



# 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



# 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

# 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

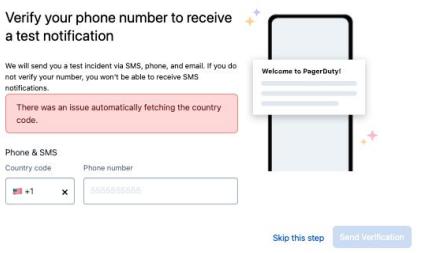
We will send you a test incident via SMS, phone, and email. If you do not verify your number, you won't be able to receive SMS notifications.

There was an issue automatically fetching the country code.

Phone & SMS

Country code:  Phone number:

[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)

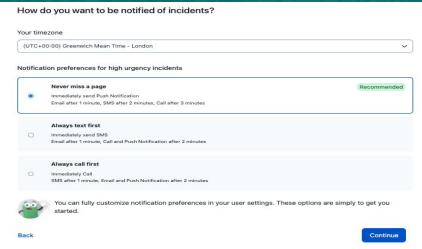
Immediately send SMS: Email after 1 minute, SMS after 2 minutes, Call after 3 minutes

Always text first

Always call first

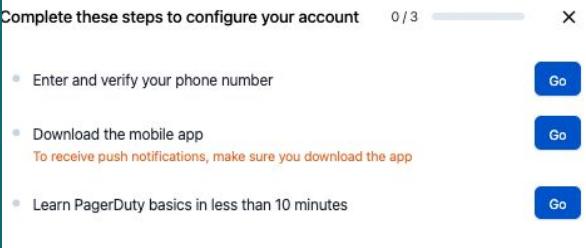
 You can fully customize notification preferences in your user settings. These options are simply to get you started.

[Back](#) [Continue](#)



Complete these steps to configure your account 0 / 3

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



# Let's get started!

PagerDuty **ON TOUR**

# Scenario



## Well understood

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



100%  
AI & Automation

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 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@pagerduty.com

Password

pdot@sydney\$

Hit Log In to join

## See this

PagerDuty

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.

Username

pdot-accra@pagerduty.com



Password

.....



Log In

## 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

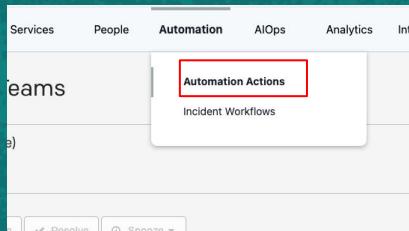
The screenshot shows the RBA interface with the following details:

- Header:** PDOT-Sydney
- Sidebar:** A vertical sidebar on the left with icons for Dashboard, Jobs (highlighted with a red box), Nodes, Commands, Activity, Runner Management, Webhooks, Schedules, Calendars, and Project Settings.
- Main Content:**
  - All Jobs:** 12
  - Jobs Tab:** Selected tab.
  - Filter Buttons:** LINUX (3), REMEDIATION (1), WORKSHOP (1).
  - Job Categories:**
    - Partially Understood:**
      - Code Rollback on System Failure
      - Diagnostic for Cloud Resources
      - Diagnostic Script for System Failure
      - Fix for Cloud Budget Deviation
    - Well Understood:**
      - Kubernetes Pod CrashLoopBackOff
      - Log File Growth
      - SSL expiry detection
    - Today's Workshop:**
      - Check Directory for files less than 1k > Sample copied from the Command
      - Diagnoses - Top CPU & Memory Processes and Top Disk Consuming Files
      - Further Diagnostics (Student Modified)
      - Automation Action - Incident Response
      - Automated Remediation - Advanced Example (for reference only)
  - Activity for Jobs:** Shows 1 - 10 of 22 Executions, filtered by 'any time'. One execution is listed: 29/03/2025 21:56 Yesterday at 21:56, status 1 ok, 9 seconds, by P1WXGMT, Well Understood/S.

## 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)

→ Put your “<username>” in front of the “Action name” to make it unique.

