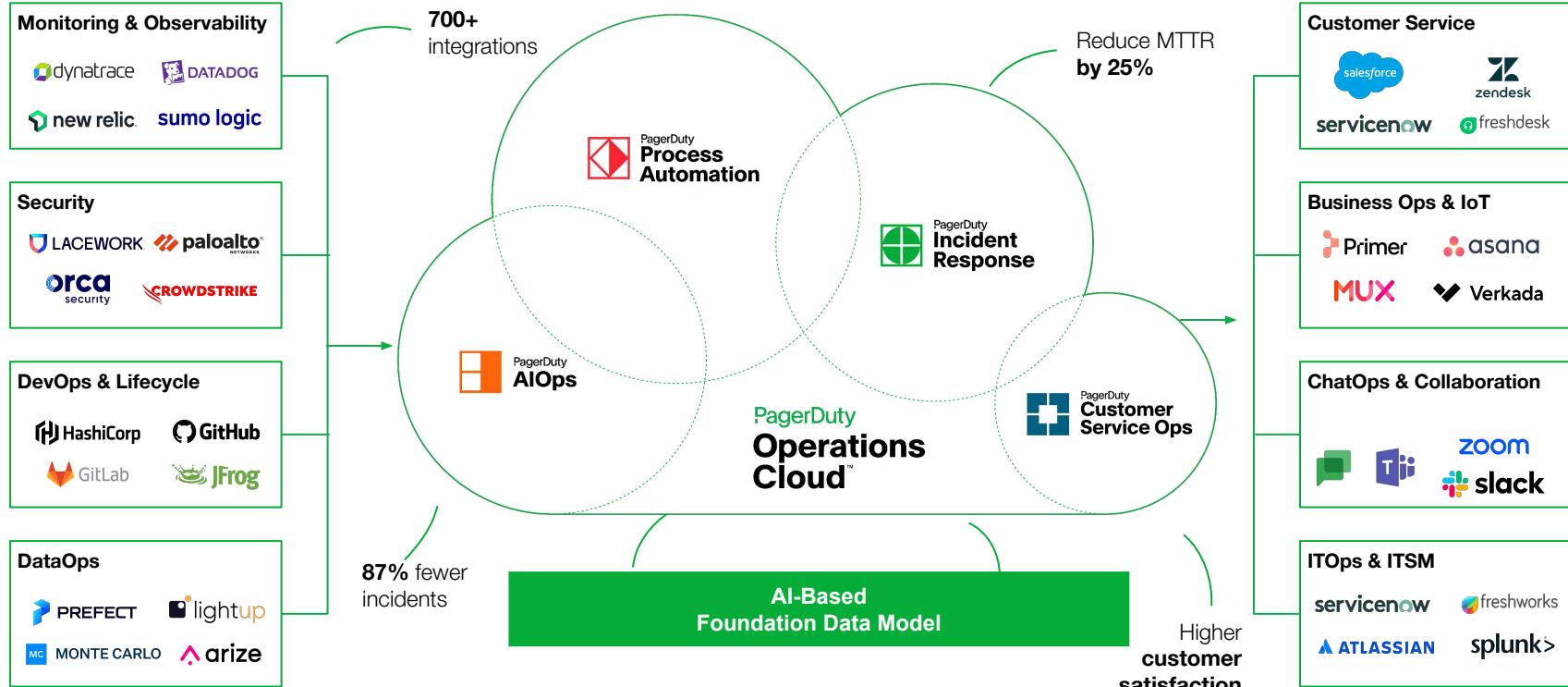


AIOps and Automation Workshop

Aug 2024, Melbourne/Sydney

The PagerDuty Operations Cloud



Have you ever heard?

PagerDuty is constantly going off!



I get notified for the same thing 10 times

10

It takes us way too long to figure out the service at fault



It takes hours to resolve a major incident



We don't have context on our incidents



We are spending too much time triggering the same action





Inundated
by alert noise

Fewer Incidents

Identify incidents that matter



Confused about
where & what

Faster Resolution

Improve situational awareness



Wasted time
on manual tasks

Greater Productivity

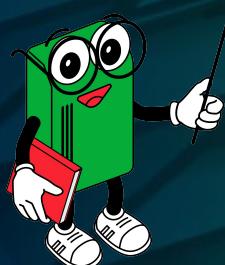
Automate manual toil

Learning Objectives

Reduce Noise

Improve context

Automate manual toil



The Power of PagerDuty AIOps

Dedup



Feedback & Learning Loop



RESOLVED!

Let's Dive In!



Set Up an Event Orchestration

1. Menu: AIOps > Event Orchestration > +New Orchestration
2. Name your orchestration
Ex: [YOURCOLOUR] Orchestration
3. Navigate to Service Routes
4. Edit Catch All Rule to direct events to **your** Technical Service

The screenshot shows the PagerDuty web interface for managing event orchestrations. The top navigation bar includes links for Incidents, Services, People, Automation, Analytics, Integrations, Status, and a search bar. Below this, a breadcrumb trail shows 'Orchestrations > Camden Orchestration > Service Routes'. The main content area is titled 'Camden Orchestration' and contains a message stating 'Global orchestration can be edited by admins or managers on any team.' A large central box has a '+ New Service Route' button. Below this, a message says 'You aren't routing these events to any services yet.' with a 'New Service Route' link. At the bottom, there's a search bar, pagination controls ('25 per page', '1 of 1'), and a modal window titled 'Edit Catch All Rule'.

Service Routes

Camden Orchestration

Global orchestration can be edited by admins or managers on any team.

+ New Service Route

You aren't routing these events to any services yet.
New Service Route

Search 25 per page 1 of 1

1. For all events, route to Unrouted Event Orchestration Unrouted Orchestration

Edit Catch All Rule

Events not matching a previous rule should not be sent to a specific service & should never create an incident.

Send all events not matching a previous rule to a service

Select a service

Checkout Function

Default Service

Send Some Example Incidents

1. Select your Global Integration Key
2. Go to
3. Paste your integration key
4. Check the box for each event
5. Execute!

The screenshot shows the 'Send Events' interface in the PagerDuty web application. On the left, under 'Events to send', there is a list of five items, each with a checked checkbox:

- A noisy Splunk event
- A noisy Datadog event
- A real Datadog problem
- A problem reported by New Relic
- A vague description

Below this list is a green 'Send' button. To the right, under 'Selected Events', are five JSON objects representing the events:

```
[{"event_action": "trigger", "payload": {"summary": "This is noise", "source": "splunk", "severity": "info"}, "routing_key": "noise"}, {"event_action": "trigger", "payload": {"summary": "Alarm noise", "source": "datadog", "severity": "warning"}, "routing_key": "noise"}, {"event_action": "trigger", "payload": {"summary": "Oh oh, problem!", "source": "datadog", "severity": "error"}, "routing_key": "problem"}, {"event_action": "trigger", "payload": {"summary": "Another Problem", "source": "newrelic", "severity": "critical"}, "routing_key": "problem"}, {"event_action": "trigger", "payload": {"summary": "Vague Description", "source": "newrelic", "severity": "critical", "component": "My Metric", "custom_details": {"usage": "bad"}, "routing_key": "vague"}]
```

Noise Reduction

How Do We Group Alerts?

Group alerts to reduce noise and improve triage response

Time-based

Groups based on specified timeframe

Easy to configure and widely available

Content-based

Groups based on an exact match of a user defined PD-CEF field

Have complete control over alert grouping

Intelligent

Groups based on machine learning looking at alert summaries, triggered time, and past responder behavior

Works out of the box and gets smarter over time

Intelligent + Content

Combine both Content-Based and Intelligent Alert Grouping with a flexible time window for maximum precision and correlation control

Specify which fields and criteria need to be met before Intelligent Alert Grouping is applied



Intelligent Alert Grouping

Auto group alerts based on ML that learns over time from patterns in inbound signals *and* responder behavior.

Stop alert fatigue from alert storms

Alerts Status Updates Timeline Past Incidents Related Incidents 1

ALERTS Group

7 triggered

Alert grouping details

FILTERS: No active

Start grouping: 1:16 PM

Stop grouping: When this incident is resolved or the valid grouping period has passed.

Automatic alert grouping is enabled for this incident

Alert grouping type: Intelligent alert grouping. Related alerts on this service will be added automatically based on the alert summary, the time of the alert, and the pattern of how alerts were grouped in the past.

Created	Service	App Liveness
at 1:17 PM		

aws:ec2:useast-1:web-app01-db01 unreachable

aws:ec2:useast-1:web-app01-db01 unreachable

aws:ec2:useast-1:web-app01-db01 unreachable

aws:ec2:useast-1:web-app01-db01 unreachable

aws:ec2:useast-1:web-app01-db01 unreachable

Code Release

1:18

Incident #11608 Alert

TRIGGERED Today, 1:18 PM Service Monitors (Checkout API Health Check violated API Request Failure)

TRIGGERED Today, 1:18 PM Service Monitors (Checkout API Health Check violated API Request Failure)

TRIGGERED Today, 1:18 PM Service Monitors (Checkout API Health Check violated API Request Failure)

TRIGGERED Today, 1:18 PM Service Monitors (Checkout API Health Check violated API Request Failure)

TRIGGERED Today, 1:18 PM Service Monitors (Checkout API Health Check violated API Request Failure)

TRIGGERED Today, 1:18 PM Service Monitors (Checkout API Health Check violated API Request Failure)

TRIGGERED Today, 1:18 PM Service Monitors (Checkout API Health Check violated API Request Failure)

TRIGGERED Today, 1:18 PM Service Monitors (Checkout API Health Check violated API Request Failure)

TRIGGERED Today, 1:18 PM Service Monitors (Checkout API Health Check violated API Request Failure)

Home Incidents My Shifts Services More



Content-Based Alert Grouping

Indicate the criteria for your alert grouping mechanism.

