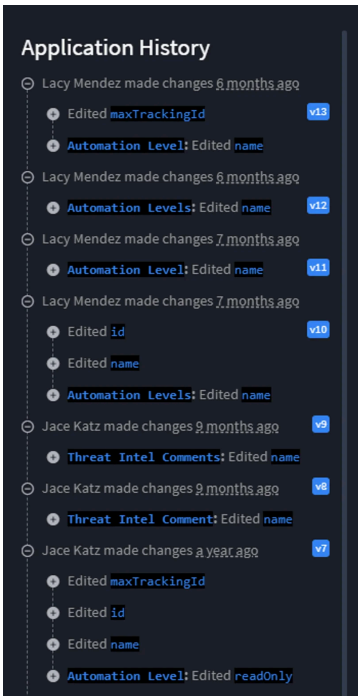
Add specific field-level permissions by role, if needed, and then click **Apply.**

2.4.2.7. View Revision History - Application

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**.

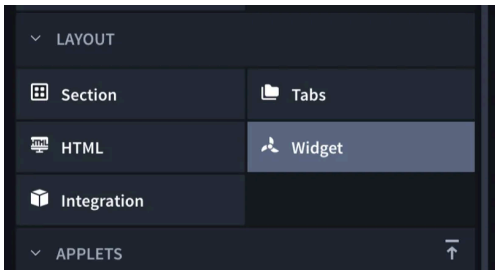


Revisions are listed by version number and show which user made the change and which fields were affected. Expand the revision to see additional detail.

2.4.3. Building Applets

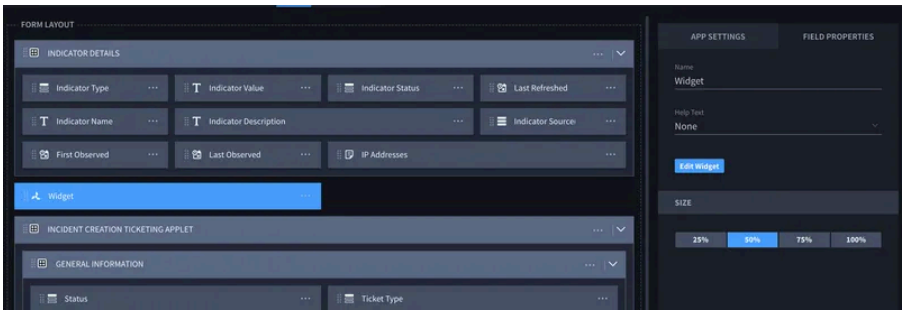
2.4.4.3. Create Widgets

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



> You can add multiple widgets per application or applet.

To edit the widget, select the widget in the layout. Then, in the properties sidebar, click **Edit Widget.**



Widgets are rendered on the record page and have access to the record values through the record attribute. Every change to the record will automatically update the widget, by calling its `update` method.

You can trigger playbooks from widgets. The application from which you are building the widget must contain a playbook button. Use the button property in context data to trigger a playbook upon clicking the playbook button. This reuses the playbook inputs specified in the playbook.

Use this code to set up a playbook button widget:

```
import { SwimlaneElement, html } from '@swimlane/swimlane-element@2';export
default class extends SwimlaneElement {
  handleClick() {
    this.triggerButton(this.contextData.application.buttons[0].id);
```

2.4.5.6. Workflow Best Practices

Best Practices

Workflow Design

1. **Start Simple** - Begin with basic conditions and actions, then add complexity
2. **Use Descriptive Names** - Name stages and actions clearly
3. **Document Complex Logic** - Add comments or documentation for complex workflows
4. **Test Incrementally** - Test each stage as you build it

Performance Considerations

1. **Limit Condition Complexity** - Avoid overly complex nested conditions
2. **Optimize Field References** - Reference fields efficiently
3. **Use Appropriate Operators** - Choose operators that match your data types
4. **Avoid Circular Dependencies** - Don't create workflows that reference themselves

Maintenance

1. **Review Regularly** - Periodically review workflows for optimization
2. **Document Changes** - Keep track of workflow modifications
3. **Test After Changes** - Always test workflows after modifications
4. **Version Control** - Use application versioning to track workflow changes

