

PagerDuty **ON TOUR**



To-Do: Getting ready

Those who have done the pre-work

- ✓ Log into your dev instance

Those who haven't

- ✓ Get a 'card' from one of the helpers
- ✓ Login using the username on the card

Everyone!

- ✓ Access "<https://press4ack.com>"
- ✓ Access "<https://bit.ly/4hYZ5ux>"
- ✓ Password is "pdot@sydney\$"

AP: Doltone
House Events

Password:
DHCorpClient

Operational Excellence: Scaling using AI and Automation



PagerDuty **ON TOUR**



Well understood

Teams have seen this issue before and know exactly what to do



100%
AI & Automation



Partially understood

Teams have seen this before and know potential remediations



AI & Automation +
Responder assisted



New & novel

Brand new incidents, or incidents requiring expert attention



Responder-led +
AI & Automation

Before we Begin

When you see this - People need to catch up!

PAUSE



Do this

SKIPPING ONBOARDING

→ **SKIP THIS STEP**

→ **NEVER MISS A PAGE**

→ **CLOSE THE ONBOARDING STEPS**

CONTINUE TO THE MAIN INCIDENT PAGE

See this

Verify your phone number to receive a test notification

We will send you a test incident via SMS, phone, and email. If you do not verify your number, you won't be able to receive SMS notifications.

There was an issue automatically fetching the country code.

Phone & SMS

Country code: +1 Phone number: 0000000000

Skip this step Send Verification

How do you want to be notified of incidents?

Your timezone: UTC+00:00 Greenwich Mean Time - London

Notification preferences for high urgency incidents

Never miss a page (Recommended)

Always text first

Always call first

Back Continue

Complete these steps to configure your account 0 / 3

- Enter and verify your phone number
- Download the mobile app
To receive push notifications, make sure you download the app
- Learn PagerDuty basics in less than 10 minutes

Go Go Go

Let's get started!

PagerDuty **ON TOUR**

1. Well Understood

100% AI and Automation

Scenario

A **Kubernetes pod has crashed** at 2am and your payment system is experiencing issues. You have **seen this incident many times** before and know that by restarting the pod the issue will resolve.

But **currently this is still done manually** by the SRE / ITOps team so after getting a notification from PagerDuty you will need to execute the restart manually. We don't want to wake up at 2am just to restart the pod.

Let's automate the entire end-to-end process by configuring the following

- RBA Job to remediate
- Automation Actions
- Event Orchestration

Do this

Let's look into RBA and see what jobs we can use (Step 1)

<https://bit.ly/4hYZ5ux>

User Name

dev-xxxx

pdot-xxxx

User Name

pdot@sydney\$

Hit Log In to join

See this

The screenshot shows the PagerDuty login interface. At the top, the PagerDuty logo is displayed in green. Below the logo, there is a legal notice in white text on a dark background: "By clicking on the 'Log In' button, you acknowledge that you have read and reviewed the Terms of Service and Privacy Policy and agree to be subject to those terms and policies." The main form area has two input fields: "Username" containing "student@25pdontour.com" and "Password" containing a series of dots. A blue "Log In" button is located at the bottom right of the form.

Do this

Let's look into RBA and see what jobs we can use (Step 2)

→ The jobs are already defined for you today.

→ Let's have a look at them. These will be used with the end-to-end workflow

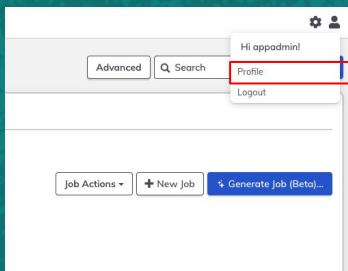
See this

The screenshot shows the RBA interface with the 'JOBS' icon highlighted in the sidebar. The main view displays a list of 'All Jobs' (12) under the 'Jobs' tab. The interface includes navigation tabs for 'Dashboard', 'Graph', and 'Show Favorites'. Filter options for 'LINUX (3)', 'REMEDIATION (1)', and 'WORKSHOP (1)' are available. The job list is organized into sections: 'Partially Understood' (with items like 'Code Rollback on System Failure', 'Diagnostic for Cloud Resources', 'Diagnostic Script for System Failure', and 'Fix for Cloud Budget Deviation'), 'Well Understood' (with items like 'Kubernetes Pod CrashLoopBackOff', 'Log File Growth', and 'SSL expiry detection'), and 'Today's Workshop' (with numbered steps 1 through 5). At the bottom, an 'Activity for Jobs' section shows 1 - 10 of 22 Executions from 'any time' with a 'Save Filter...' button. A single execution entry is shown: '29/03/2025 21:56 Yesterday at 21:56' with status '1 ok 9 seconds' and 'by P1WXGMT Well Understood/S'.

Do this

Lets connect these runbooks to PagerDuty (Step 1)

→ Click your profile silhouette and access “Profile”



→ Click ‘+’ next to “User API Tokens” and generate a token

→ Copy your token. You will use it in the next section.

See this

A screenshot of the AppDynamics Profile page. On the left, under "User API Tokens", there is a table listing tokens with columns "TOKEN" and "EXPIRATION". The tokens listed are "PagerDutyOnTourSydney" (expiring in 359d6h), "pdt-sydney" (expiring in 358d13h), "Self-Service-Automation" (expiring in 332d20h), "pdt-hsingh" (expiring in 241d4h), and "workshop-apj" (expiring in 142d10h). On the right, a modal dialog titled "Generate New Token" is open. It has fields for "Name" (set to "New Token - 2025-03-30 02:56:38"), "User" (set to "appadmin"), and "Roles" (empty). There is a "Expiration in" field set to "0 Minutes". A red box highlights the "Generate New Token" button at the bottom of the dialog. The bottom right corner of the main profile page shows the roles "appadmin" and "admin.user".

Do this

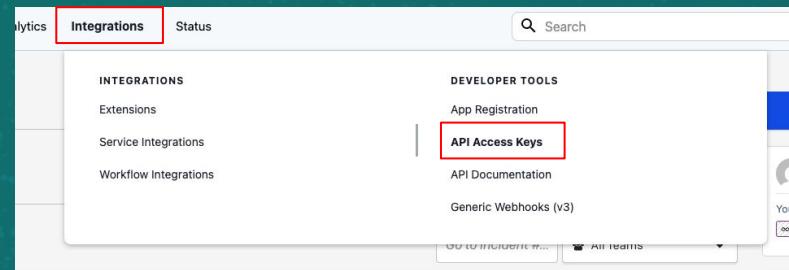
Lets connect PagerDuty to these Runbooks (Step 1)

→ Switch to your PagerDuty account and goto “Integrations” → “API Access Key”

→ Click on the “Create New API Key” button, enter a description and create one. (leave “Read-only API Key” unchecked)

→ Copy your token. You will use it in the next section.