Group alerts based on of the following field(s):

FIELD NAME

Source

+ Add Field

Recent Alerts Examples

Summary	Created	
Device Down l2NdAw1J	Feb 9, 2022 4:44 PM PST	<input type="checkbox"/> HIDE DETAILS
{ "component": "Voyatouch", "group": "DataAccess", "severity": "critical", "source": "NewRelic", "summary": "Device Down l2NdAw1J" }		
Device Down 2dfLTzgu	Feb 8, 2022 4:44 PM PST	<input type="checkbox"/> SHOW DETAILS
Device Down llrYDbAc	Feb 7, 2022 4:44 PM PST	<input type="checkbox"/> SHOW DETAILS
Device Down O805kQlb	Feb 6, 2022 4:44 PM PST	<input type="checkbox"/> SHOW DETAILS
Device Down EiLzyHK7	Feb 5, 2022 4:44 PM PST	<input type="checkbox"/> SHOW DETAILS

Gain control in how alerts are grouped to reduce alert noise and fatigue

Specify service rules that group alerts by their fields and description



Intelligent + Content-Based Alert Grouping

Noise Reduction Reduce Noise

Reduce Noise

Combine similar alerts into a single incident to reduce notification noise and provide more context when responding to incidents. [Learn more.](#)

Services to reduce noise across:

Billing High Urgency Billing Low Urgency Payment Processing Type to search...

Alert Grouping

New alerts will be grouped across services, under the first incident that is created.

Intelligent Recommended
Events across these services will be grouped intelligently, based on historic alert patterns and past merged alerts. [Preview.](#)

Alert Content
Group alerts when they match on alert fields.

Intelligent + Alert Content
Use intelligence to group if alerts match on of these fields. [See example alerts.](#)

Match alerts based on

Time only

Alerts may be grouped if they arrive within a rolling window. [?](#)

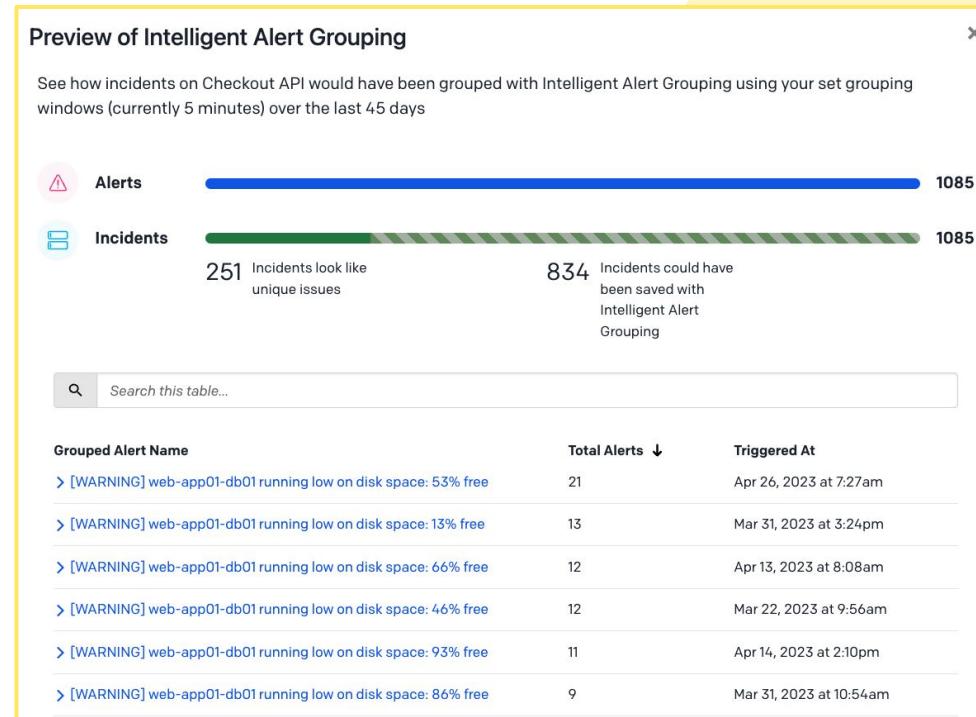
Gain control in how alerts are grouped to reduce alert noise and fatigue

Specify service rules that group alerts by their fields and description

Preview Intelligent Alert Grouping

 Preview

- View how many incidents could have been saved over 45 days
- Expand alert groups to see what would have been grouped together





Auto-Pause Incident Notifications



Transient Alerts

Pause incident creation and notification for alerts that are transient. Alerts that typically auto-resolve through integrations within minutes will be suspended for the selected duration.

- Auto-pause incident notifications** Recommended
Automatically detect transient alerts and pause notification

5 minutes

- Do not auto-pause incident notifications**

Automatically remove unnecessary noise from flapping alerts with the click of a button.

PagerDuty uses machine learning to detect and pause transient alerts that historically auto-resolve themselves so that responders can stay focused on work that matters.

Practice Problem: Reduce Noise

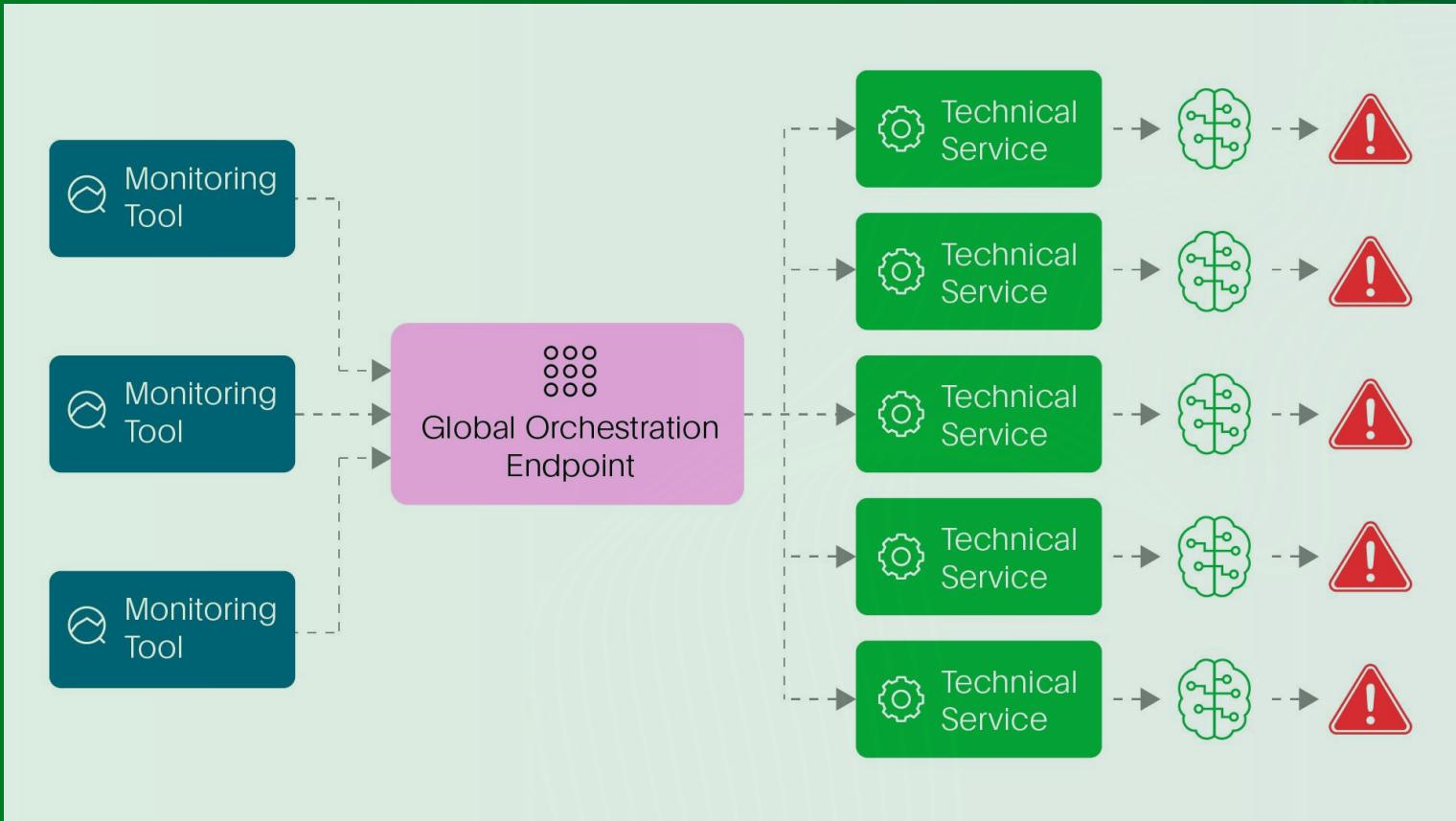
1. Resolve all open incidents
2. Navigate to Services > Service Directory
3. Select **your** Checkout Service
4. Navigate to the Settings tab
 - a. Click the Edit button in the Reduce Noise section
5. Click on Time-based alert grouping and save
6. Send at least two events from the Event Sender page

How many incidents were triggered this time?