Common Patterns

Pattern 1: Progressive Disclosure

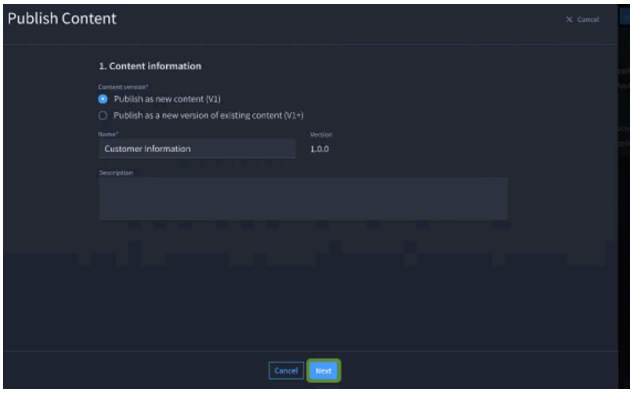
```
If Record Type = "Incident"
  ├── Show Incident Fields
  └── If Severity = "High"
      └── Show High Priority Fields
```

2.4.7. Publishing

2.4.7.1. Publishing an Applet

An Applet can be published to the Turbine Library as content that can be shared across your Turbine account. The publishing process includes:

1. **Publish as New Content (V1)**
2. **Publish as a New Version of Existing Content (V1+)** (including handling merge conflicts if a remote repository has been configured)



Publish as New Content (V1)

This process applies when the applet is being published for the first time.

Steps:

1. Open the applet settings menu and select **Publish Applet** .
2. In the **Publish Content** window:
3. Select **Publish as New Content (V1)**.
4. Provide a name and description for the content.

2.5.3.5. Tenants

Use the **TENANTS** tab to view and manage the list of tenants. From the **TENANTS** page, you can perform the following actions:

- Rename a tenant by clicking the button and selecting **Rename**.
- Add a new tenant by selecting the **Add tenant** button.
- Search for tenants by using the **Search by tenant name** field.
- Edit a tenant's identifier (color or image) by selecting **Edit** from the **More options** menu (**...**).

Tenant Identifier Customization

Each tenant can be assigned a visual identifier to support quick recognition across the user interface. Identifiers include the following:

- **Color:** A random color is automatically assigned to existing and newly created tenants. You can change the color to align with a theme or to help group tenants by category. Specify a HEX value, such as `#b8eafe`.
- Browse for your logo file or drag and drop it into the designated field.
    - **Supported File Formats:** SVG and PNG
    - **Minimum Resolution:** 512x512 pixels
    - **Recommendation:** Use square logos with a transparent background for best results.

Tenant identifiers appear in the following areas:

- The tenant switcher in the top navigation bar.
- The **Identifier** column in the **TENANTS** table.

Tenant Table Columns

The **TENANTS** table contains the following columns:

- **Identifier:** Displays the selected color or uploaded icon.
- **Name:** The name of the tenant. This column supports alphabetical sorting.

2.5.4.1.2. Sessions and Security

Sessions & Security settings are only available to Turbine administrators. Use these settings to configure user session management, password policies, and authentication methods for your account.

This section includes the following topics:

- **Setting Up Session Timeout Parameters**: Configure how long user sessions remain active before requiring re-authentication
- **Setting Up Security Parameters**: Configure password policies, complexity requirements, and failed login attempt limits
- **Enable Two-Factor Authentication**: Add an extra layer of security by requiring a second authentication factor
- **Enable SAML for SSO**: Configure single sign-on using Security Assertion Markup Language (SAML) under the **Authentication** section
- **Global Logout Behavior**: Control whether users are logged out of all devices or just the current session when they log out

2.5.4.1.3. Directory Services

Swimlane Turbine integrates with two Directory Service types: Microsoft's Active Directory (AD) and Open LDAP. By integrating with Directory Services, administrators can streamline user management and ensure consistent authentication across their organization.