→ 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 page with the following sections:

- Select or add a category:** A dropdown menu showing "Any Category".
- Services:** A section with a checkbox for "Make this action available to run on all services." Below it is a "Associate this action with service(s) to make the action available to run on the services." section. This section includes a "Find service(s)" input field and a list of service types: "Application Service", "Database Service", and "Infrastructure Service". The "Infrastructure Service" option is highlighted with a red border.
- Invocation Options:** A section with three checkboxes:
  - Only allow invocation on unresolved incidents
  - Allow invocation manually
  - Allow invocation from event orchestration

At the bottom right of the form 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 xxxxxxxxxxxxxxxxx

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

# PAUSE

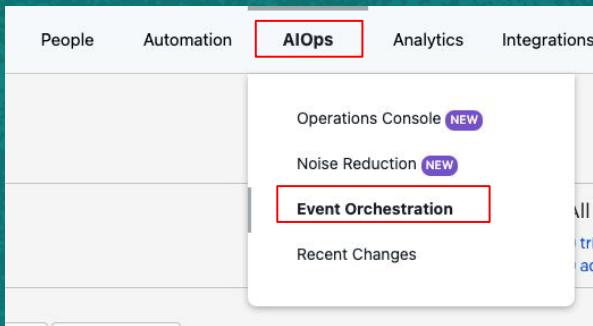


PagerDuty **ON TOUR**

## 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”

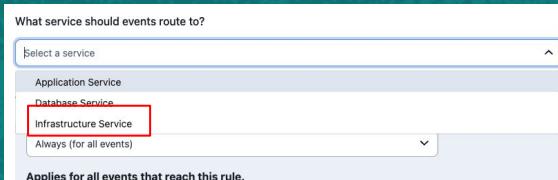
## See this

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" 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".

## Do this

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

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

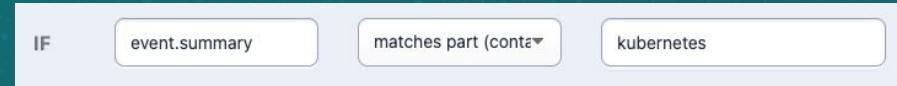
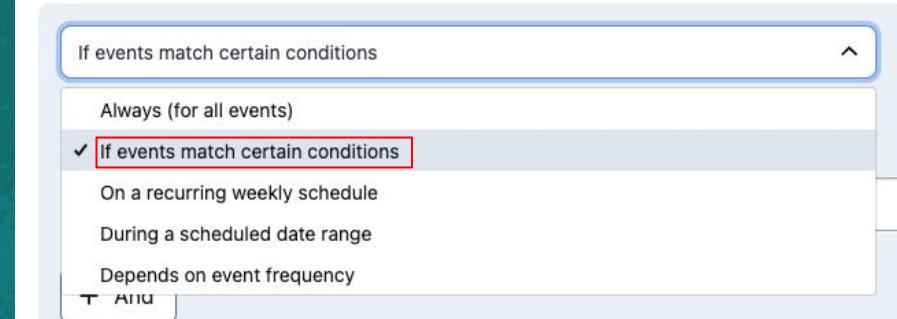


→ For “When should events be routed here?” Select “If events match certain conditions”

→ 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

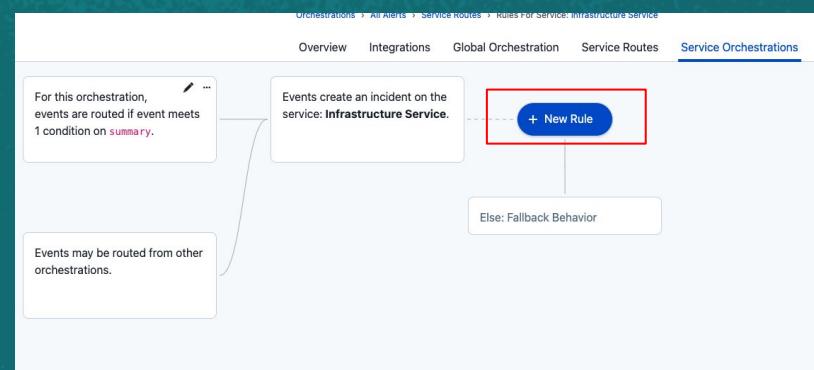
1. If event meets 1 condition on [summary](#),  
route to service: **Infrastructure Service**