PagerDuty

Global Event Orchestration

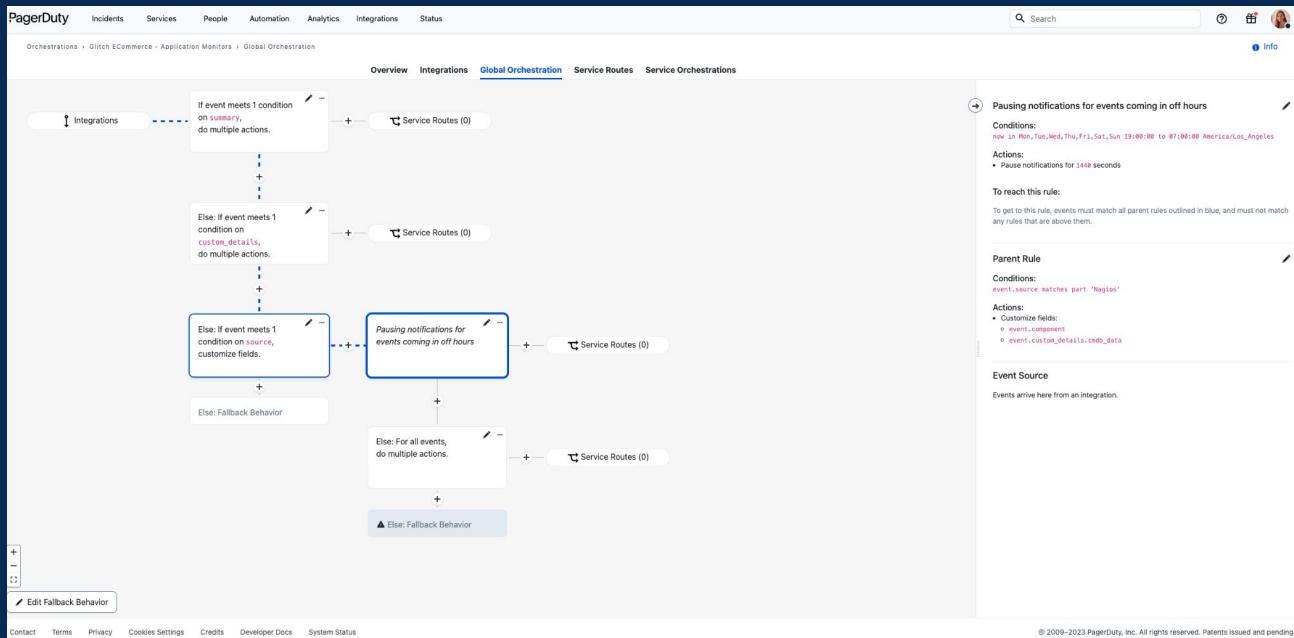
Global Event Orchestration



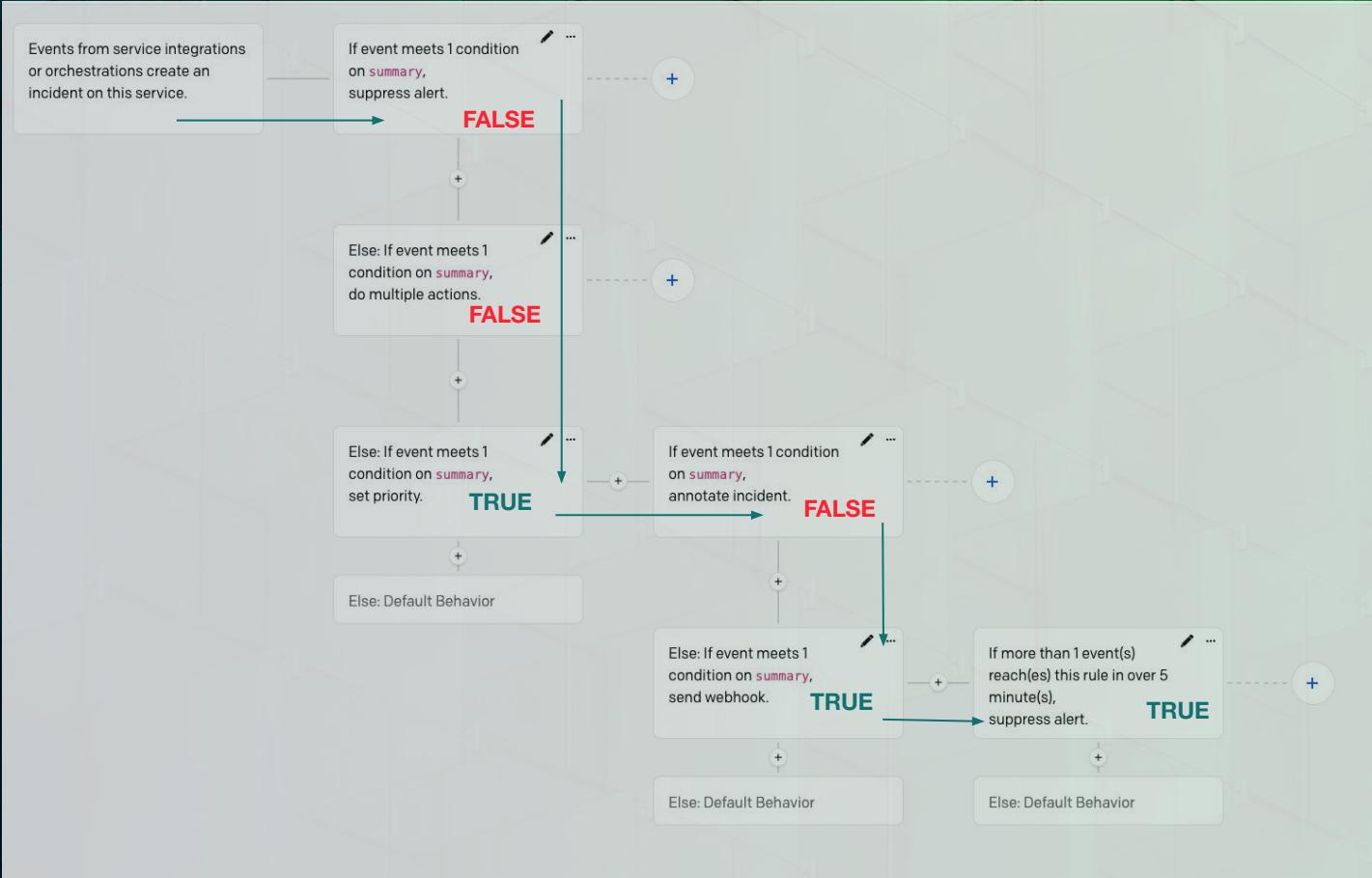
Global Event Orchestration

Normalize data and automate incident response for the entire organization

Route, enrich, and modify events on ingest to remove noise and automate processes across any or all services within PagerDuty



Reading a Service Orchestration



Building a Global Event Orchestration

Practice Problem: Reduce Noise

1. AIOps > Event Orchestration
 - a. Open Orchestration created earlier
2. Navigate to the Global Orchestration tab
 - a. Click New Rule
3. Click on the two example events with “Noise” in the titles
 - a. Click on the Summary fields
 - b. Click Next
4. Select “Pause notification” for 60 seconds and Save

The screenshot shows the 'Step 1: When should this rule be applied?' section of a new rule creation. It includes an 'IF' condition where 'event.summary' matches part '(c)' and contains 'This is noise'. An 'OR' condition is present, which also has an 'IF' clause with the same criteria. To the right, a sidebar lists recent events: 'Vague Description', 'Another Problem', 'Uh oh, problem!', and 'Alarm noise'. Below the conditions, there's an 'Override default incident creation:' section with three options: 'Suppress incident and notifications' (unchecked), 'Pause notifications. Suspend alert for [60] seconds before triggering an incident' (checked), and 'Drop incident and stop processing' (unchecked). The '60' value in the pause duration field is highlighted with a blue box.

Practice Problem: Enrich & Enhance

1. Create Else Rule
 - a. Select Always (for all events)
 - b. Click Next
2. Set Priority to P1
3. Add a note
 - a. ex: Check out our wiki for help! www.runbook.example
4. Save

Step 1: When should this rule be applied?

Always (for all events) ▼

Applies for all events that reach this rule.

+ New Condition

Step 2: What action(s) should be applied?

Incident Data >

Alert Data >

Transformation: >

Webhooks >

Override default incident creation:

Suppress incident and notifications

Pause notifications. Suspend alert for 300 seconds before triggering an incident

Drop incident and stop processing ?

Set incident priority to P1 ×

Add incident note:

Check out our wiki for help! www.runbook.example

Additional Enrichment

Step 2: What action(s) should be applied?

Incident Data >

Alert Data >

✓ Transformations >

Webhooks >

Define Custom Variables

Add Variable

Name Regex ⓘ

component

=

.*

Source ⓘ

event.component



Name Regex ⓘ

Usage

=

.*

Source ⓘ

event.custom_details.Usage



Replace Event Fields

Add Event Field

Event Field (CEF) ⓘ Value ⓘ

summary



Value ⓘ

{{variables.component}} is in a {{variables.Usage}} state

Template



Back to Conditions

Save

Practice Problem: Service Routes

1. Navigate to the Service Routes tab
2. Click on the New Service Route button
3. Route to the Checkout Service
4. Set the condition
payload.summary **does not match** the word noise
5. Click save

What service should events route to?

Checkout Function x

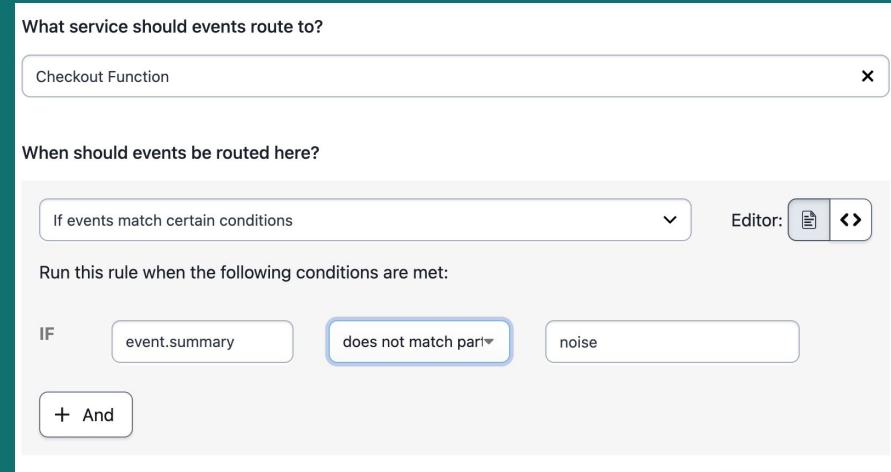
When should events be routed here?