See this

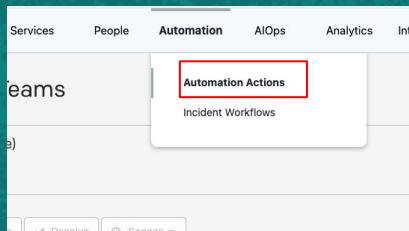


| API Access Keys | | | |
|--------------------------------------|-----------|--------------|------------------|
| + Create New API Key | | | |
| ID | API Key | Description | Created |
| P7DEXTX |2tXQ | Workshop key | Mar 2023 by K... |

Do this

Let's connect PagerDuty to these Runbooks (Step 1)

- On PagerDuty goto “Automation”
- “Automation Actions”



- Click on the “Add Action” button and select the “PDOT Runner”

- Select the “Kubernetes Pod CrashLoopBackOff” job and click next.

See this

Select Job

| Jobs | Project |
|--|-------------|
| <input type="radio"/> Diagnostic for Cloud Resources Diagnostic for Cloud Resources | PDOT-Sydney |
| <input type="radio"/> Diagnostic Script for System Failure Diagnostic Script for System Failure | PDOT-Sydney |
| <input type="radio"/> Fix for Cloud Budget Deviation Fix for Cloud Budget Deviation by Shutting Down Unused Resources | PDOT-Sydney |
| <input checked="" type="radio"/> Kubernetes Pod CrashLoopBackOff Script for the restart_crashloopbackoff_pod.sh to restart a Kubernetes pod that is stuck in CrashLoopBackOff | PDOT-Sydney |
| <input type="radio"/> Log File Growth Clear Unwanted Logs Files | PDOT-Sydney |
| <input type="radio"/> SSL expiry detection SSL Certificate Expiry Warning | PDOT-Sydney |

Back Next

Do this

Lets connect PagerDuty to these Runbooks (Step 2)

→ Scroll down to the “Services” section and from the “Find service(s)”.

→ From the pulldown choices select “Infrastructure Service” and click “Next”

See this

The screenshot shows a configuration dialog for associating an action with services. At the top, there's a search bar labeled "Select or add a category" with "Any Category" selected. Below it is a "Services" section with a checkbox for "Make this action available to run on all services." A dropdown menu titled "Find service(s)" lists "Application Service," "Database Service," and "Infrastructure Service." The "Infrastructure Service" option is highlighted with a red border. Underneath, there's an "Invocation Options" section with three checkboxes: "Only allow invocation on unresolved incidents" (unchecked), "Allow invocation manually" (checked), and "Allow invocation from event orchestration" (checked). At the bottom right of the dialog are "Back" and "Next" buttons.

Do this

Let's connect PagerDuty to these Runbooks (Step 3)

→ Scroll down to the “Enter argument (optional)” and enter below with spaces in between.

-pd_incident_id \${incident.id}
-pd_user_id pdot-user@pagerduty.com

People who have done the pre-work

-pd_api_key keys/project/PDOT-Sydney/dev-xxxx
Others
-pd_api_key keys/project/PDOT-Sydney/pdot_token

→ Click “Create Action” once finished.

→ If you accidentally clicked “Create Action” before entering the values then go back to action and “Edit”

See this

Create an Automation Action

SELECTED RUNNER

PDOT Runner
PDOT Runner (created via TF)

TYPE
Runbook Automation

STATUS

SELECTED JOB

Kubernetes Pod CrashLoopBackOff
Script for the restart_crashloopbackoff_pod.sh to restart a Kubernetes pod that is stuck in CrashLoopBackOff

PROJECT
PDOT-Sydney

NAME

Kubernetes Pod CrashLoopBackOff

DESCRIPTION

Script for the restart_crashloopbackoff_pod.sh to restart a Kubernetes pod that is stuck in CrashLoopBackOff

SERVICES

Infrastructure Service

Define your action

Enter arguments (optional)

-pd_incident_id \${incident.id} -pd_user_id \${user.id} -pd_api_key xxxxxxxxxxxxxxxxxxxx

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

Back

PAUSE

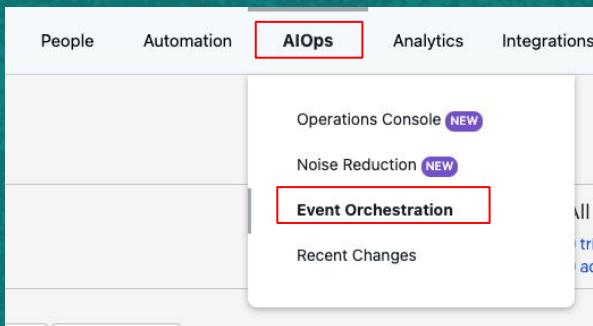


PagerDuty **ON
TOUR**

Do this

Let's orchestrate an alert to use the Automation Action (Step 1)

→ Goto “AIOps” → “Event Orchestration”



→ Select the “All Alerts” orchestration and go into “Service Routes”

→ Click on “New Service Route”

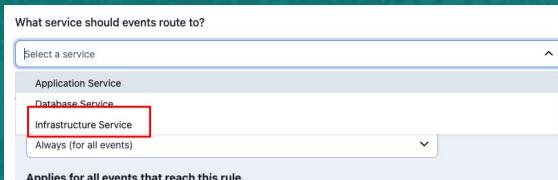
See this

A screenshot of the All Alerts orchestration page. At the top, there are tabs: Overview (which is highlighted with a blue underline), Integrations, Global Orchestration, Service Routes, and Service Orchestrations. Below the tabs, it says "All Alerts" and "Owner: team-user. Created on: March 28, 2025. Global orchestration for All Alerts". There are three main sections: "Integrations (1 integration)", "Global Orchestration (0 rules)", and "Service Routes (unrouted)". The "Service Routes" section has a red border around it. At the bottom, it says "All Alerts" and "Global orchestration owned by team-user. Can also be edited by admins or managers on any team.". It shows a message "You aren't routing these events to any services yet." with two buttons: "New Service Route" and "New Dynamic Route". A blue button labeled "+ New Service Route" is located on the right side of the "All Alerts" section.

Do this

Lets orchestrate an alert to use the Automation Action (Step 2)

→ For “What service should events route to?” choose “Infrastructure Service”

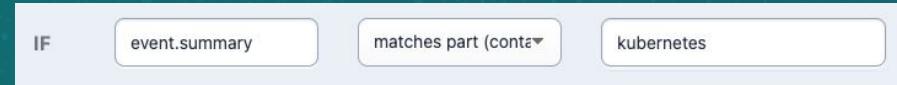
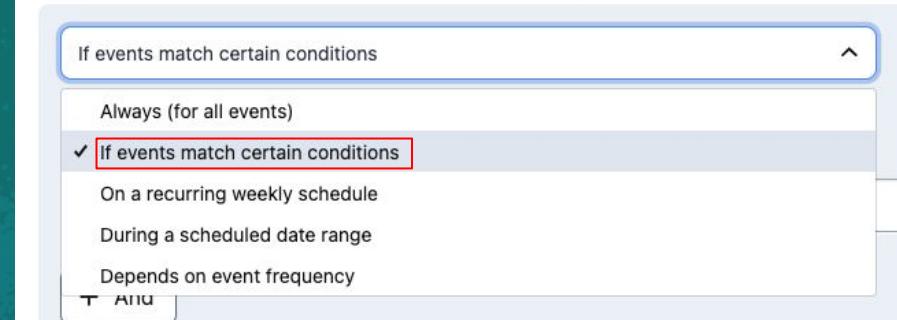


→ For “When should events be routed here?” Select “If events match certain conditions”

→ Enter the condition
IF “event.summary” matches part
“kubernetes”. Click “Save”

See this

When should events be routed here?