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

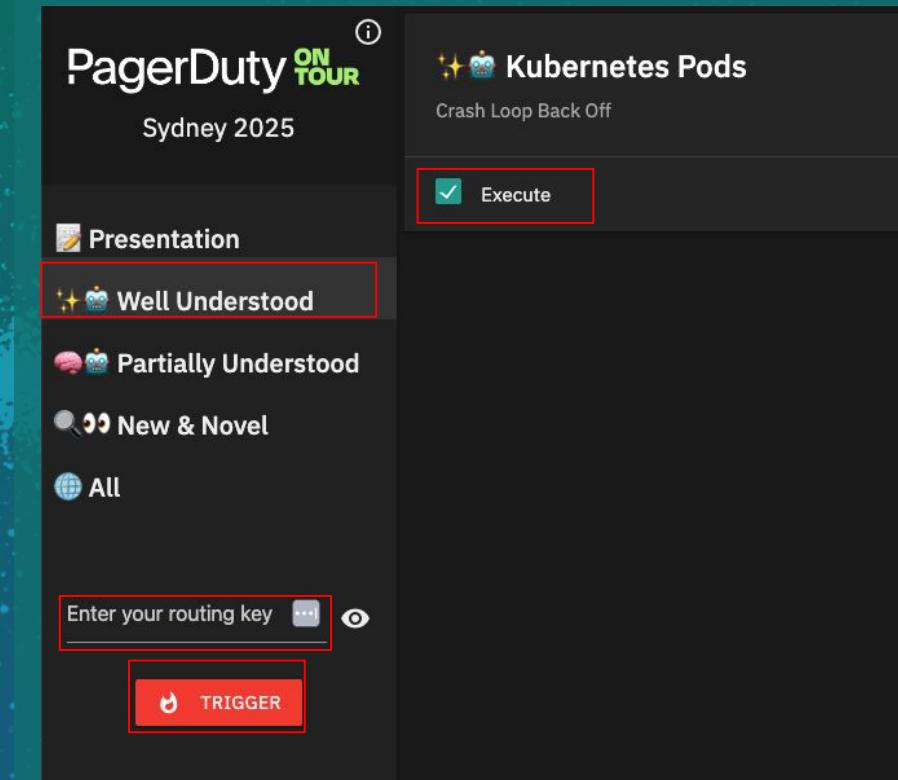
The screenshot shows a user interface for managing integrations. At the top, there are tabs: Overview, Integrations (which is selected), Global Orchestration, Service Routes, and Service Orchestrations. Below the tabs, the title "Integrations All Alerts" is displayed, followed by a note about integrating 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" with value "R028GKC6ZJ3PIFPUKX173EO2O0T16YDR", "HTTP Endpoint for API" with value "https://events.pagerduty.com/v2/enqueue", and "Email Address" with value "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 “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 detailed view of an incident in a cloud-based incident management system. At the top, it displays the title "Kubernetes Pod CrashLoopBackOff - payment-service" with a status indicator "FREQUENT". Below this, there are buttons for "New Postmortem Report", "Run Workflow", "Run Actions", "Send Status Update", and "More". The main card shows the incident is "Resolved" with an "Open from 9:08 AM to 9:08 AM (for a few seconds)" duration. It lists the "Escalation Policy" as "ep-user" and includes service-level details: "Main Service" is Infrastructure Service, "Impacted Service" is Infrastructure Service, and "Service Description" is "Managed by Terraform". A "Custom Fields" section indicates no custom fields are configured. The "Notes" section shows a log entry from April 1, 2025, at 9:08am, where a user named "dot-user" noted a successful restart. The "Automation Actions Log" tab is selected, showing a single action for a "Kubernetes Pod CrashLoopBackOff" incident. The log entries are as follows:

```
[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
```

Payload content (redacted):