If events match certain conditions v Editor: File Copy Link

Run this rule when the following conditions are met:

IF event.summary does not match part noise

+ And



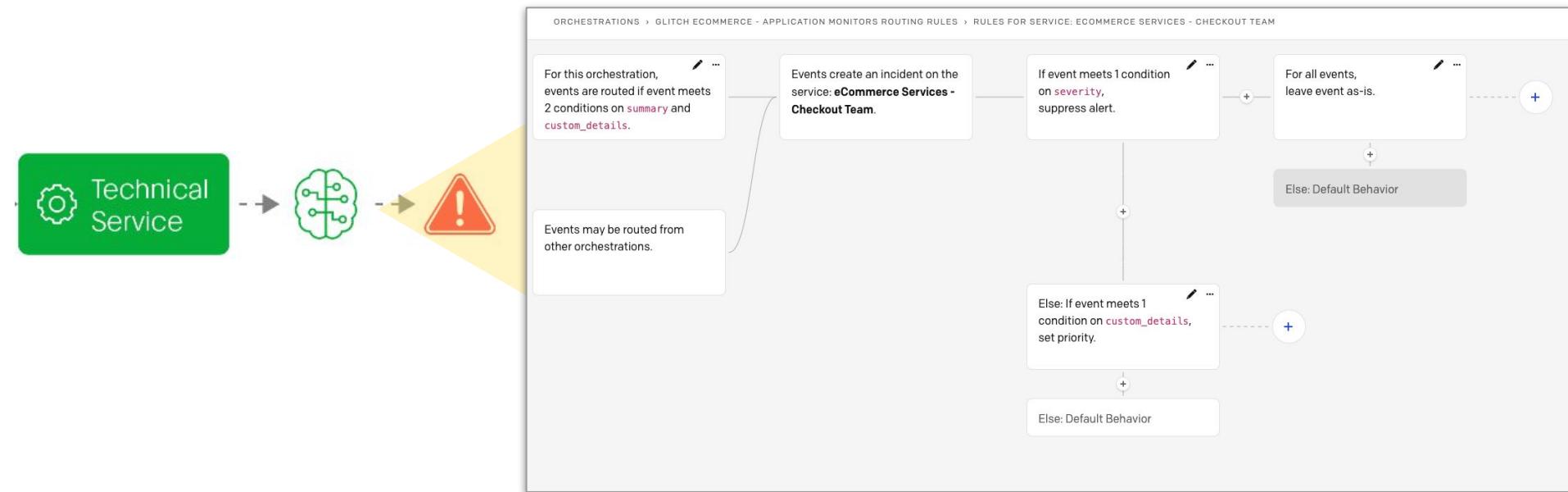
Send in events!

- View updated priority and summary
- Check out notes
- What about those noisy alerts?



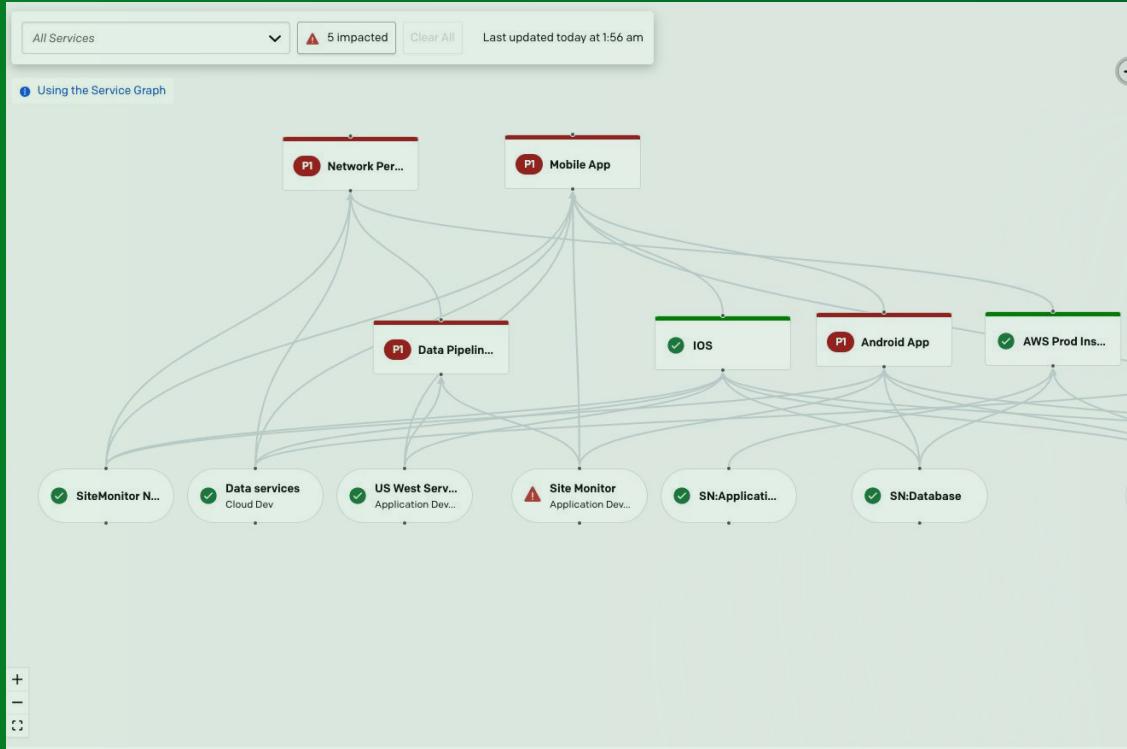
Think about it: How might you edit your Checkout Service further?

Service Orchestration Rules



Root Cause Analysis

Service Graph



- ➊ Understand how technical and business services work together at a glance
- ➋ See the full blast radius of an issue
- ➌ Zero-in on probable cause

Outlier Incidents

Using historical incident data identify incident types that are anomalies, rare occurrences or frequent offenders.

- Provide context on frequency of certain types of incidents
- Classification of incidents as anomalies, rare or frequent events
- Analytics on frequency of occurrence

The image displays three separate screenshots of the PagerDuty web application interface, each showing a different type of outlier incident:

- Screenshot 1:** Shows an incident titled "INCIDENTS > INCIDENT #208". The summary states "Requests to Checkout API are failing in prod" and includes an orange "Anomaly" badge. A tooltip for "Anomaly" indicates it's "Not similar to any incidents on this service in the preceding 30 days."
- Screenshot 2:** Shows an incident titled "INCIDENTS > INCIDENT #207". The summary states "Checkout API timing out" and includes an orange "Anomaly" badge. A tooltip for "Anomaly" indicates it's "similar to 50% of incidents on this service in the preceding 30 days." It also shows a detailed "Incident Outliers" section with frequency details: "Frequent: this incident type occurs greater than 20% of the time on this service." and "Rare: this incident type occurs less than 5% of the time on this service."
- Screenshot 3:** Shows an incident titled "INCIDENTS > INCIDENT #166958". The summary states "Request Response Time is High for prod" and includes an orange "Anomaly" badge. A tooltip for "Anomaly" indicates it's "Not similar to any incidents on this service in the preceding 30 days."



Past Incidents

- Accelerate triage through past response data from similar incidents
- Find out who resolved it and how it was resolved

Screenshot of the PagerDuty interface showing a past incident summary and historical data.

Incident Summary:

- Title:** Promotions API: Request Response Time is High for prod - (95th percentile > 100 ms on average during the last 10m)
- Status:** Triggered
- Priority:** P2
- Urgency:** High
- Opened:** Apr 15, 2020 at 12:08 PM (a day ago)
- Assigned To:** Greg Delgado
- Escalation Policy:** Promotions On-Call
- Responders:** 1 (Greg Delgado)
- Slack Channel:** Set Channel
- Impacted Service:** New Coupon Banner
- Service:** pulls from our partner API that surfaces new deals, coupons, and other specials
- Description:** 21 alerts. Last alert added to this incident on Apr 15, 2020 at 12:09 PM.

Historical Data (Median Incident Duration: 5h 52m):

Time Period	Number of Past Incidents
Last 6 months	311
Last 7 days	8

Timeline Heatmap:

Top 5 Past Incidents:

Title	Duration	Created	Last changed by
Promotions API: Request Response Time is High for prod - (95th percentile > 100 ms on average during the last 10m)	18m 25s	on Apr 15, 2020 at 3:34 PM	Sarah Hoffman
Promotions API: Request Response Time is High for prod - (95th percentile > 100 ms on average during the last 10m)	1d 2h	on Apr 15, 2020 at 7:51 AM	...
Promotions API: Request Response Time is High for prod - (95th percentile > 100 ms on average during the last 10m)	16h 58m	on Apr 14, 2020 at 2:51 PM	Sarah Hoffman
Promotions API: Request Response Time is High for prod - (95th percentile > 100 ms on average during the last 10m)	5h 27m	on Apr 14, 2020 at 8:38 AM	Sarah Hoffman
Promotions API: Request Response Time is High for prod - (95th percentile > 100 ms on average during the last 10m)	5h 52m	on Apr 14, 2020 at 8:13 AM	Sarah Hoffman

Resources:

- Incident Lifecycle
- Mobilizing multiple responders
- PagerDuty Common Event Format
- Past Incidents
- Related Incidents

Related Incidents

View real time contextual insights across multiple services and teams

- See what is happening right now across the business
- Understand if issue is local or impacts others
- Find the right people and work together to fix the problem

The screenshot displays the PagerDuty interface for managing related incidents. At the top, there are tabs for Alerts, Status Updates, Timeline, Past Incidents, and Related Incidents (which is currently selected). Below the tabs, a section titled "Impact Summary" provides an overview of current responders (3), business services (1), and technical services (3) related to the incident.

Two specific incidents are highlighted:

- Incident #97999 (Triggered at 9:57 AM Jun 26):** A Splunk alert regarding MySQL connection errors. It shows 1 alert, assigned to Mary McKenna, and the service is the Inventory API. The dependency notes that the Inventory Database is used by the Inventory API, which in turn is used by the Business Service Inventory.
- Incident #97998 (Triggered at 9:56 AM Jun 26):** An alert about high request response times for the Inventory API. It shows 20 alerts, assigned to Cheryl McLaughlin, and the service is the Inventory API. The dependency notes that the Inventory Database is used by the Inventory API, which in turn is used by the Business Service Inventory.

At the bottom of each incident card, there is a "Help: Is this related to incident #97997?" button with thumbs up and thumbs down icons, and a count of 0 for both.

To the right, a smartphone screen shows the same "Related Incidents" view, indicating the mobile compatibility of the feature.

Probable Origin

- Jumpstart triage efforts with an auto-generated list of likely origin points for faster resolution.
- Use historical data of correlated incident patterns to surface where (and where not) to look first when troubleshooting major incidents.

Probable Origins i

BETA

Incident: [#123456] AWS Health Event:
us-east-1 EBS :
AWS_EBS_VOLUME_LOST

100% likely origin

Service: AWS Infrastructure

Status: Resolved at 5:13 AM (4 hours ago)

Incident: [#123457] AWS Health Event: us-east-1 AWS_EC2_INSTANCE

100% likely origin

P2

Service: AWS Infrastructure

Status: Resolved at 7:18 AM (2 hours ago)

How often do changes lead to incidents?