Do this

Lets orchestrate an alert to use the Automation Action (Step 3)

→ You should see the new route to the Infrastructure being added. Click on “Infrastructure Service Orchestration”

→ Once you enter into the Infrastructure service orchestration, click on the “+New Rule” button.

See this

All Alerts

Global orchestration owned by team-user. Can also be edited by admins or managers on any team.

Search

>Create a dynamic route

Are you sending service names or IDs in the payload?

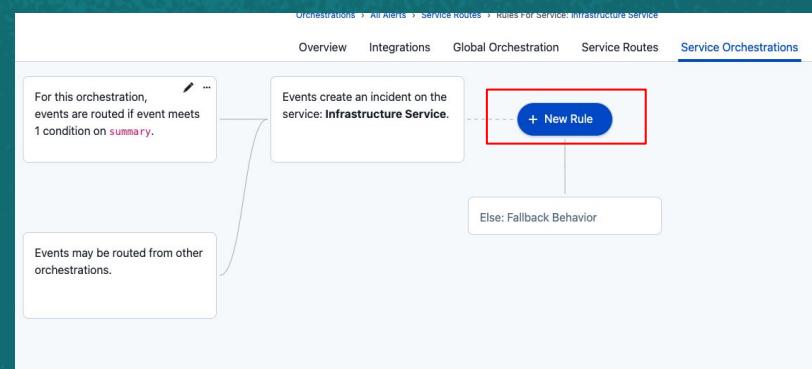
+ Set a field

1. If event meets 1 condition on [summary](#),
route to service: **Infrastructure Service**

Infrastructure Service Orchestration Service owned

2. Else: For all events, route to Unrouted Event
Orchestration

Unrouted Orchestration



Do this

Lets orchestrate an alert to use the Automation Action (Step 4)

→ Leave “When should this rule be applied?” set as “Always”. Click “Next”.

→ Under the “Automation” section of the rule select “Automation Actions” and select the action “Kubernetes Pod CrashLoopBackOff”. Click “Save”

See this

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

Incident Data

Basic Event

Event Fields

Custom Fields

Variables

Alert Data

Automation

Webhook Actions

Automation Actions

Automation Action

Kicks off an Automation Action as soon as an incident is created.

No automation selected

✓ No automation selected

Kubernetes Pod CrashLoopBackOff

Do this

Lets simulate the alert! (Step 1)

→ In your “All Alerts” Orchestration goto “Integrations” and copy the “Integration Key”. You will use this key to simulate the alert

See this

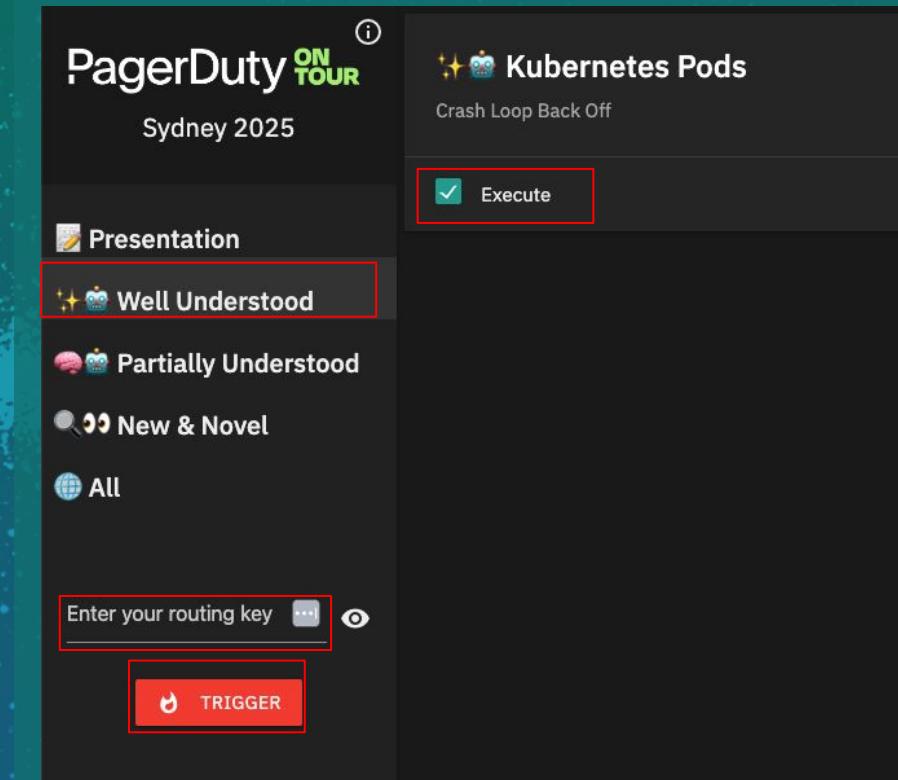
The screenshot shows a web interface for managing integrations. At the top, there are tabs: Overview, Integrations (which is selected), Global Orchestration, Service Routes, and Service Orchestrations. Below the tabs, the title "Integrations All Alerts" is displayed, followed by a note about integrating with any system and using an API v2 payload. A blue button labeled "+ New Integration" is visible in the top right corner. The main section is titled "All Alerts Default Integration". It contains three input fields: "Integration Key" (value: R028GKC6ZJ3PIFPUKX173EO2O0T16YDR), "HTTP Endpoint for API" (value: <https://events.pagerduty.com/v2/enqueue>), and "Email Address" (value: R028GKC6ZJ3PIFPUKX173EO2O0T16YDR@dev-kotsukajelidev.pagerduty.com). Each field has a "Copy to Clipboard" button to its right. At the bottom of the integration card is another "+ New Integration" button.

Do this

Lets simulate the alert! (Step 2)

- Open a new browser tab and goto "<https://press4ack.com/>"
- Click on “Well Understood” on the left menu bar and make sure the “Kubernetes Pods” alert is checked.
- Enter your “Integration Key” you have copied in the previous step into the “Enter your routing key” field.
- Click “Trigger”

See this



PAUSE



PagerDuty **ON
TOUR**

Do this

Your Incident is created.

This of course is core Incident Response functionality.

Look at the incident resolve automatically!

Check the below to see what happened:

- Notes
- Timeline
- Automation Action Log

See this

The screenshot shows a web-based incident management system. At the top, it displays a summary for a 'Kubernetes Pod CrashLoopBackOff - payment-service' incident. The status is 'Resolved', the urgency is 'High', and the main service is 'Infrastructure Service'. Below this, there's a section for 'Custom Fields' which is currently empty. In the 'Notes' section, a note from 'dot-user' on April 1, 2025, states 'Resolution Note: Restart Successful'. The 'Automation Actions Log' tab is selected, showing a single action for a 'Kubernetes Pod CrashLoopBackOff' that ran at 9:08am. The log details the script used to restart the pod. At the bottom, there are pagination controls for '5 per page' and '1 - 1'.

Base Incident FREQUENT Similar to 100% of incidents on this service in the preceding 30 days. View past similar incidents.