Use Cases and Benefits

Many Turbine administrators leverage Directory Services to:

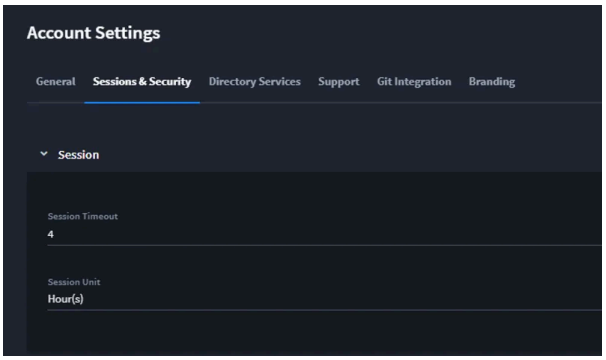
- Enable SOC Engineers and Analysts to log in to Turbine with previously established directory credentials.
- Automate user and group management, reducing administrative overhead for large teams.
- Increase security by centralizing authentication and maintaining compliance with

2.5.5.4. Setting Up Session Timeout Parameters

As an administrator, you control how long a Turbine session can remain idle before the user must re-authenticate or log out and log back in. The timeout is based on **inactivity**, not total time signed in. User activity such as mouse movement, keyboard input, or touch resets the idle timer.

To set up session timeout parameters:

1. From the **Sessions & Security** tab, click **>** to expand the **Session** controls.
2. Specify the length and the unit of time for the user session, then click **Save.**
3. Use the drop-down menu to specify the **Session Unit**, such as *Hour(s)* or *Minute(s)*.



> When you adjust the timeout session parameters here, users will need to log out and log back in for the updated time frame to apply. All user sessions will retain the previously-set timeout value until their current sessions expire or they log out and log back in.

Overriding a User's Session Timeout

Administrators can override a specific user's session timeout. This can be useful, for example, when using a Turbine session as a dashboard for a shared work area. It can also be helpful when you need to immediately disable a user's session.

To override a user's session timeout:

1. From the **Admin Panel**, select **Users**.
2. Choose the user whose session timeout you want to override from the Users list.
3. In the user's profile settings, click the **Session** tab.

2.5.7.1.3. Events Dashboard

The **Events** dashboard allows you to track event activity across tenants. The following metrics are displayed:

- **Total Events:** Displays the total number of events for the selected date range, tenants, and event types.
- **Average Events per Day:** Displays the average number of events per day based on the selected filter. For example, if the last 30 days Date Range filter is applied, the dashboard calculates the average by dividing the total event count by 30.
- **Events Graph:** A graph visualizing event trends over the selected date range:
  - The horizontal axis shows the dates.
  - The vertical axis shows the number of events.
- **Events Table:** The table provides details such as:
  - Tenant
  - Event Type
  - Events
  - Triggered Playbooks
- You can sort the table by clicking on the column headers.

Filters

You can configure the Events dashboard using the following filters:

- **Date Range**: Filter event data based on a specific time period. Only data from the **last 90 days** is available. Options include:
  - Today
  - Last 7 days
  - Last 30 days (default)
  - Last 90 days
  - Custom Date Range
- **Tenant**: Select tenants to filter events.

2.5.8.3. Assigning Pools to Actions

About Pools

- Pools determine which agents execute specific playbook actions.
- Agents can belong to multiple pools; pools can contain multiple agents.

Examples

- Assign agents in different networks to separate pools for resource isolation.
- Assign agents across zones to the same pool for high availability.

Assigning a Pool in a Playbook Action

1. In your playbook action, click **Show advanced settings**.
2. The default is $default. Use the dropdown to select your desired pool.
3. Save your action.

Important: Transformation and Python actions do not function with remote agents.

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.

2.6.1.1. AI SOC Interfaces

This document lists the interface contracts available in the AI SOC Solution. 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 (AI SOC)**.

AI SOC Interfaces

The AI SOC Solution provides the following interfaces for building components and playbooks. These interfaces use the extended Turbine Schema fields defined in **Turbine Schema Reference (AI SOC)**.

Alert to Alert

Purpose: Converts alert objects while preserving all alert data. Use this interface for alert normalization, enrichment, and transformation workflows.

