

# PagerDuty + Ayehu Integration Benefits

---

- *Reduce resolution time of incidents by triggering automated remediation workflows in response to alerts from PagerDuty*
- *Cut down on opex by removing manual labor from the loop - remediation can run completely unattended (or involve a human in the loop for approval of sensitive actions, if needed)*
- *Bi-directional communications update the status of the incident in PagerDuty*
- *All actions are logged in detail, providing a full audit trail of work performed*

## How it Works

---

- Incidents in Pager duty trigger execution of a workflow in Ayehu NG, which automatically remediates the incident, and updates PagerDuty with the status upon completion
- Once the incident was remediated successfully, a resolve event will be sent to the PagerDuty service to resolve the alert, and associated incident on that service.
- The main workflow in Ayeandles 3 types of events from Pager Duty:
  - If the event type is "incident.trigger" it will open a new internal incident within Ayehu NG. this incident should be setup to trigger a workflow to assist with remediation.
  - If the event type is "incident.resolve" this will close the internal incident within Ayehu NG. if a policy action with a recovery workflow is setup, it will execute it.
  - If the event type is "incident.custom" this will call a sub workflow associated with the "Action" chosen in the Pager Duty UI. Action name should match a workflow name.

## Requirements

---

- A working instance of Ayehu NG 1.6 or later is required. The latest version can be downloaded here: <https://info.ayehu.com/download-free-30-day-trial-ng>
- Ayehu NG Web service configuration to be enabled to receive inbound events from Pager Duty (modify firewall rules as needed)
- A PagerDuty account with available an Auth Token and Routing Key

## Support

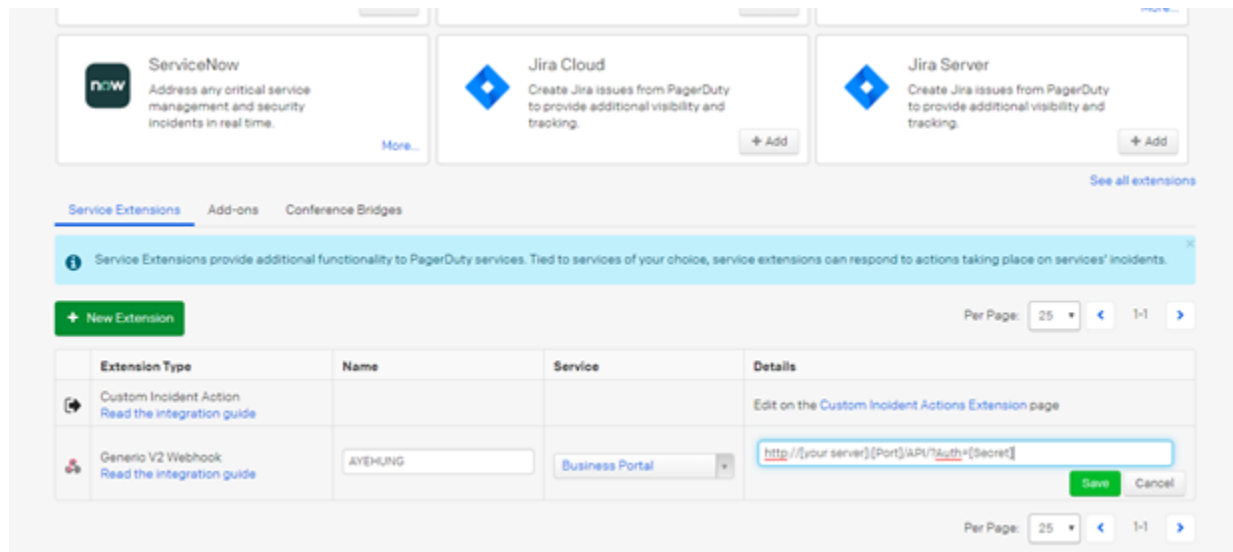
---

If you need help with this integration, please contact [yaron@ayehu.com](mailto:yaron@ayehu.com)

# Integration Walkthrough

## In PagerDuty

Create a service extension to the desired services you wish to communicate to Ayehu NG every time they will be triggered.