+ New Postmortem Report Run Workflow Run Actions ↗ Send Status Update More

STATUS Resolved INCIDENT TIMES Open from 9:08 AM to 9:08 AM (for a few seconds) URGGENCY High

MAIN SERVICE Infrastructure Service

ESCALATION POLICY ep-user IMPACTED SERVICE Infrastructure Service

SERVICE DESCRIPTION Managed by Terraform

Custom Fields

No custom fields configured for incidents. [Configure custom fields](#) to display them on this and other incidents.

Notes

Apr 1, 2025

P Resolution Note: Restart Successful
dot-user 9:08am

Health Check Details:
Starting Kubernetes Pod Restart
Namespace: application
Pod Name: application-service-7f9d9b7c8f-x2y4z
Deployment: application-service
Sending webhook notification for incident Q2QYC10R0RTSXU
Payload:
{
 "payload": {
 "summary": "Kubernetes Pod CrashLoopBackOff - application-service-7f9d9b7c8f-x2y4z",
 "timestamp": "2025-03-31T22:08:17Z",
 "severity": "error",
 "source": "Kubernetes Cluster - Demo",
 "class": "CrashLoopBackOff",
 "component": "application-service-7f9d9b7c8f-x2y4z",
 "custom_details": {
 "message": "Pod application-service-7f9d9b7c8f-x2y4z is in CrashLoopBackOff state."
 }
 }
}

| Status | Severity | Summary |
|---------|--------------|---------|
| 1 Alert | Grouping off | |

Alerts Status Updates Timeline Automation Actions Log

Automation Action / Node Info Start Ran by

Kubernetes Pod CrashLoopBackOff Script for the restart_crashloopbackoff job to restart a Kubernetes pod. an Event Orchestrator

1 [1:14pm] Job summary
2 [1:14pm] Job started: 9:08am
3 [1:14pm] Job ended: 9:08am
4 [1:14pm] Successful nodes: 1
5 [1:14pm] Failed nodes: 0

5 per page 1 - 1

2. Partially Understood

AI & Automation + Responder assisted

Scenario

A **system failure** has occurred due to **a new code that was deployed** for a Web Application Service. Similar incidents has happened before in the past but you still require **human in the loop** to eyeball key **context and information** before making any decisions around the resolution.

Currently the information and context gathering require **multiple escalations** and teams to be engaged. We want to prevent disrupting the team as much as possible.

Let's leverage AI and automation for the context gathering process

- Incident Workflow
- RBA Job to diagnose
- Automation Actions
- Event Orchestration
- AIOps (Demo)

Do this

Let's look into RBA and see what jobs we can use (Step 1)

<https://workshop2-apj.runbook.pagerduty.cloud/>

User Name

XXXXXX

User Name

pdot@sydney\$

Hit Log In to join

See this

The screenshot shows the PagerDuty login interface. At the top, the PagerDuty logo is displayed in green. Below the logo, there is a note in white text: "By clicking on the \"Log In\" button, you acknowledge that you have read and reviewed the Terms of Service and Privacy Policy and agree to be subject to those terms and policies." The main form area has two input fields: "Username" containing "student@25pdontour.com" and "Password" containing a series of asterisks. A blue "Log In" button is located at the bottom right of the form.

Do this

Let's look into RBA and see what jobs we can use

→ The jobs are already defined for you today.

→ Lets have a look at them. These will be used with the end-to-end workflow

See this

The screenshot shows the RBA interface with the 'JOBS' icon highlighted in the sidebar. The main view displays a list of 'All Jobs' (12) under the 'Jobs' tab. The list is categorized into 'LINUX (3)', 'REMEDIATION (1)', and 'WORKSHOP (1)'. The 'LINUX' category contains the following items:

- Code Rollback on System Failure
- Diagnostic for Cloud Resources
- Diagnostic Script for System Failure
- Fix for Cloud Budget Deviation

The 'REMEDIATION' category contains:

- Kubernetes Pod CrashLoopBackOff
- Log File Growth
- SSL expiry detection

The 'WORKSHOP' category contains:

- Today's Workshop
- 1. Check Directory for files less than 1k
- 2. Diagnostics - Top CPU & Memory Processes and Top Disk Consuming Files
- 3. Further Diagnostics (Student Modified)
- 4. Automation Action - Incident Response
- 5. Automated Remediation - Advanced Example (for reference only)

Below the job list, there is an 'Activity for Jobs' section showing 1 - 10 of 22 Executions. The most recent execution is dated 29/03/2025 at 21:56, which is marked as 'ok' and took 9 seconds. The status is 'Well Understood'.

Do this

Lets connect PagerDuty to these Runbooks (Step 1)

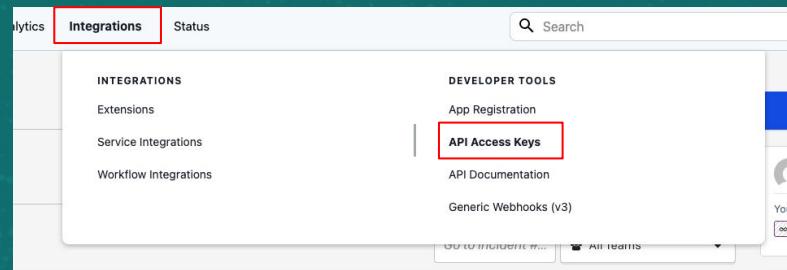
→ You can skip this step if you have the API key created on the “Well Understood” use case.

→ Switch to your PagerDuty account and goto “Integrations” → “API Access Key”

→ Click on the “Create New API Key” button, enter a description and create one. (leave “Read-only API Key” unchecked)

→ Copy your token. You will use it in the next section.

See this

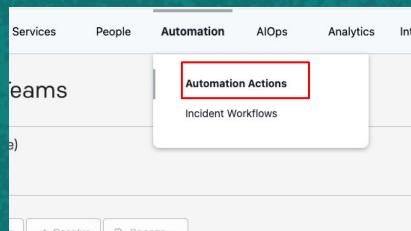


| API Access Keys | | | |
|--------------------------------------|-----------|--------------|---------------|
| + Create New API Key | | | |
| ID | API Key | Description | Created |
| P7DEXTX |2tXQ | Workshop key | Mar 1 by K... |

Do this

Lets connect PagerDuty to these Runbooks (Step 2)

→ Goto “Automation” → “Automation Actions”



→ Click on the “Add Action” button and select the “PDOT Runner”

→ Select the “Diagnostic Script for System Failure” job and click next.

See this

Select Job

| Jobs | Project |
|---|-------------|
| Code Rollback on System Failure Code Rollback on System Failure | PDOT-Sydney |
| Diagnostic for Cloud Resources Diagnostic for Cloud Resources | PDOT-Sydney |
| <input checked="" type="radio"/> Diagnostic Script for System Failure Diagnostic Script for System Failure | PDOT-Sydney |
| Fix for Cloud Budget Deviation Fix for Cloud Budget Deviation by Shutting Down Unused Resources | PDOT-Sydney |
| Kubernetes Pod CrashLoopBackOff Script for the restart_crashloopbackoff_pod.sh to restart a Kubernetes pod that is stuck in CrashLoopBackOff | PDOT-Sydney |

Back Next

Do this

Lets connect PagerDuty to these Runbooks (Step 3)

→ Scroll down to the “Services” section and from the “Find service(s)”.