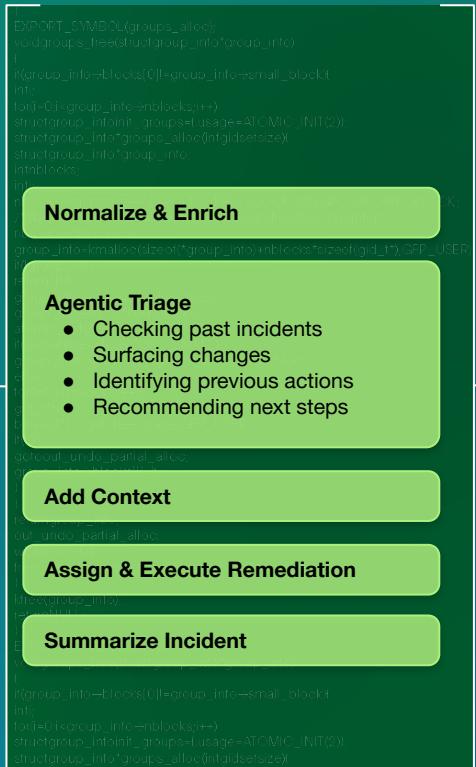
```
{
  "payload": {
    "summary": "Kubernetes Pod CrashLoopBackOff - application-service-7f9d9b7c8f-x2y4z",
    "timestamp": "2025-03-31T22:08:17Z",
    "severity": "error",
    "source": "Kubernetes Cluster - Demo",
    "class": "CrashLoopBackOff",
    "component": "application-service-7f9d9b7c8f-x2y4z",
    "custom_details": {
      "message": "Pod application-service-7f9d9b7c8f-x2y4z is in CrashLoopBackOff state."
    }
}
```

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

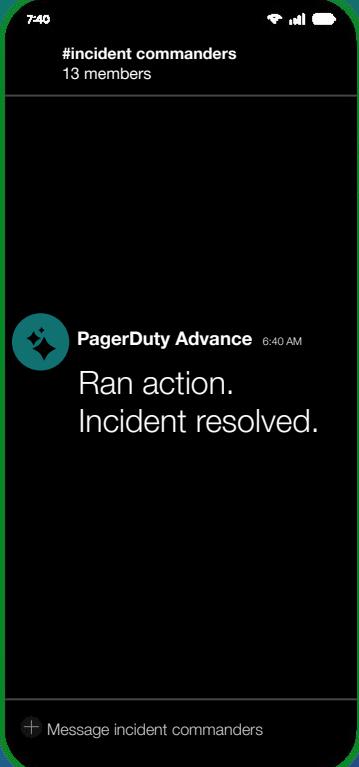


## Well understood incident

Events  
CMDB Data  
Customer Incidents  
Telemetry  
Diagnostics  
Logs



## Post Incident Learning



Resolved

PagerDuty **ON TOUR**

# Scenario



## Partially understood

Teams have seen this before and know potential remediations



AI & Automation + Responder assisted

A **system failure** has occurred due to **a new code that was deployed** for a Web Application Service. Similar incidents have 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://bit.ly/4hYZ5ux>

User Name

dev-xxxx

pdot-xxxx@pagerduty.com

Password

pdot@sydney\$

Hit Log In to join

## See this

PagerDuty

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.

Username

pdot-accra@pagerduty.com



Password

.....