Change Correlation

Surface the changes that most likely caused the incident based on time, related service, or machine learning analysis of similar incidents

Immediately eliminate irrelevant changes and pinpoint the potential contributing factor

Recent Changes

- Modified payments table to support multiple exchanges**
Payments UI
October 12 at 2:29PM
- This incident occurred 10 minutes after this change.
- Updated design system dependencies**
design-system-components
October 11 at 1:27PM
- ⚠ Similar incidents have previously occurred with changes like this
- Changed primary button CTAs**
Payments UI
October 11 at 1:22PM
- ⚡ This change occurred on a dependency of Payments UI



Change Events

The screenshot shows the PagerDuty Service Directory interface. At the top, there are tabs for Services, Recent Changes (which is selected), and Maintenance Windows. Below this, there are filters for SERVICE (Shopping Cart App Server), SOURCE (Any source), and LAST CHANGE (Last hour). A button for '+ Add New Event Stream' is also present. The main area is titled 'Summary' and lists recent changes for the 'Shopping Cart App Server'. Each entry includes a status icon, the change type (miketoh Merge pull request #79 from PagerDuty/DUE-1445 scheduler-ud), the service name, the source (GitHub), the creator (miketoh), and the creation time (Today at 10:04AM). Each entry also has a 'View on GitHub' link.

Summary	Service	Source	Created by	Created
miketoh Merge pull request #79 from PagerDuty/DUE-1445 scheduler-ud	Shopping Cart App Server	GitHub	miketoh	Today at 10:04AM View on GitHub
miketoh Merge pull request #79 from PagerDuty/DUE-1445 scheduler-ud	Shopping Cart App Server	GitHub	miketoh	Today at 10:04AM View on GitHub
miketoh Merge pull request #79 from PagerDuty/DUE-1445 scheduler-ud	Shopping Cart App Server	Buildkite	miketoh	Today at 10:04AM View on Buildkite
miketoh Merge pull request #79 from PagerDuty/DUE-1445 scheduler-ud	Shopping Cart App Server	GitHub	miketoh	Today at 10:04AM View on GitHub
miketoh Merge pull request #79 from PagerDuty/DUE-1445 scheduler-ud	Shopping Cart App Server	Buildkite	miketoh	Today at 10:04AM View on Buildkite
miketoh Merge pull request #79 from PagerDuty/DUE-1445 scheduler-ud	Shopping Cart App Server	Buildkite	miketoh	Today at 10:04AM View on Buildkite
miketoh Merge pull request #79 from PagerDuty/DUE-1445 scheduler-ud	Shopping Cart App Server	GitHub	miketoh	Today at 10:04AM View on GitHub
miketoh Merge pull request #79 from PagerDuty/DUE-1445 scheduler-ud	Shopping Cart App Server	Buildkite	miketoh	Today at 10:04AM View on Buildkite
miketoh Merge pull request #79 from PagerDuty/DUE-1445 scheduler-ud	Shopping Cart App Server	GitHub	miketoh	Today at 10:04AM View on GitHub
miketoh Merge pull request #79 from PagerDuty/DUE-1445 scheduler-ud	Shopping Cart App Server	Buildkite	miketoh	Today at 10:04AM View on Buildkite

Know exactly what changes occurred and whether it could be the cause of an incident

Prevent a problem from getting worse and coordinate an effective response with contextual information about recent changes

**Automation - Now we
know there's a problem. What can
we do?**

Event Driven Automation

Virtual Responder or Responder assisted

Automated diagnostics & resolution



Runbook Automation

Access to knowledge and tools, agnostic for scripting or Infrastructure as Code.

Allow developers to run code and democratize automation for larger groups.



Workflow Automation

Simplicity in mind

Built for Low Code/No Code use cases with human in the loop.

Enterprise citizen developers

Generative AI/Machine Learning

Noise Reduction, with human reinforcement

PagerDuty CoPilot

Enhance & Simplify Context

Describe and build Runbooks

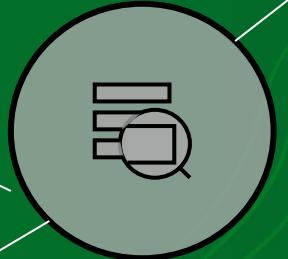
PagerDuty

Event Driven Response & Diagnostics

Enrich existing events with relevant data
Time, date, status & logs
Platform status
Service status
3rd party status
Kubernetes status
Restart Servers
Restart Services
DB Unlocks
Flush Storages
Clearing Files/Memory
Adding more Disk or Memory Space
Open/Update/Close Tickets
Initiate DR Failover
Healing
Escalation

Service Request Automation

Infrastructure Provisioning
Auto shut down unused multi cloud resources
Onboarding/ Deleting Users
Decommission Hardware
Adding Servers
Adding Storage
Software Deployment
Software Updates
De/Provisioning AWS Services
Opening Ports, Switching/Routing
Production Patching
Vulnerability Patching
Increasing Capacity
Security Settings
Validate Security
Change Configs
Adding VLANs
Creating Slack channels
Adding DNS hosts
Firewall port settings
SSL Certificates validation checks
Get next available IP from DDI
White list/ blacklisting IP/ domains
Dbase creation inside a SQL instance
Benchmark testing
...
....(creativity)



Data Distribution

Task / Job Scheduling
ETL (Extract-Transform-Load)
File Transfers
Mass data Removal
GDPR data removal
Complex Workflow / Rules
Big Data Replication
Data Remodeling
Dbase jobs, restart
Export-Import mass data

General Use Case Examples

PagerDuty

Event Driven Response & Diagnostics

- Getting Pod Status & description
- Getting Cluster Status
- Running Processes & Limits
- Get & Parse events
- Tail container logs
- Execute in-container commands
- Run Kubernetes Jobs
- Run Ephemeral containers, with additional tooling to capture debug state
- Check other status (eg cron jobs)
- Rollout history



Service Request Automation

- Self Service for Kubernetes application deployments
- Job Trigger delegation with Guardrails
- Interaction with ITSM Tools
- Manage cloud services, AWS, Azure, GCP
- Wrap existing developer tools
- Trigger existing pipelines
- Trigger 3rd party APIs
- Increasing Resource limits for deployments
- Triggering deployments
- Triggering external components (eg Jenkins/Git/Azure Pipelines)
- Pod Deletion
- Scale deployment to zero, then back up to original
- Undo deployments
- Execute script inside container
- Execute K8s jobs

PagerDuty

Event driven Response

- Revoke access (eg AD, Okta etc)
- Rate Limit API (eg Network Gateway)
- Disable Network Interface
- Block specific IP Address (IP Tables eg)
- Shutdown a service
- Disconnect a network share
- Stop Running Processes
- Disable a user account
- Change network configuration via terraform
- Update CDN Settings
- Update DNS Settings
- Run additional direct SSH commands to appliance
- Suspend an account
- Collect logs
- Snapshot Environments
- Invoke devops pipeline to rebuild
- Apply Patches
- Reset sessions

Service Request Automation

- User account creation/deletion.
- Password reset requests.
- Deploying software updates.
- Security patch application.
- Log file parsing/analysis.
- Password complexity checks.
- User account lockouts.
- Disk encryption tasks.
- Firewall rule additions.
- VPN connection testing.
- Security alert notifications.



Network Examples

PagerDuty

Self Service/Scheduled DBA Orchestration

- Backup/Verify
- Apply Patches
- Check/Extend Disk space
- Create/Modify/Delete User Accounts
- Analyze/Optimise Queries
- Execute Data Integrity Scripts
- Manage DB Log Rotation
- DB Switch over for DR
- Removal or correction of outdated, duplicated, or incorrect records.

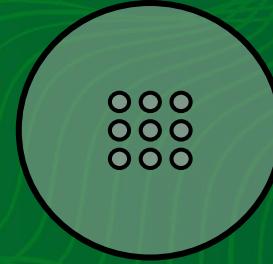
Moving Transaction Logs

Automate the application of schema changes across multiple environments.

Automate the process of moving data from one database or format to another.

Distribute Reports

Package SQL Queries for less technical teams



Exercise 1

First Job

Objective

Utilise Runbook automation to run a basic command

Save that command as a reusable job that could be triggered by:

- Colleagues
- An event
- A schedule
- A webhook
- An API call from a website

Scenario

- You are on call
- Its 3am. Your phone pings.
- You are notified by PagerDuty of a service issue.
“Cannot Run ServiceAPP - Insufficient space on volume”,
And its having a customer impact...

You open your Runbook to see what happens with this kind of failure.....



Scenario

Responder Runbook

Operator Guidance for error...

1. This error can occur when the directory fills with small temporary files.

Instructions

Log onto the machine and run the following command to assess filesystem.

```
find /home/ec2-user/temp -type f -size -100k
```

Conclusion

Your very first job was created by running a single command

- You used a command in your runbook
- You ran it interactively
- You saved it as a reusable job that could be triggered by
 - An event such as an Incident
 - Manually from within PagerDuty
 - From an ITSM tool like ServiceNow or Jira
 - From the self-serve UI
 - From a Webhook from another application
 - From a schedule

Multiple end point commands,

Exercise 2

Automation

Actions

Scenario

- You are on call (again)
- Its 3am. Your phone pings.
- You are notified by PagerDuty of a service issue.
“Machine Resource issue on environment. Customer Experience Impacted”

You open your Runbook to see what happens with this kind of failure.....



Scenario

Responder Runbook

Operator Guidance for error...

1. You will need to run 3 different sets of diagnostics to figure out what's Happening.

Log onto the machine with your credentials

- CPU : `ps -eo pcpu,pid,cmd | sort -k 1 -nr | head -5`
- Memory: `ps -eo pmem,pid,cmd | sort -k 1 -nr | head -5`
- Disk: `sudo du -Shx / | sort -rh | head -10`

Objective

Utilise a pre-built Automation action to trigger both manually and automatically from a PagerDuty Event

- Set up your first Automation Action in PagerDuty
- Gather multiple diagnostics information in Incident notes
- Trigger incident and manually run diagnostics process

Actions

Do this:

1. Log into PagerDuty Incident Management using your credentials
2. Select the Automation Tab, Automation Actions
3. Click on Add Action
4. Select the Workshop RBA connection to pick up the jobs for the server

Screenshot/Notes

Sign in to workshop.eu.pagerduty.com

The screenshot shows the PagerDuty Incident Management interface. At the top, there's a sign-in form with fields for Email (containing "user-pink@pagerduty.com") and Password (containing a masked value). Below the sign-in is a "Remember me" checkbox and a "Forgot Password?" link, followed by a blue "Sign In" button.