→ From the pulldown choices select “Application Service” and click “Next”

See this

The screenshot shows a configuration dialog for associating an action with services. At the top, there's a search bar labeled "Select or add a category" with "Any Category" selected. Below it is a "Services" section with a checkbox for "Make this action available to run on all services." A dropdown menu titled "Find service(s)" lists several options: "Application Service" (which is highlighted with a red border), "Database Service", "Infrastructure Service", and "Find team(s)". Underneath, there's an "Invocation Options" section with three checkboxes: "Only allow invocation on unresolved incidents" (unchecked), "Allow invocation manually" (checked), and "Allow invocation from event orchestration" (checked). At the bottom right of the dialog are "Back" and "Next" buttons.

Do this

Lets connect PagerDuty to these Runbooks (Step 4)

→ Scroll down to the “Enter argument (optional)” and enter below with spaces in between.

-pd_incident_id \${incident.id}
-pd_user_id pdot-user@pagerduty.com
-pd_apikey <your_api_key>

→ Click “Create Action” once finished.

→ If you accidentally clicked “Create Action” before entering the values then go back to action and “Edit”

See this

Create an Automation Action

SELECTED RUNNER

| | | |
|------------------------------|--------|-------------------------------------|
| PDOT Runner | TYPE | Runbook Automation |
| PDOT Runner (created via TF) | STATUS | <input checked="" type="checkbox"/> |

SELECTED JOB

| | | |
|--------------------------------------|---------|-------------|
| Diagnostic Script for System Failure | PROJECT | PDOT-Sydney |
| Diagnostic Script for System Failure | | |

NAME

Diagnostic Script for System Failure

DESCRIPTION

Diagnostic Script for System Failure

SERVICES

Application Service

Define your action

Enter arguments (optional)

```
-pd_incident_id ${incident.id} -pd_user_id pdot-user@pagerduty.com -pd_apikey xxxxxxxxxxxxxxxx
```

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

[Back](#) [Create Action](#)

Do this

Lets connect PagerDuty to these Runbooks (Step 5)

→ Repeat Step 2~4 for the “Code Rollback on System Failure” Action

See this

Select Job

| Jobs | Project |
|---|-------------|
| Check/Remediate/Check Pattern with ongoing notes in Incident. Assumptions. - Key/Email are set at the plugin group level - You will need to modify both the check scripts and the remediation script (Steps 2, 8 and 9 by default) - Outputs for the check script are assumed to be OK or Failed - you will need to change the regex if anything else. - For these values OK or Failed - you can use the format *[1-13] unless:export.second_check=-OK.* if you want a NOT condition - Add options for additional parameters you might want to pass | PDOT-Sydney |
| <input checked="" type="radio"/> Code Rollback on System Failure Code Rollback on System Failure | PDOT-Sydney |
| <input type="radio"/> Diagnostic for Cloud Resources Diagnostic for Cloud Resources | PDOT-Sydney |
| <input type="radio"/> Diagnostic Script for System Failure Diagnostic Script for System Failure | PDOT-Sydney |
| <input type="radio"/> Fix for Cloud Budget Deviation | |

Back Next

PAUSE



PagerDuty **ON TOUR**

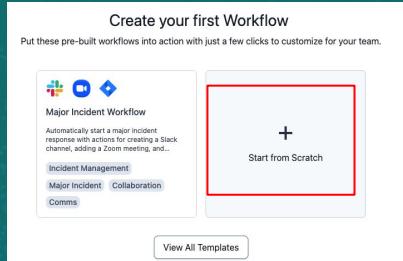
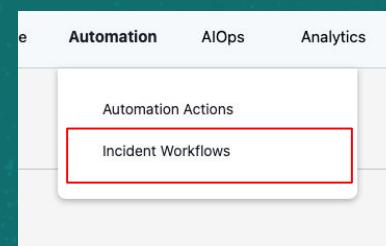
Do this

Lets build the diagnostic job into the incident as a workflow (Step 1)

→ Goto “Automation” → “Incident Workflows”

→ Click “Start from Scratch” and name your workflow “Auto-diagnostic workflow” and “Create”

See this

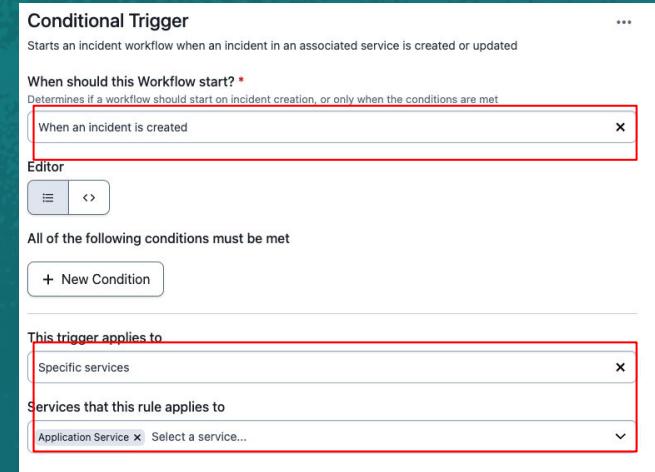
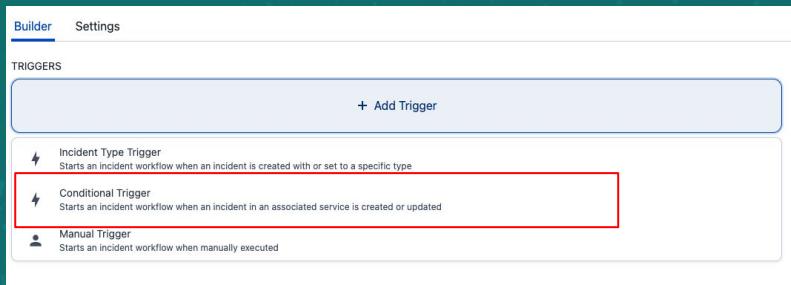
A screenshot of a "Create Incident Workflow" dialog box. It includes fields for "Name*" (containing "Auto-diagnostic workflow"), "Description" (with a note about understanding the workflow), and "Who can edit this Workflow?" (set to "All admins and global managers"). At the bottom right are "Cancel" and "Create" buttons, with the "Create" button highlighted by a red rectangular box.

Do this

Lets build the diagnostic job into the incident as a workflow (Step 2)

- In the builder click “+ Add Trigger” and select “Conditional Trigger”
- On the right side of the editor, for “When should this Workflow start?” select “When an incident is created”
- For “This trigger applies” to select “Specific services” and choose “Application Services”
- Click “Save”

See this



Do this

Lets build the diagnostic job into the incident as a workflow (Step 3)

- In the builder click “+ Add Action” and
- From the window that pops up scroll to find “PagerDuty Incident Management” → “Run an Automation Action”
- Click “Save” and then on the top left section of the builder click “Publish” → “Publish” to enable the workflow.