Input schema: Full Alert object (see **Turbine Schema Reference (AI SOC)**)

Output schema: Full Alert object (same structure as input)

Use cases:

- Alert data normalization and transformation
- Alert enrichment pipelines
- Cross-platform alert data exchange

Alert Triage Ingestion to Array of Alert

Purpose: Ingests alerts from alerting tools (SIEM, EDR, AV) using time-based search parameters and returns an array of standardized alert objects.

Input schema:

Field	Type	Required	Description
`organizati`	Strin	Option	The organization impacted by the alerts

3.2.1. Playbook Generator Agent Reference

The **Playbook Generator Agent** runs in **Playbook Building Mode** on playbook and component editor pages. It creates or modifies **one flow at a time**. For procedures, see **Create and Modify Playbooks with Hero AI**.

Supported Capabilities

The agent can:

- Create and modify a playbook (individual flow) or a component

Component:

- Modify user-defined interfaces (input and output schemas)
- Publish fields to the properties in the output schema

Playbook:

- Add a new flow to a playbook or modify an existing flow
- Add, modify, or delete **CRON** triggers
- Delete Record, Flow, and Webhook triggers
- Modify conditions for Record, Flow, and Webhook triggers
- Add, configure, and delete components available in the tenant
- Add, configure, and delete actions of connectors available in the tenant
- Add or change the asset used by a connector action
- Add, configure, and delete native actions: condition, loop, parallel, create/update variable, search records, create record, delete records, scripts, transformations, Hero AI
- Connect actions with on-success, on-failure, and on-complete connections

Not Supported

- (Playbook) Add, replace, or configure schemas for Flow, Webhook, or Record triggers
- (Component) Add or modify existing interfaces (intents)
- Enable or disable a flow via prompt

4.1. Reference

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)

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

4.2.8. 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"
   ```

3. 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
   ```