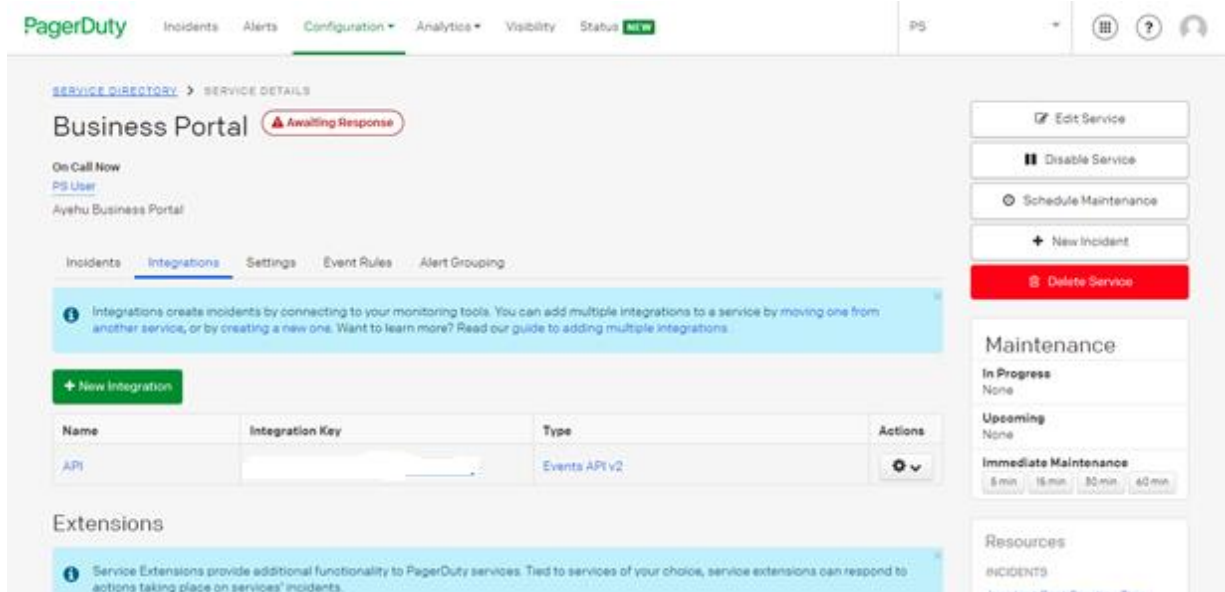


Use: `http://[your server]:[Port]/API/?Auth=[Secret]`

- \*\* Server should be the one that hosts the Ayehu NG Web Service
- \*\* Port and Secret will be the ones you configured in Ayehu (see below)

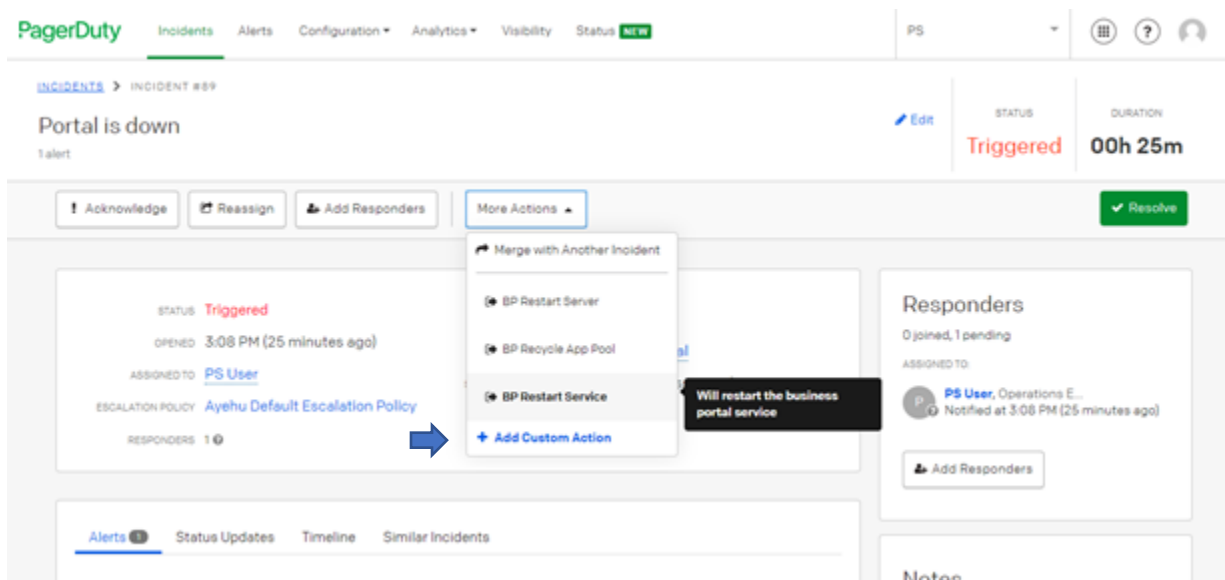
Test by triggering the service with expected new audit trail captured in Ayehu NG.