See this

The screenshot shows the PagerDuty Incident Management builder interface. At the top, there's a sidebar with icons for 'P' (PagerDuty Incident Management), '16 actions', 'Roles', and '2 actions'. Below the sidebar, the main area has a title 'Run an Automation Action' with a subtitle 'Run scripts installed on your infrastructure through PagerDuty Runbook Automation or Process Automation'. A dropdown menu titled 'Automation Action *' is open, showing several options: 'Diagnostic Script for System Failure' (selected), 'Code Rollback on System Failure' (Process Automation), 'Diagnostic Script for System Failure' (selected), and 'Kubernetes Pod CrashLoopBackOff' (Process Automation). At the bottom of the dropdown are 'Remove', 'Cancel', and 'Save' buttons.

PAUSE

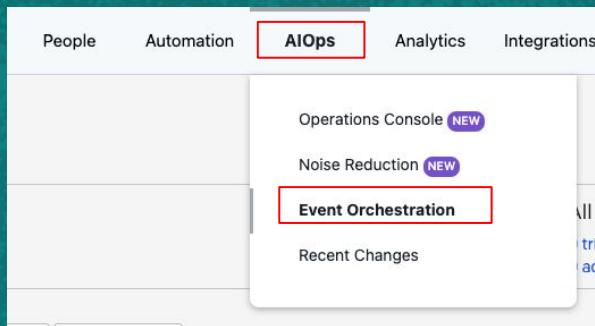


PagerDuty **ON
TOUR**

Do this

Lets orchestrate an alert to route to the Database Service (Step 1)

→ Goto “AIOps” → “Event Orchestration”



→ Select the “All Alerts” orchestration and go into “Service Routes”

→ Click on “New Service Route”

See this

A screenshot of the All Alerts orchestration page. At the top, there are tabs: Overview (which is highlighted with a blue underline), Integrations, Global Orchestration, Service Routes, and Service Orchestrations. Below the tabs, it says "All Alerts" and "Owner: team-user. Created on: March 28, 2025. Global orchestration for All Alerts". There are three main sections: "Integrations (1 integration)", "Global Orchestration (0 rules)", and "Service Routes (unrouted)". The "Service Routes" section has a red border around it. At the bottom, it says "All Alerts" again, "Global orchestration owned by team-user. Can also be edited by admins or managers on any team.", and "You aren't routing these events to any services yet." with buttons for "New Service Route" and "New Dynamic Route".

Overview Integrations Global Orchestration Service Routes Service Orchestrations

All Alerts

Owner: team-user. Created on: March 28, 2025. Global orchestration for All Alerts

Integrations (1 integration)
Connect your alerting systems to PagerDuty

Global Orchestration (0 rules)
Create rules to process, transform, and automate, before dispersing to services.

Service Routes (unrouted)
Send events to the right service.

All Alerts

Global orchestration owned by team-user. Can also be edited by admins or managers on any team.

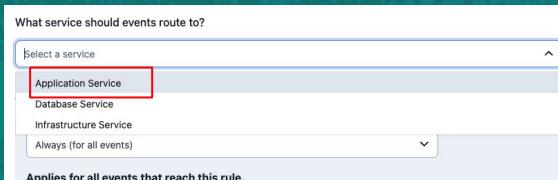
You aren't routing these events to any services yet.

New Service Route New Dynamic Route

Do this

Lets orchestrate an alert to use the Automation Action (Step 2)

→ For “What service should events route to?” choose “Application Service”

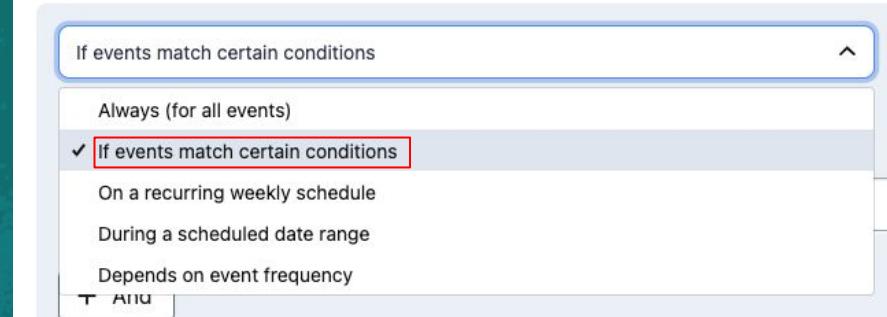


→ For “When should events be routed here?” Select “If events match certain conditions”

→ Enter the condition
IF “event.summary” matches part
“system failure”. Click “Save”

See this

When should events be routed here?



Do this

Lets simulate the alert! (Step 1)

→ In your “All Alerts” Orchestration goto “Integrations” and copy the “Integration Key”. You will use this key to simulate the alert

See this

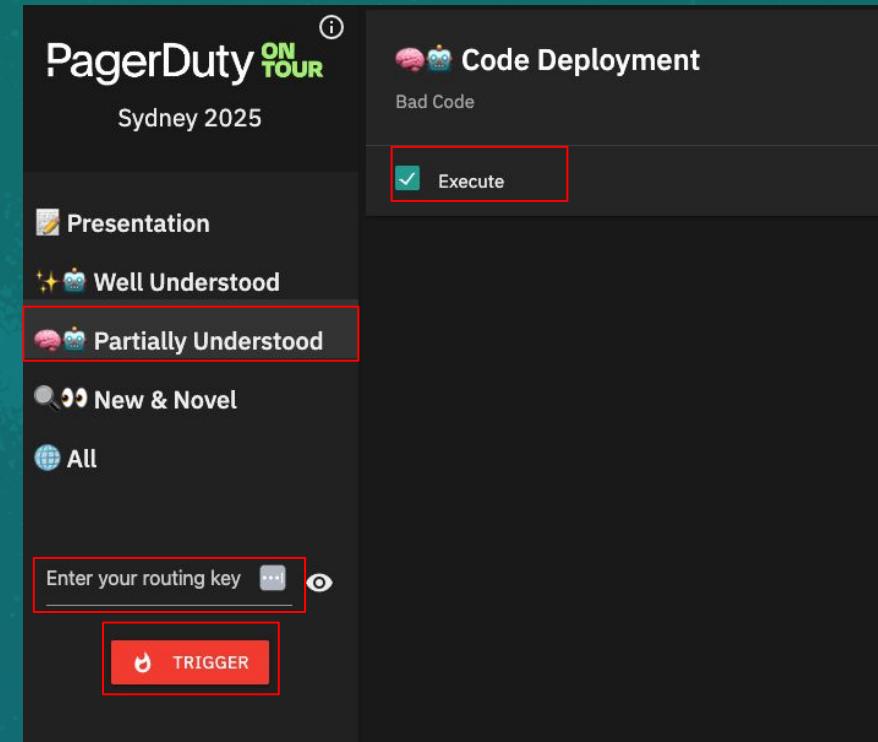
The screenshot shows a web interface for managing integrations. At the top, there are tabs: Overview, Integrations (which is selected), Global Orchestration, Service Routes, and Service Orchestrations. Below the tabs, the title 'Integrations All Alerts' is displayed, followed by a note about integrating with any system and using an API v2 payload. A blue button '+ New Integration' is visible on the right. The main section is titled 'All Alerts Default Integration'. It contains three input fields: 'Integration Key' (R028GKC6ZJ3PIFPUKX173EO2O0T16YDR), 'HTTP Endpoint for API' (https://events.pagerduty.com/v2/enqueue), and 'Email Address' (R028GKC6ZJ3PIFPUKX173EO2O0T16YDR@dev-kotsukajelidev.pagerduty.com). Each field has a 'Copy to Clipboard' button to its right. At the bottom left of this section is another '+ New Integration' button.