The main area is titled "Automation Actions". It has tabs for "DETECT" (Event Orchestration) and "REMEDIATE & DIAGNOSE" (Automation Actions). The "Automation Actions" tab is selected. A sub-section titled "Event Orchestration" is visible. Below this, a note states: "An Automation Action invokes jobs and workflows that are staged in Runbook Automation. It may also execute a command line script run by a Process Automation infrastructure." There are two tabs at the top of this section: "Actions" (selected) and "Runners".

The "Actions" tab shows a search bar with placeholder "Search by action name" and dropdown menus for "SERVICES" (set to "Any Service") and "TEAMS" (set to "All Teams"). Below the search bar, a message says "There's nothing here yet". A note below it says: "Create an action which will be available to run from an incident. An action can be any type of script, which could trigger a remediation or diagnostic action." A blue "+ Add Action" button is present.

To the right of the main content, there are two numbered steps: 1. Select Runner and 2. Select Job. Below this, a section titled "Create an Automation Action" is shown with the note: "An Automation Action invokes jobs and workflows that are staged in Runbook Automation. It may also execute a command line script run by a Process Automation runner." A "Select a runner" section follows, with a note: "Pick a runner to execute the action. If you don't have a runner, click here to add one." A search bar for "Search by runner name" is shown, with a result "Runners" listed. Under "Runners", a radio button is selected for "Workshop RBA Runner" (Automation Workshop Runner (created via TF)).

Actions

4. Select Job to populate Job list
5. Use Command + F (Shift F for windows) in your web browser find your project name
6. Browse through until you find this Job for your Project. Eg user-red

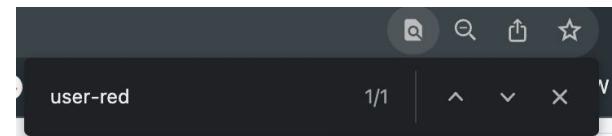
The job is called

2. Diagnostics - Top CPU & Memory Processes and Top Disk Consuming Files

Select this job by checking the option box.

Screenshot/Notes

Select Job



- ② [2. Diagnostics - Top CPU & Memory Processes and Top Disk Consuming Files](#)
Diagnostics of a Linux Node's CPU, Memory & Disk to a PagerDuty incident.

user-red_pagerduty.com

Actions

7. Keep the defaults but select the Service to match your service eg `service-red`

(This is where you can scope an action to multiple services and teams)

8. Add your Team eg `team-red`

8. Select Next

Screenshot/Notes

Name and Description

Action Name*

2. Diagnostics - Top CPU & Memory Processes and Top Disk Consuming Files

Responders use the name and description to select the right action during an incident.

Action Description*

Diagnostics of a Linux Node's CPU, Memory & Disk to a PagerDuty incident.

Help responders understand what the action does and when to use it. Explain any risks.

Select category

Any Category

Services

Associate this action with service(s) to make the action available to run on the services' incidents.

service-pink ✖ Find service(s)

Teams

Associate with a team to limit who has access to run this action.

Find team(s)

Next

Actions

We can add any data from the event we need to pass over in the argument
(eg Server name, Namespace, Container Name). For this we will just use the Incident ID.

9. Copy Paste this into the argument field

-pd_incident_id \${pd.incident.id}

10. Hit Create Action to finish

Congratulations you have created your first Automation Action

Screenshot/Notes

NAME

2. Diagnostics - Top CPU & Memory Processes and Top Disk Consuming Files

DESCRIPTION

Diagnostics of a Linux Node's CPU, Memory & Disk to a PagerDuty incident.

SERVICES

service-pink

Define your action

Enter arguments (optional)

-pd_incident_id \${pd.incident.id}

At runtime, PagerDuty context variables will be replaced with context data. [Learn more](#).

Enter node filter (optional)

The basic format is a sequence of 'attributename: value' pairs. [Learn more about node filter syntax](#). [Learn more](#).

[Create Action](#)

JOB

2. Diagnostics - Top CPU & Memory Processes and Top Disk Consuming Files
Diagnostics of a Linux Node's CPU, Memory & Disk to a PagerDuty incident.

PROJECT
user-pink_pager_duty_com

ARGUMENTS

-pd_incident_id \${pd.incident.id}

ⓘ At runtime, PagerDuty context variables are replaced with context data.
[Learn more](#).

Actions

Let's trigger an incident to test the Automation Action:

1. Select your Global Integration Key
2. Go to
3. Paste your integration key
4. Check the box for an event
5. Send!

Screenshot/Notes

The screenshot shows a user interface for sending events. At the top, there's a green header bar with a frog icon and the text "Send Events". Below this, on the left, is a form with a "Routing Key" input field containing "R0217SL0UK3FTYUXGOLXKG0Z51". On the right, under "Selected Events", are five green rounded boxes, each containing a JSON object representing an event payload. The events are:

- A noisy Splunk event:{"event_action": "trigger", "payload": {"summary": "This is noise", "source": "Splunk", "severity": "info"}}
- A noisy Datadog event:{"event_action": "trigger", "payload": {"summary": "Alarm noise", "source": "Datadog", "severity": "warning"}}
- A real Datadog problem:{"event_action": "trigger", "payload": {"summary": "Uh oh, problem!", "source": "Datadog", "severity": "error"}}
- A problem reported by New Relic:{"event_action": "trigger", "payload": {"summary": "Another Problem", "source": "NewRelic", "severity": "critical"}}
- A vague description:{"event_action": "trigger", "payload": {"summary": "Vague Description", "source": "NewRelic", "severity": "critical", "component": "CPU metric", "custom_details": {"Usage": "bad"}}

Below the routing key input is a section titled "Events to send" with five checkboxes, all of which are checked:

- A noisy Splunk event
- A noisy Datadog event
- A real Datadog problem
- A problem reported by New Relic
- A vague description

At the bottom left is a green "Send" button.

Actions

5. Back to Pagerduty Incident Management

6. Open the incident on your service

7. Select Run Actions, and run the Diagnostics Job

Screenshot/Notes

Open Incidents (1)

<input type="checkbox"/> Status	Priority ▾	Urgency	Alerts	Title
<input type="checkbox"/> Triggered		High	1	Sample Error for Automated Diagnostics Test #39 + SHOW DETAILS (1 triggered alert)

Run Actions ▾ Send Status Update More ▾

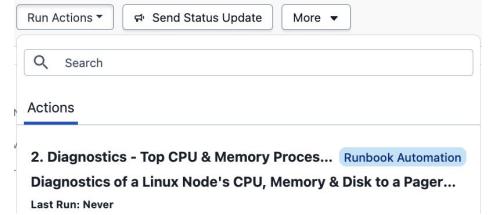
Search

Actions

2. Diagnostics - Top CPU & Memory Processes... Runbook Automation

Diagnostics of a Linux Node's CPU, Memory & Disk to a PagerDuty incident.

Last Run: Never



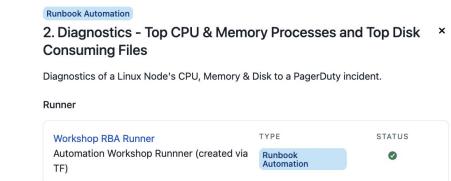
Runbook Automation

2. Diagnostics - Top CPU & Memory Processes and Top Disk Consuming Files

Diagnostics of a Linux Node's CPU, Memory & Disk to a PagerDuty incident.

Runner

Workshop RBA Runner	TYPE	STATUS
Automation Workshop Runner (created via TFI)	Runbook Automation	Green



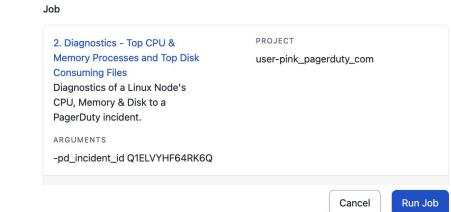
Job

2. Diagnostics - Top CPU & Memory Processes and Top Disk Consuming Files	PROJECT
Diagnostics of a Linux Node's CPU, Memory & Disk to a PagerDuty incident.	user-pinkk_pagerduty_com

ARGUMENTS

-pd_incident_id Q1ELVYHF64RK6Q

Cancel Run Job



Actions

Shortly you should be able to see the three diagnostics jobs in the notes and the timeline.

(These are also available in Slack/Teams or the mobile app)

9. Click the link, for the detailed diagnostics in the Runbook Automation UI.

10. **Resolve** your incident for our next test

Time permitting, edit the job to investigate the 3 different steps to gather machine level diagnostics.

Screenshot/Notes

Oct 7, 2023

A

Top CPU Processes:
%CPU PID CMD
0.0 1279484 head -5
0.0 1279483 sort -k 1 -nr
0.0 1279482 ps -eo pcpu,pid,cmd
0.0 1279466 bash -c ps -eo pcpu,pid,cmd | sort -k 1 -nr | head -5

Top Memory Processes:
0.4 2010 /usr/libexec/sssd/sss_nss --uid 0 --gid 0 --logger=files
0.2 2304 /usr/bin/ssm-agent-worker
0.2 1121 /usr/lib/systemd/systemd-journald
0.2 1 /usr/lib/systemd/systemd --switched-root --system --
deserializer 32
0.1 1279485 sshd: ec2-user [priv]

Top Disk Consuming Files:
240M /var/log/journal/e2b74b27c00f4a4591046759d7d01533
214M /usr/lib/locale
156M /usr/bin
80M /usr/lib64
29M /usr/lib/python3.9/site-packages/babel/locale-data
27M /var/lib/sss/mc
27M /boot
25M /usr/sbin
23M /var/cache/dnf
18M /var/cache/dnf/amazonlinux-635fbbe8d2f21e74/repo

Click here for detailed diagnostics:

https://pduniversity.runbook.pagerduty.cloud/project/user-pink_pagerduty_com/execution/show/4991#output

Automation 5:22pm

Actions

Clearly these can be more complex, and split across different environments.

Here's an example of Kubernetes, Database and Availability diagnostics.