A PagerDuty Routing key should be obtained for the incident to be triggered:



Create custom incident actions if needed. This should use the same URL to forward to Ayehu NG.

\*\* button name should match exact workflow name to execute.



PagerDuty Incidents Alerts Configuration Analytics Visibility Status **NEW** PS

[EXTENSIONS](#) > CUSTOM INCIDENT ACTIONS

## Custom Incident Actions

[+ New Action](#)

Per Page: 10 < 1-3 >

Button Label	URL Endpoint	Service
BP_Restart_Server		Business Portal
BP_Recycle_App_Pool		Business Portal
BP_Restart_Service		Business Portal

Per Page: 10 < 1-3 >

**About this extension**

Create a button on PagerDuty incidents, in both web and mobile, which fires a webhook to your specified URL. This enables you to automate actions like restarting a server or triggering an action from a run book.

[Read More](#)

[EVENT RULES](#) > GLOBAL EVENT ROUTING

## Event Rules

### ▼ Incoming Event Source

To use event rules, configure your monitoring tools and other integrations to send events using your new account-level integration key. Event rules are compatible with [200+ integrations](#), including:

splunk >

**DATADOG**

**Nagios**

New Relic

[More...](#)

Integrations will require either a PagerDuty integration key, an e-mail address, or an HTTP endpoint which are listed below.

Integration Key	<input type="text" value="R259VN2DYOURR8JVSABVKEY01X4CWQ"/>	Copy to Clipboard
Email address	<input type="text" value="R259VN2DYOURR8JVSABVKEY01X4CWQ@yourname@company.com"/>	Copy to Clipboard
HTTP endpoint for API	<input type="text" value="https://events.pagerduty.com/v2/enqueue"/>	Copy to Clipboard

The integration key must be part of the [API v2 payload](#)

[+ New Event Rule](#)

## In Ayehu NG

### Step 1: Import activities

- Obtain Ayehu NG PagerDuty activities from Ayehu Github Community.

<https://github.com/Ayehu/custom-activities/tree/master/PagerDuty>. Download ALL pager duty activities binaries. For each one, only the AYH file is required. Details for each activity may be found in the README file.

- Open the Ayehu NG activity designer
- Create a new category named PagerDuty by selecting “New Category” on the left menu.
- Import all the activities (AYH files) from Step 1, by selecting the “New Activity” option from the left menu and then clicking on “Import” on the top right.
- Enable the activity by clicking “Enable” on the top right.
- Click on the “Save” on top right and then select the folder to be saved “Pager Duty”
- Do this for all the activities, one by one.

## **Step 2: Setup Ayehu NG Webservice**

The webservice will be used to receive events coming from PagerDuty. This web service can be configured to use any custom port as long as it's available to access PagerDuty