Do this

Lets simulate the alert! (Step 2)

- Open a new browser tab and goto "<https://press4ack.com/>"
- Click on “Partially Understood” on the left menu bar and make sure the "Code Deployment" alert is checked.
- Enter your “Integration Key” you have copied in the previous step into the “Enter your routing key” field.
- Click “Trigger”

See this



Do this

Lets simulate the alert! (Step 3)

→ On PagerDuty click on “Incidents” → “All Incidents”

→ You should see a new incident being triggered called “Unexpected System Failure Detected”. Click on the title.

→ You should see the diagnostic information being posted on the “Notes” section of the incident asking you to rollback”

See this

The screenshot shows the PagerDuty web interface. At the top, there's a navigation bar with 'PagerDuty' on the left and 'Incidents' and 'Services' on the right. Below the navigation bar, a sidebar on the left has 'RESPONSE' at the top, followed by 'All Incidents' (which is highlighted with a red box), 'Alerts', and 'Visibility Console'. At the bottom of the sidebar is a link 'Jeli Post Incident Review'. The main content area is titled 'Incident' and displays 'Your open incidents' with counts '0 triggered' and '0 acknowledged'. Below this is a table with columns: Status, Priority, Urgency, Type, and Title. A single row is shown, indicating a 'Triggered' status, priority '--', urgency 'High', type 'Base Incident', and title 'Unexpected System Failure Detected'. A red box highlights the 'Title' column for this row. To the right of the table, there are two buttons: 'SHOW DETAILS (1 triggered alert)' and a small 'X' icon.

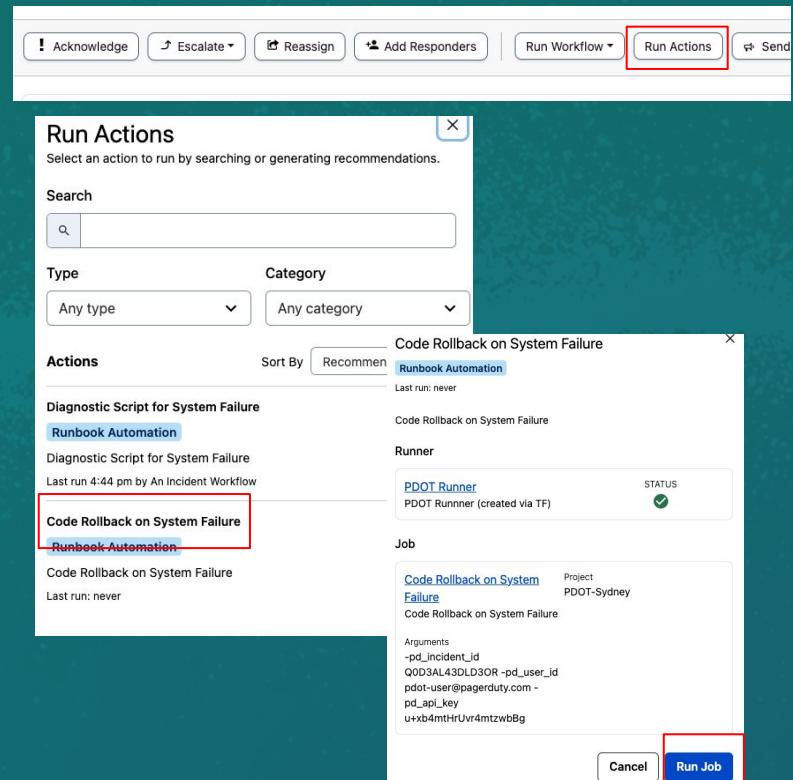
Do this

Lets simulate the alert! (Step 4)

→ On the incident click the “Run Actions” button and select “Code Rollback on System Failure”

→ Click “Run Job”. You should see the output of the job taking action for the rollback

See this



PAUSE and AIOps Demo



PagerDuty **ON TOUR**

Do this

Your Incident is created.

This of course is core Incident Response functionality.

Look at the incident with the context gathered! Now resolve it!

Check the below to see what happened:

- Notes
- Timeline
- Automation Action Log

See this

The screenshot shows a web-based incident management system. At the top, a header reads "Base Incident" and "Kubernetes Pod CrashLoopBackOff - payment-service". A status indicator says "FREQUENT" and notes "Similar to 100% of incidents on this service in the preceding 30 days. View past similar incidents." Below the header are buttons for "New Postmortem Report", "Run Workflow", "Run Actions", "Send Status Update", and "More".

The main incident card displays the following details:

- STATUS:** Resolved
- INCIDENT TIMES:** Open from 9:08 AM to 9:08 AM (for a few seconds)
- URGENCY:** High
- MAIN SERVICE:** Infrastructure Service
- IMPACTED SERVICE:** Infrastructure Service
- SERVICE DESCRIPTION:** Managed by Terraform

A section titled "Custom Fields" states: "No custom fields configured for incidents. Configure custom fields to display them on this and other incidents."

A "Notes" section shows a log entry for "Apr 1, 2025": "Resolution Note: Restart Successful" by user "dot-user" at 9:08am.

The "Automation Actions Log" tab is selected, showing one alert: "1 Alert Grouping off". The log table has columns for "Status", "Severity", and "Summary".

The "Automation Actions Log" table lists a single action:

| Automation Action / Node | Info | Start | Ran by |
|---------------------------------|--|--------|-----------------------|
| Kubernetes Pod CrashLoopBackOff | Script for the restart_crashloopbackoff job to restart a Kubernetes pod. | 9:08am | an Event Orchestrator |

The "Summary" section of the log table shows the following log entries:

```
[1] [1:41pm] Job summary
[2] [1:41pm] Job started: 9:08am
[3] [1:41pm] Job ended: 9:08am
[4] [1:41pm] Successful nodes: 1
[5] [1:41pm] Failed nodes: 0
```

At the bottom right, there are pagination controls: "5 per page" and "1 - 1".

3. New & Novel

Responder-led + AI & Automation

Scenario

You recently ***had a database migration and it has caused an outage***. The incident you see is brand new to the team and it is causing ***disruption to your customers and to your business***. We need to mobilize key ***technical teams*** into a war room whilst making sure ***business stakeholders*** are aware of what is going on. Post incident reports need to be consolidate after the issue is resolved.

Currently this is still done manually whereby the initial responder would initiate the war room and escalate to the teams asking them to join and sending updates via email to the Stakeholders. Reports take days to consolidate. We want to move quickly as time matters.