We have a solution pack containing many modern diagnostics jobs [here](#)

Screenshot/Notes

The screenshot displays a timeline of monitoring notes and system status across several panels:

- Deployment Status:** Shows a deployment named "travelduty-frontend" with 1 revision, no change cause, and 1 pod status. All pods are running and ready.
- AWS RDS Status:** Notes for "rds-us-east-1" indicate the service is operating normally with increased error rates.
- Endpoint Tests:** Successes for "http://www.cloudflare.com" and "https://www.stripe.com".
- Database Log:** Notes for "Database Log" with "Last Errors in Log".
- Core Platform Diagnostics:** Notes for "Core Platform Diagnostics" with "System Information".

Conclusion

Utilise a pre-built Automation action to trigger both manually and automatically from a PagerDuty Event

- Set up your first Automation Action in PagerDuty
- Gather multiple diagnostics information in Incident notes
- Trigger incident and manually run diagnostics process

Exercise 3

Automate

The

Trigger

Objective

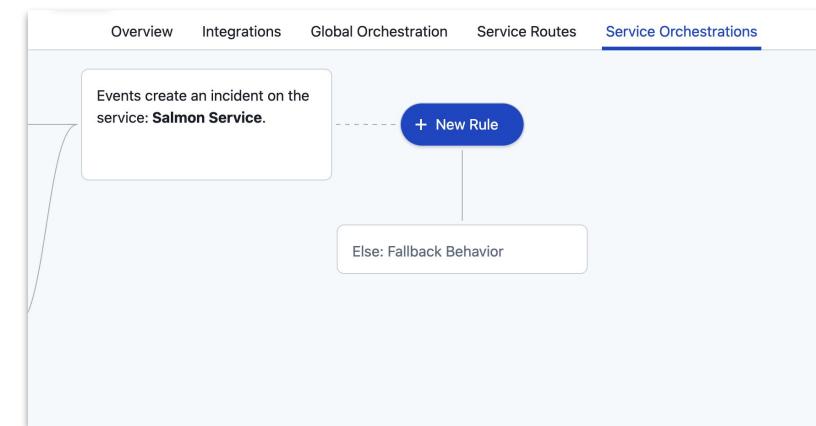
Manual actions are great for real time,
user-driven response but for diagnostics, automatic is a great solution

- Orchestrate our service to execute the Action each time an incident triggers

Actions

1. Log back into PagerDuty Incident Management
2. Go to your Event Orchestration and navigate to Service Orchestrations
3. Select your service and create a New Rule
4. Leave the condition as *Always (for all events)*

Screenshot/Notes



New Event Rule

Click to add a description for this rule

Step 1: When should this rule be applied?

Always (for all events)

Applies for all events that reach this rule.

+ New Condition

Examples of recent events:

Sample Error for Automated Diagnostics Test

```
{  
  "class": "disk",  
  "component": "mysql",  
  "custom_details": {  
    "free space": "1%",  
    "load avg": 0.75,  
    "ping time": "1500ms"  
  },  
  "dedup_key": "srv01/mysql",  
  "event_action": "trigger",  
  "group": "prod-datapipe",  
  "severity": "critical",  
  "source": "prod-datapipe03.example.com",  
  "summary": "Sample Error for Automated Diagnos",  
  "timestamp": "2015-07-17T08:42:58.315+0000"  
}
```

Next →

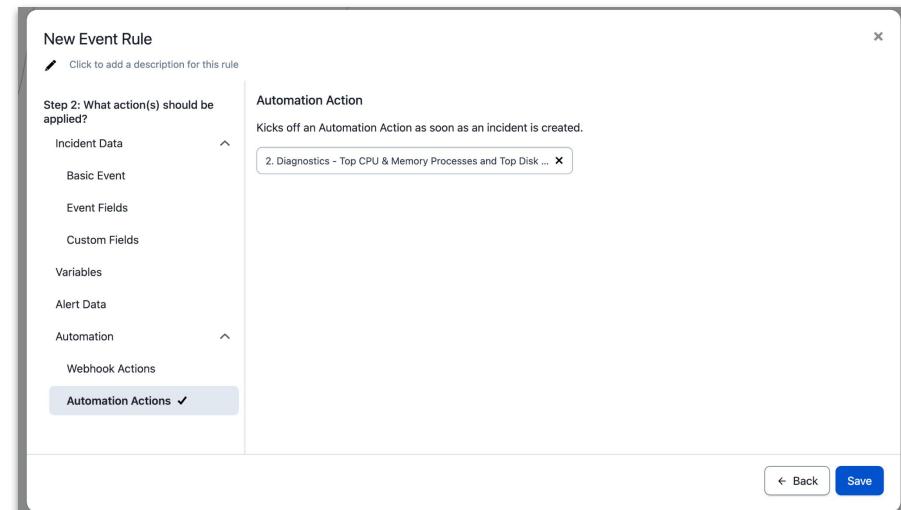
Actions

5. Select Automation > Automation Action

And select our previously created Action

6. Select Save to complete.

Screenshot/Notes



Actions

Let's trigger another incident:

5. Select your Global Integration Key

6. Go to

7. Paste your integration key

8. Check the box for an event

9. Send!

Screenshot/Notes

The screenshot shows a user interface for sending events. At the top, there is a green header bar with a frog icon and the text "Send Events". Below the header, there are two sections: "Routing Key" and "Selected Events".

The "Routing Key" section contains a text input field with the value "R0217SL0UK3FTYUXGOLXKG0Z51".

The "Selected Events" section displays five event payloads, each enclosed in a light green rounded rectangle. The events are:

- {
"event_action": "trigger",
"payload": {
"summary": "This is noise",
"source": "Splunk",
"severity": "info"
}
}
- {
"event_action": "trigger",
"payload": {
"summary": "Alarm noise",
"source": "Datadog",
"severity": "warning"
}
}
- {
"event_action": "trigger",
"payload": {
"summary": "Uh oh, problem!",
"source": "Datadog",
"severity": "error"
}
}
- {
"event_action": "trigger",
"payload": {
"summary": "Another Problem",
"source": "NewRelic",
"severity": "critical"
}
}
- {
"event_action": "trigger",
"payload": {
"summary": "Vague Description",
"source": "NewRelic",
"severity": "critical",
"component": "CPU metric",
"custom_details": {
"Usage": "bad"
}
}
}

Below these sections is a "Events to send" section containing five checkboxes, all of which are checked:

- A noisy Splunk event
- A noisy Datadog event
- A real Datadog problem
- A problem reported by New Relic
- A vague description

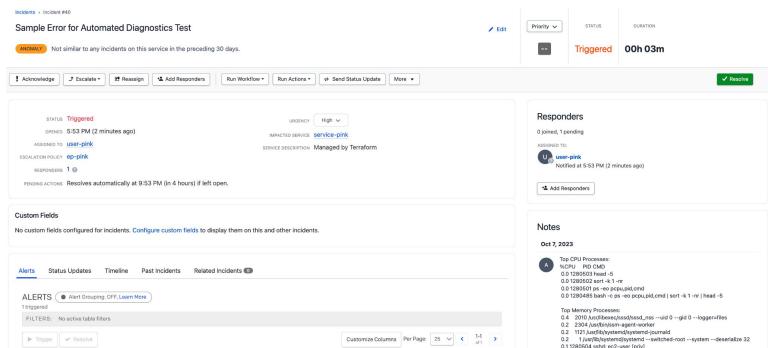
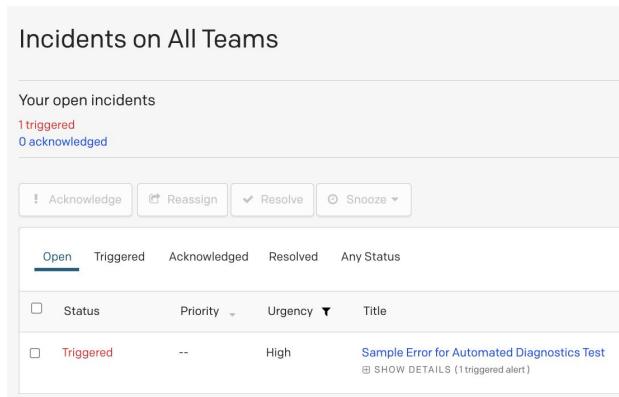
At the bottom left is a green "Send" button with a white envelope icon.

Actions

10. Browse to incidents, Select Incident

11. Diagnostics should be triggered every time the incident fires

Screenshot/Notes



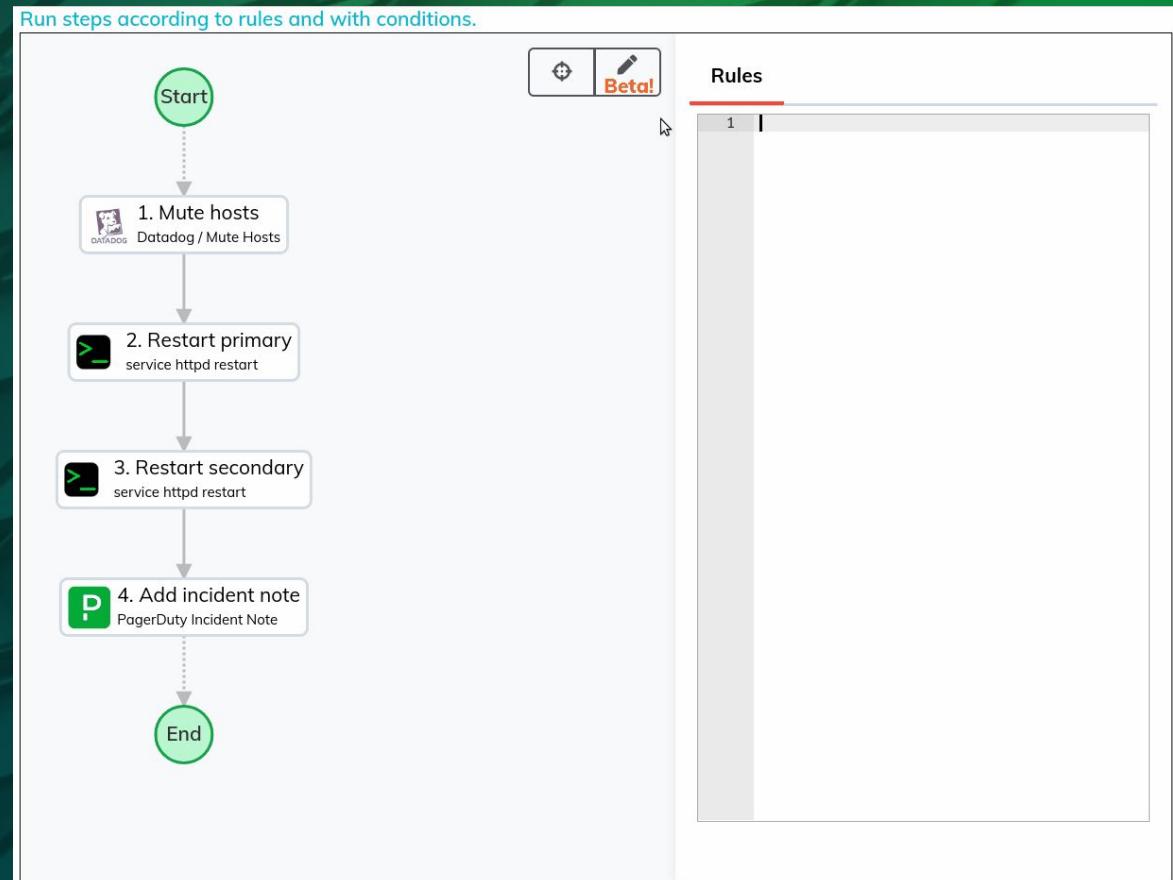
Conclusion

Automation Actions can be triggered as part of an incident flow

- We can use any part of the incoming event to trigger upon, and any part of the event data can be passed on to use in automations.