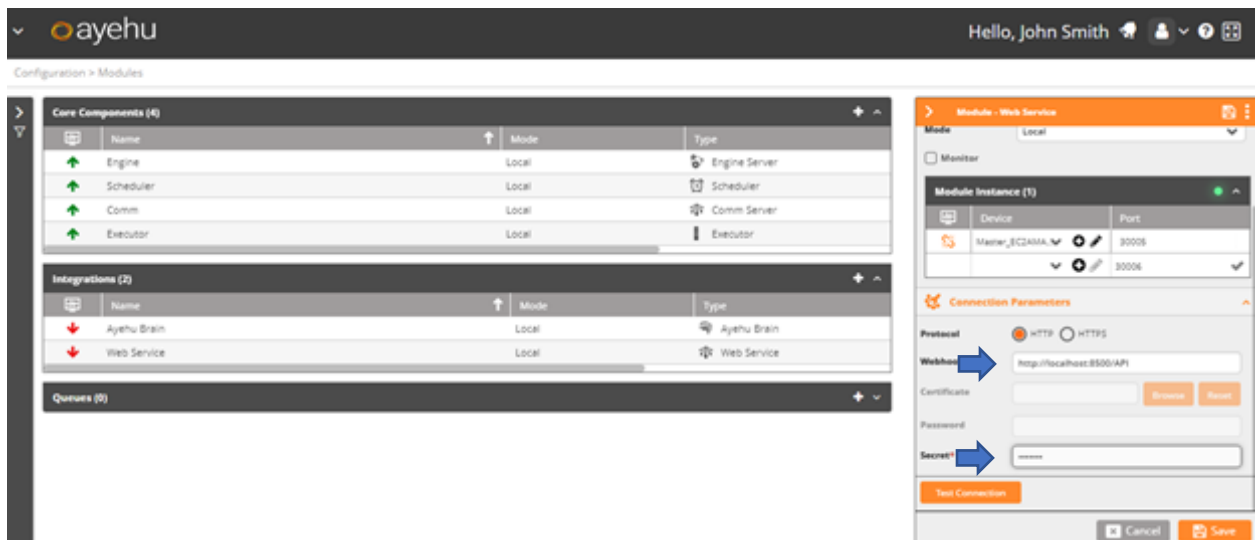
Note, the Ayehu NG WebService Module can be installed on a different network to allow the above with accessible custom port from the internal Ayehu NG to communicate with.

1. Go to the Ayehu NG modules settings
2. Under the integrations tab, Click the “Add” button and configure the following:



The automatic suggested port should be available for inbound from the Ayehu NG Comm server module if installed on a separate server.

Click “Save” and additional configuration options will be available.



\*\* You may use HTTPS by providing a valid certificate.

\*\* The port in the URL can be customized to any port, as long as there's access from PagerDuty. [http://localhost:\[port\]/API](http://localhost:[port]/API)

\*\* secret string can be anything you want.

Once saved, it will take few moments for the module status to become in green state.

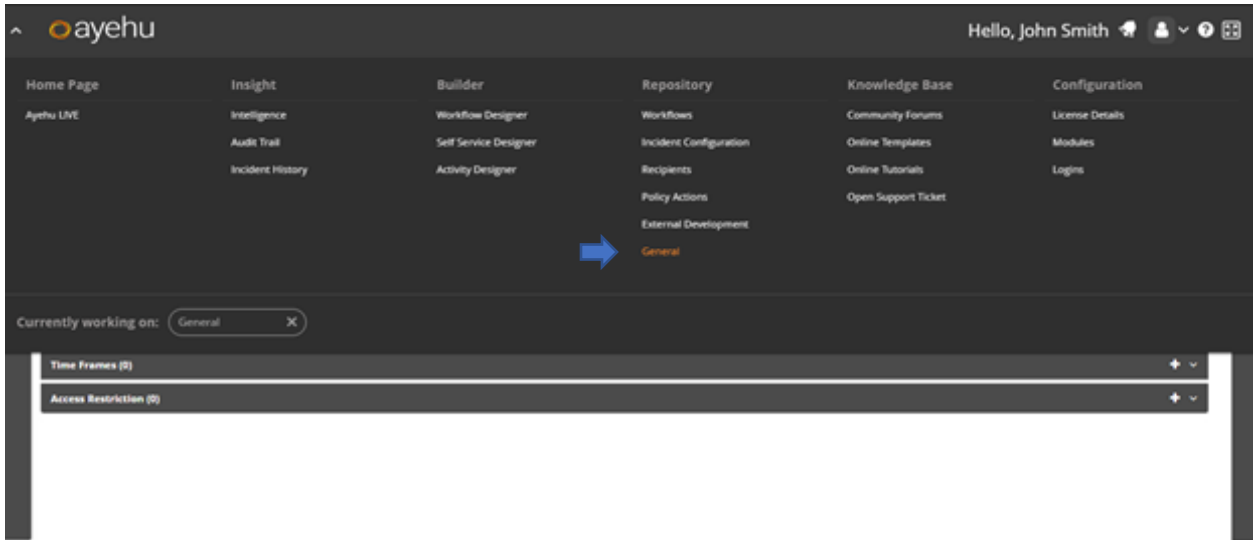
### **Step 3: Setup Workflows and Policy with Ayehu NG**