Let's automate the entire end-to-end process by configuring the following

- Incident Workflows
- PagerDuty Advance & Jeli (Demo)

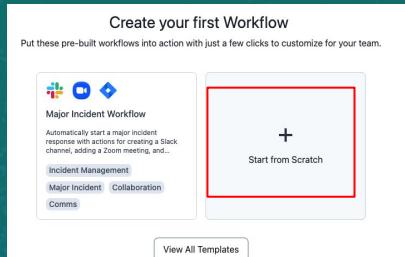
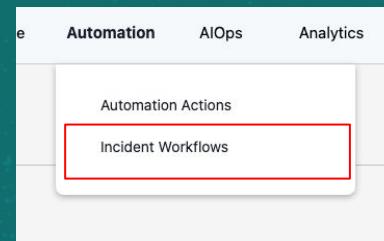
Do this

Lets build a war room scenario process as a workflow (Step 1)

→ Goto “Automation” → “Incident Workflows”

→ Click “Start from Scratch” and name your workflow “War Room workflow” and “Create”

See this

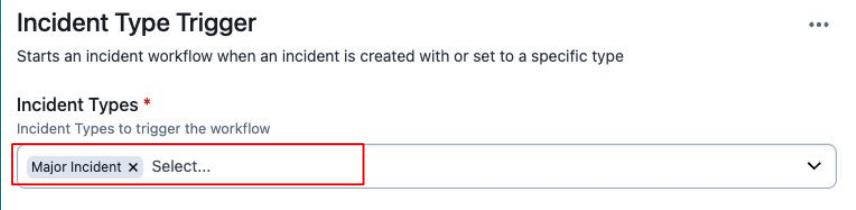
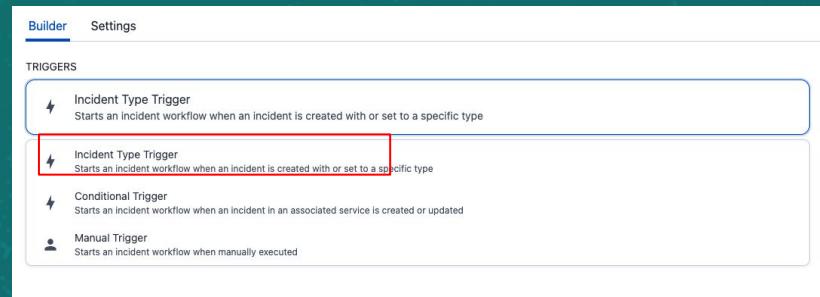
A screenshot of a "Create Incident Workflow" dialog box. It includes fields for "Name" (containing "War Room workflow"), "Description" (with a note about helping users understand), "Who can edit this Workflow?" (set to "All admins and global managers"), and a "Create" button at the bottom right which is highlighted with a red box.

Do this

Lets build the diagnostic job into the incident as a workflow (Step 2)

- In the builder click “+ Add Trigger” and select “Incident Trigger”
- On the right side of the editor, for “Incident Types” select “Major Incident”
- Click “Save”

See this



Do this

Lets build the diagnostic job into the incident as a workflow (Step 3)

- In the builder click “+ Add Action” and
- From the window that pops up scroll to find “PagerDuty Incident Management” and add the following actions
 - Add Responders: Select users
 - Add Stakeholders: Select yourself
 - Add Conference Bridge:
 - Send Status Update: enter a custom message
- Click “Save” and then on the top left section of the builder click “Publish” → “Publish” to enable the workflow.

See this



PagerDuty Incident Management

16 actions



Roles

2 actions



Add Conference Bridge

Adds a phone number and/or URL to an Incident



Add Responders

Add users or escalation policies as responders to



Add Stakeholders

Subscribe teams or users to status updates for an



Send Status Update

Post an update to the internal status pa

PAUSE



PagerDuty **ON TOUR**

Do this

Lets simulate the incident! (Step 1)

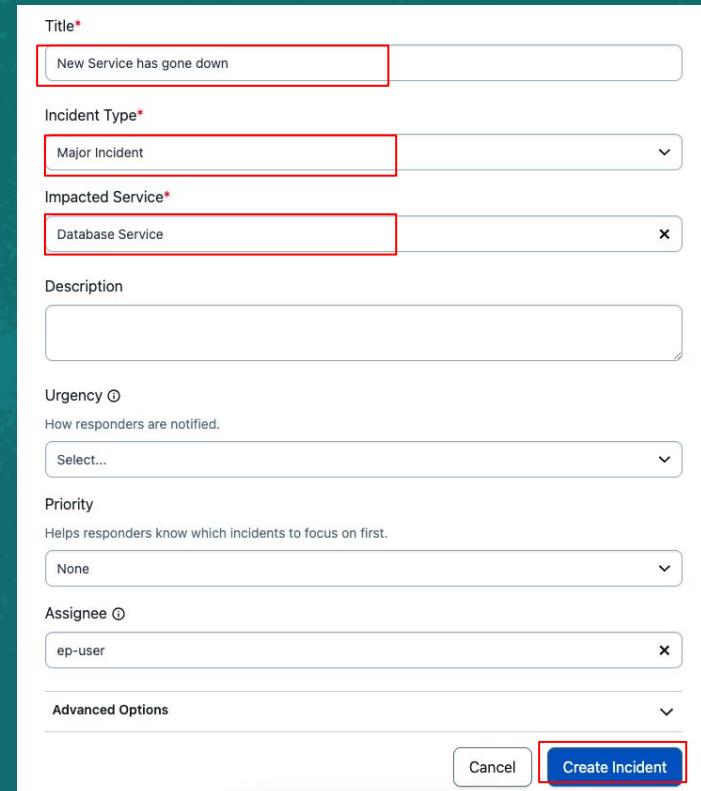
→ Click on “+ New Incident” button and create a example incident. Give it a Title!

→ Select “Major Incident” as the Incident Type

→ Select the “Database Service”

→ Click on “Create Incident”

See this



The screenshot shows a 'Create Incident' form with the following fields and values:

- Title***: New Service has gone down
- Incident Type***: Major Incident
- Impacted Service***: Database Service
- Description**: (Empty text area)
- Urgency**: (Empty dropdown menu)
- How responders are notified**: (Empty dropdown menu)
- Priority**: None
- Helps responders know which incidents to focus on first**: (Empty dropdown menu)
- Assignee**: ep-user
- Advanced Options**: (Empty dropdown menu)

At the bottom right of the form are two buttons: 'Cancel' and 'Create Incident'. The 'Create Incident' button is highlighted with a red border.

Do this

Your Incident is created.

This of course is core Incident Response functionality.

Look at the incident resolve automatically!

Check the below to see what happened:

- Responders
- Timeline
- Status Update

See this

The screenshot shows a PagerDuty incident details page for a 'Major Incident' titled 'New Service Outage'. The incident was triggered at 1:52 PM (a minute ago). It is assigned to 'pdot-user' and has an escalation policy 'ep-user' with a conference number '+61 404370845' and URL 'https://example.meeting.com'. The service is 'Database Service' managed by Terraform. There is one responder listed. Pending actions include resolving the incident automatically at 5:52 PM (in 4 hours) if left open. A section for custom fields indicates none are configured. The 'Status Updates' tab is active, showing a status update from 'pdot-user' assigned at 1:52 PM (a few seconds ago). Other responders listed are 'Unassigned Incident Commander' and 'Unassigned Customer Liaison'. The 'Status Dashboard' shows this incident is not visible on the status dashboard. Buttons for 'Add Responders' and 'Assign Roles' are at the bottom right.

PAUSE and Demo PD Advance and Jeli



PagerDuty **ON TOUR**