Advanced Automation Concepts

Rulesets

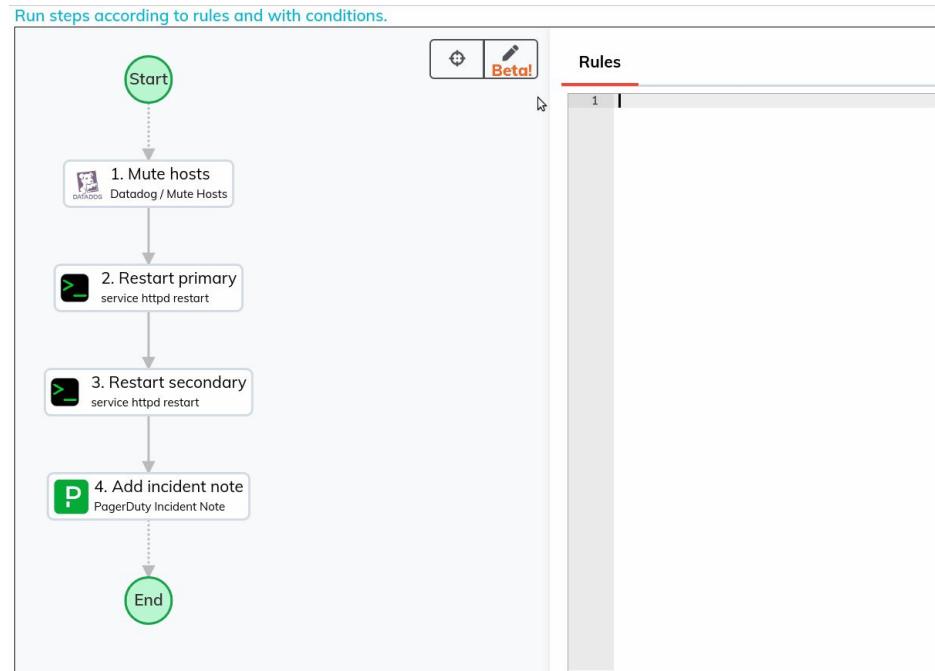


Rulesets

When you need logic you dont have in your script

IF THEN AND OR Logic

Graphic Flow Chart



Conditions

Conditions can define additional checks that must pass before a step can run, or determine when a step can be skipped. For example:

if:expression

unless:expression

expression defines a comparison or match that will be checked.

Valid expressions are:

key.name==string: a context variable such as option.myoption has a certain value

key.name!=string: not-equal check

key.name=~pattern: regular expression match

key.name!~pattern: negative regular expression match

!key.name context: variable is unset

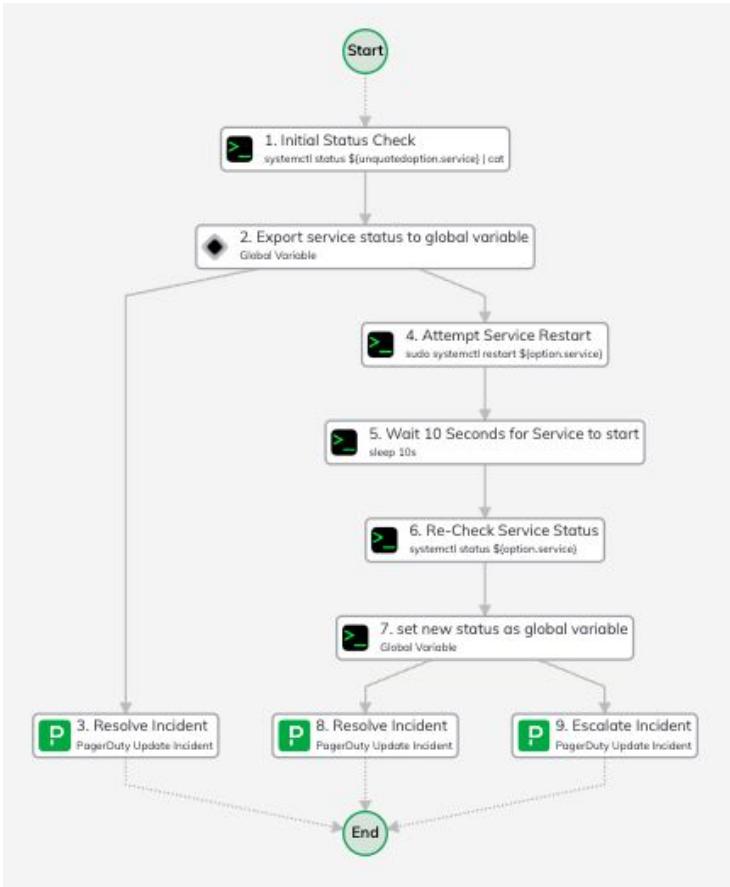
key.name>=number: greater than or equal to check

key.name<=number: less than or equal to check

key.name>number: greater than check

key.name<number: less than check

Example : Check, remediate,check,Escalate



TIP: Global Variables are essential -
Dont forget to export if you use the Ruleset

Exercise 4

Advanced Automation

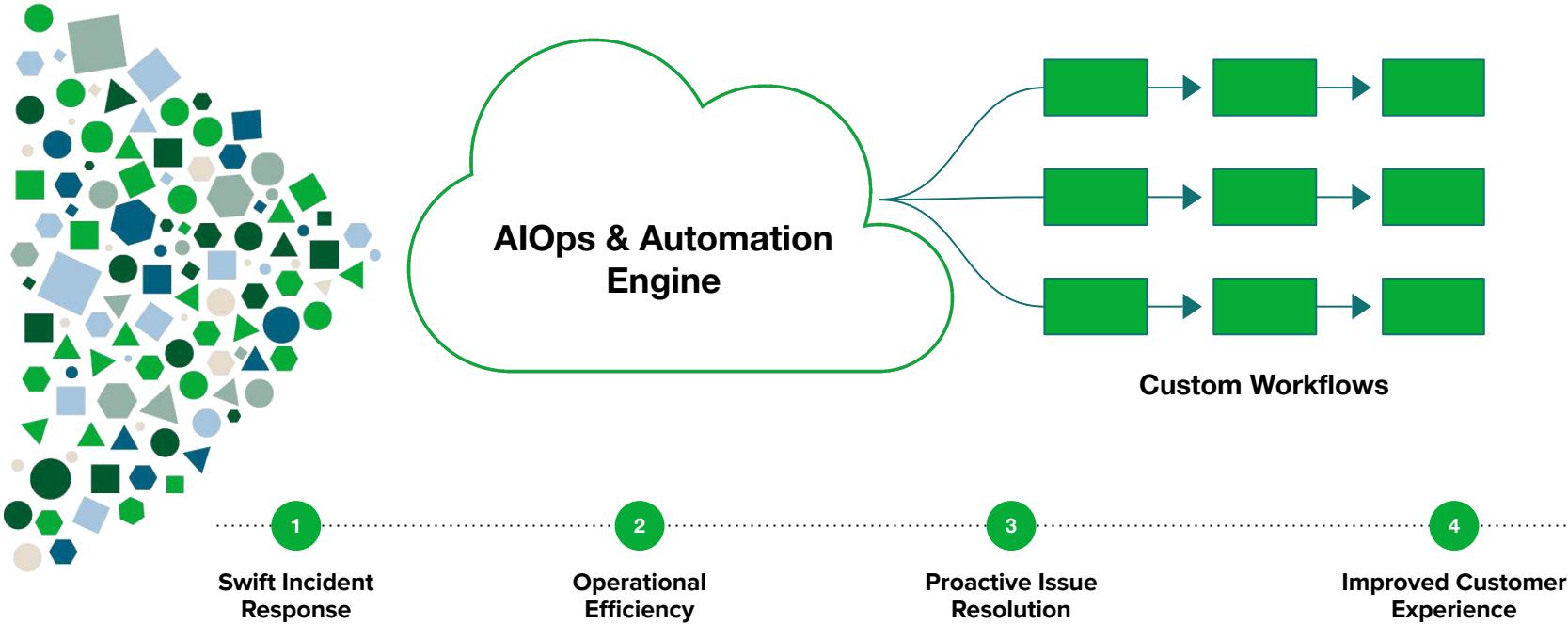
Conclusion

Automation Actions can be triggered as part of an incident flow

- We can use any part of the incoming event to trigger upon, and any part of the event data can be passed on to use in automations.

Event-Driven Automation

Event-driven automation is a dynamic approach where actions are triggered by specific events or conditions. It harnesses real-time event detection and predefined workflows to respond instantly to critical occurrences.



Solving Challenges with AIOps



Inundated
by alert noise

Fewer Incidents

- Identify incidents that matter



Confused about
where & what

Faster Resolution

- Improve situational awareness



Wasted time
in manual tasks

Greater Productivity

- Automate manual toil

Next steps

- 1 Continue your learning journey with AIOps and Automation
- 2 See how many manual tasks you can automate with Global Event Orchestration
- 3 Think of use cases that could be automated using the automation platform
- 4 Fill out the feedback form

Thank You!