1. Import the main workflow to handle all incoming PagerDuty events.
2. <https://github.com/Ayehu/custom-workflows/tree/master/PagerDuty>

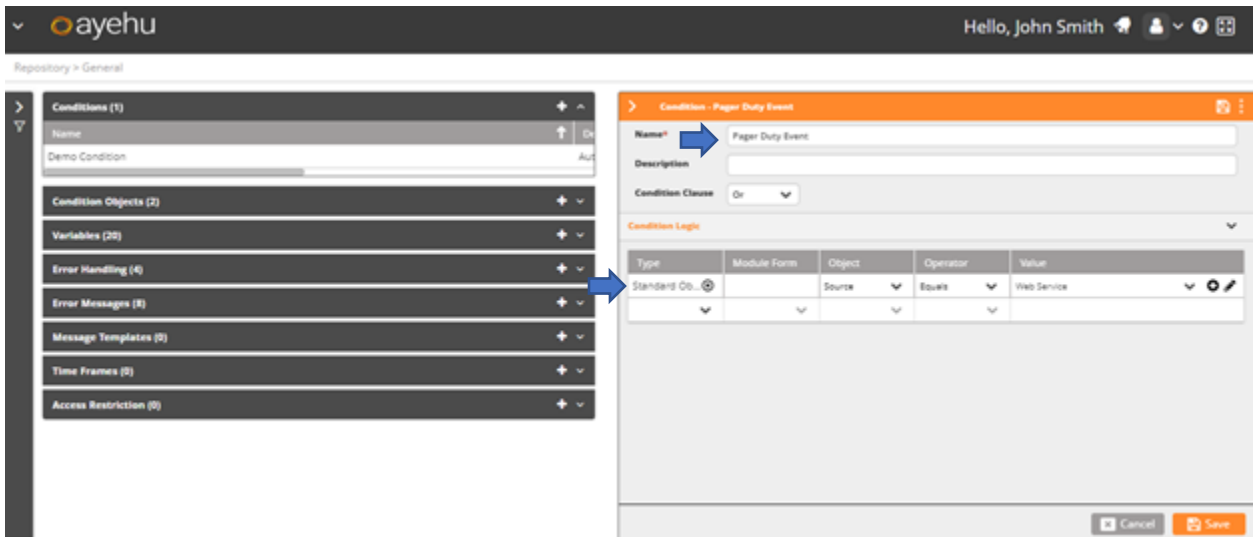
This workflow will be triggered every new event received from PagerDuty. It handles 3 scenarios as follows:

- a. If the event type is "incident.trigger" it will open a new internal incident within Ayehu NG. this incident should be setup to trigger and other workflow to assist with remediation. A new workflow and policy action will be required.
  - b. If the event type is "incident.resolve" this will close the internal incident within Ayehu NG. if a policy action with a recovery workflow is setup, it will execute it.
  - c. If the event type is "incident.custom" this will call a sub workflow associated with the "Action" chosen in the Pager Duty UI. Action name should match a workflow name.
3. Set a policy action to execute this workflow every time Ayehu NG receives event via the web service.