Log In

## Do this

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

→ 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

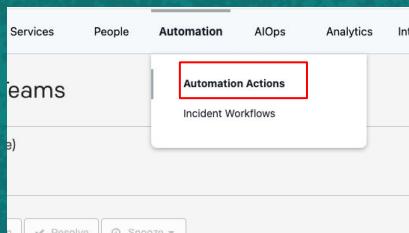
## See this

The screenshot shows the RBA interface with a sidebar on the left containing icons for Dashboard, Jobs (which is selected and highlighted with a red box), Nodes, Commands, Activity, Runner Management, Webhooks, Schedules, Calendars, and Project Settings. The main content area is titled 'All Jobs 12' and includes tabs for Jobs, Dashboard, Graph, and Show Favorites. Under the Jobs tab, there are filters for LINUX (3), REMEDIATION (1), and WORKSHOP (1). Below the filters are two expandable sections: 'Partially Understood' and 'Well Understood'. The 'Partially Understood' section contains four items: 'Code Rollback on System Failure', 'Diagnostic for Cloud Resources', 'Diagnostic Script for System Failure', and 'Fix for Cloud Budget Deviation'. The 'Well Understood' section contains five items: 'Kubernetes Pod CrashLoopBackOff', 'Log File Growth', 'SSL expiry detection', 'Today's Workshop' (with a sub-item '1. Check Directory for files less than 1k'), and 'Automated Remediation - Advanced Example (for reference only)'. At the bottom, there is a section titled 'Activity for Jobs' showing 1 - 10 of 22 Executions, a filter button, and a log entry for '29/03/2025 21:56 Yesterday at 21:56'.

## Do this

Lets connect PagerDuty to these Runbooks (Step 2)

- On PagerDuty 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
<small>Under development - check back later if you want to see more condition - Add options for additional parameters you might want to pass</small>	
<input type="radio"/> <b>Code Rollback on System Failure</b> Code Rollback on System Failure	PDOT-Sydney
<input type="radio"/> <b>Diagnostic for Cloud Resources</b> Diagnostic for Cloud Resources	PDOT-Sydney
<input checked="" type="radio"/> <b>Diagnostic Script for System Failure</b> Diagnostic Script for System Failure	PDOT-Sydney
<input type="radio"/> <b>Fix for Cloud Budget Deviation</b> Fix for Cloud Budget Deviation by Shutting Down Unused Resources	PDOT-Sydney
<input type="radio"/> <b>Kubernetes Pod CrashLoopBackOff</b> 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)

→ Put your “<username>” in front of the “Action name” to make it unique.

→ 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 interface for selecting a service. At the top, there is a search bar labeled "Select or add a category" with the placeholder "Any Category". Below the search bar is a section titled "Services" with a checkbox option "Make this action available to run on all services." Underneath this is a heading "Associate this action with service(s) to make the action available to run on the services." A dropdown menu titled "Find service(s)" is open, showing a list of service types: "Application Service" (which is highlighted with a red border), "Database Service", "Infrastructure Service", and "Find team(s)". At the bottom of the interface, there are "Invocation Options" with three checkboxes: "Only allow invocation on unresolved incidents" (unchecked), "Allow invocation manually" (checked), and "Allow invocation from event orchestration" (checked). In the bottom right corner of the interface, there 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  
People who have done the pre-work  
-pd\_api\_key keys/project/PDOT-Sydney/<subdomain>  
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	TYPE	Runbook Automation
PDOT Runner (created via TF)	STATUS	

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_api_key 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"/> <b>Code Rollback on System Failure</b> 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**

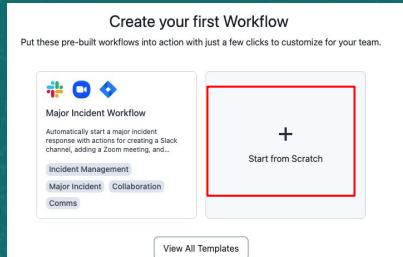
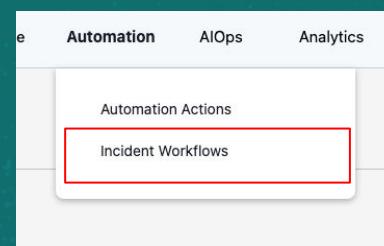
## Do this

Let's build the diagnostic job into the incident as a workflow (Step 1)

→ Goto “Automation” → “Incident Workflows”

→ Click “Start from Scratch” and name your workflow “`<username>` Auto-diagnostic” and “Create”