4. 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
   ```

4.2.9. Monitoring Script for Remote-Agent Host Health

This article provides a script that customers can use to collect system health metrics from

4.3.1.2. Configure Numeric Array Condition Expressions

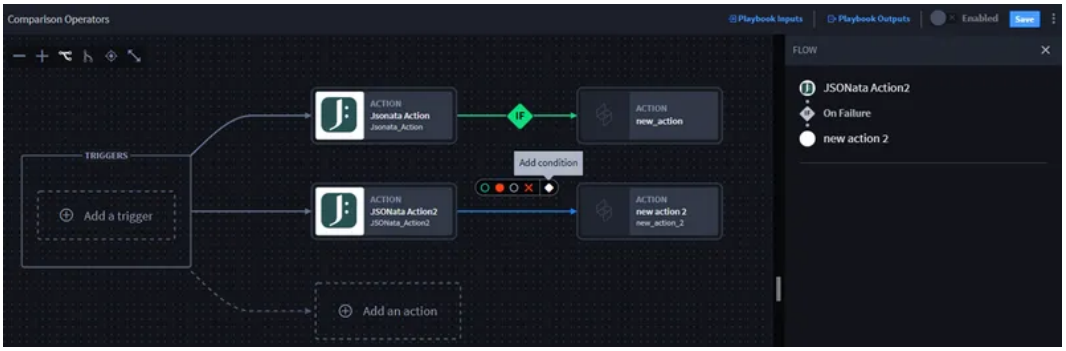
Turbine allows users to configure conditions between two actions by utilizing playbook inputs and/or action outputs.

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.

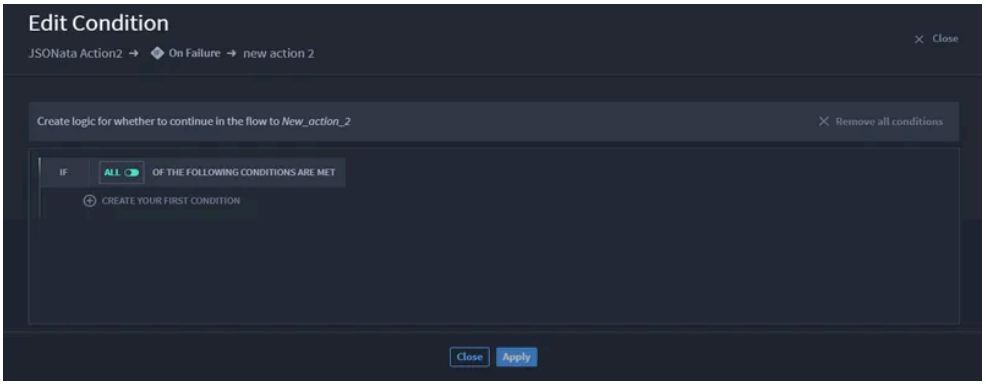
To add a condition, click the **Add condition** icon.

> 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

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.

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

Correlation Configuration

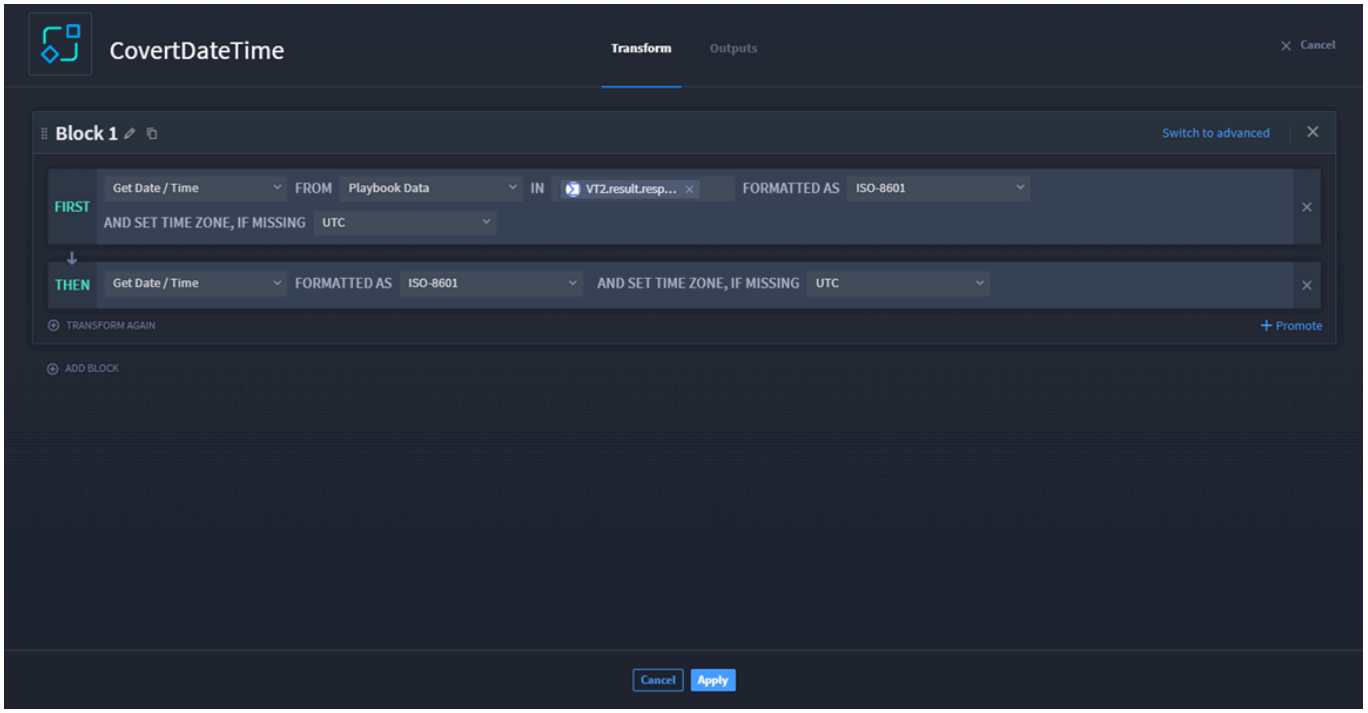
Turbine can **Correlate Records**, which allows Turbine to compare a new record to a previous record that has correlation keys. In the Case and Incident Management (CIM) application,

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.

Get Date and Time

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



Transform Block Options - Get Date/Time

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

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**.