First create the condition object,

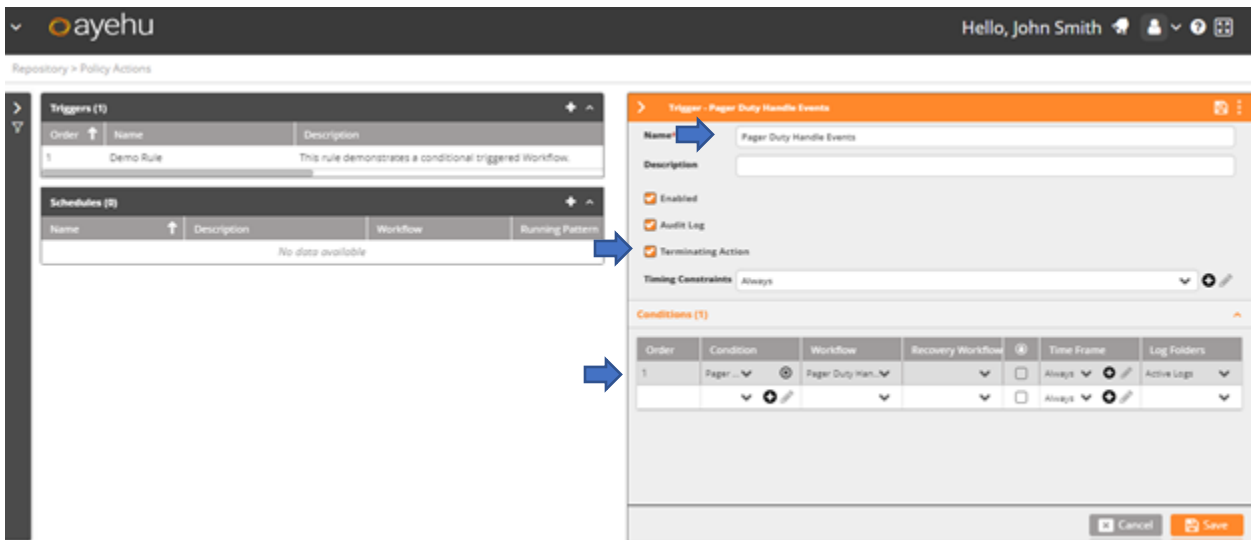
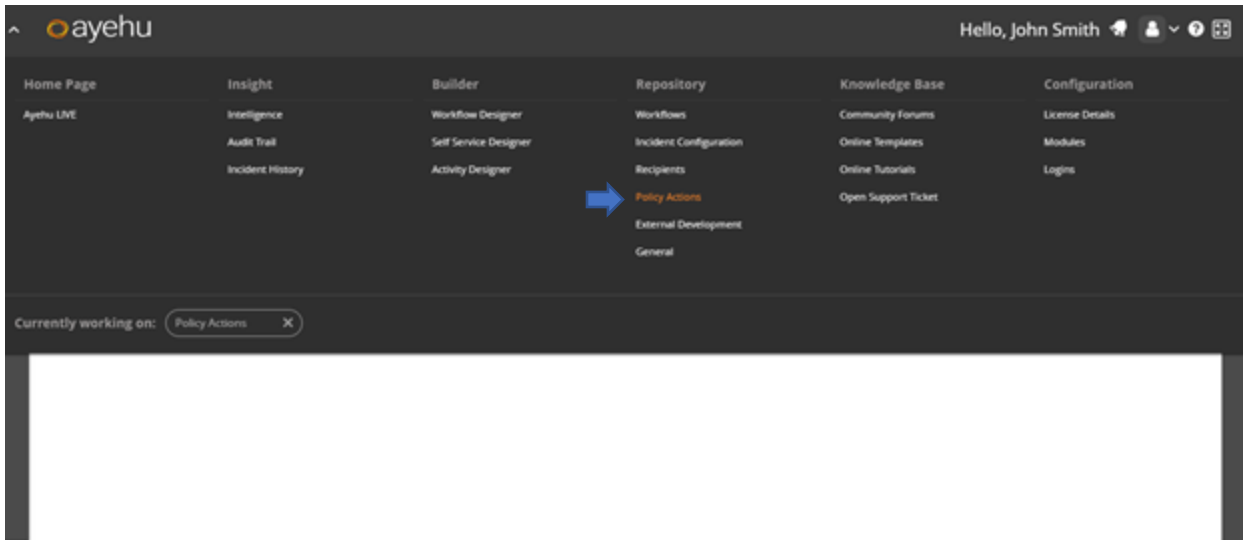


Add new condition as followed and save:



Open the policy actions and create the following,





\*\* under condition select the new condition created previously.

\*\* Under the workflow, select the imported workflow.

## How to Uninstall

*Use Windows Add/remove SW to Uninstall*