## See this

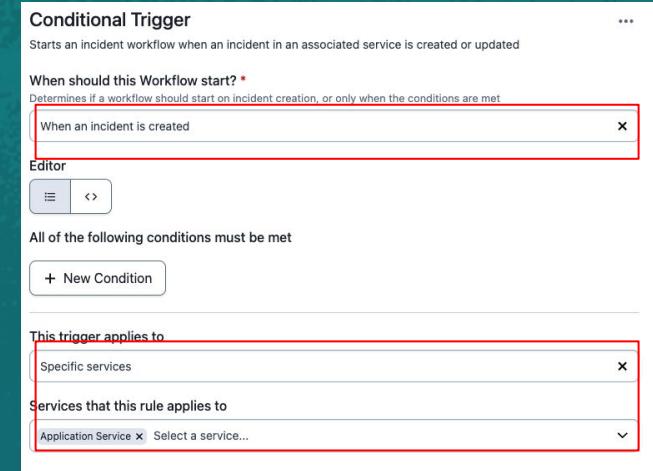
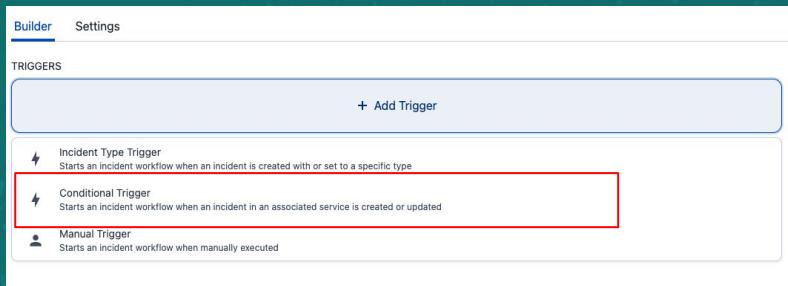
A screenshot of a "Create Incident Workflow" dialog box. It has fields for "Name" (containing "Auto-diagnostic workflow"), "Description" (with a note about help text), and "Who can edit this Workflow" (set to "All admins and global managers"). At the bottom are "Cancel" and "Create" buttons, with the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

## Do this

Let's 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

Let's 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 are two sections: "PagerDuty Incident Management" (16 actions) and "Roles" (2 actions). Below these, a list of actions is shown:

- Run an Automation Action: Run scripts installed on your infrastructure through PagerDuty Automation or Process Automation. This action is selected, showing its details: "Automation Action \* Select your automation action from the list." The option "Diagnostic Script for System Failure" is highlighted.
- Send Status Update: Post an update to the internal status page and notify subscribers.

At the bottom of the builder, there are "Remove", "Cancel", and "Save" buttons.

# PAUSE

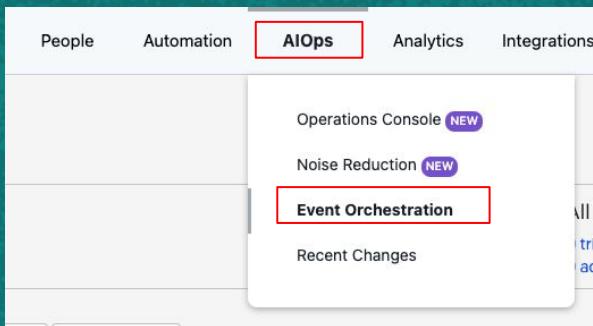


PagerDuty **ON TOUR**

## 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”

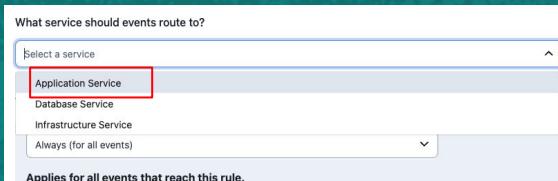
## See this

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".

## 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?

If events match certain conditions

Always (for all events)

✓ If events match certain conditions

On a recurring weekly schedule

During a scheduled date range

Depends on event frequency

IF      event.summary      matches part (conta▼)      system failure

## 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 user 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 (with a red box around the 'Copy to Clipboard' button)
- HTTP Endpoint for API: https://events.pagerduty.com/v2/enqueue (with a 'Copy to Clipboard' button)
- Email Address: R028GKC6ZJ3PIFPUKX173EO2O0T16YDR@dev-kotsukajelidev.pagerduty.com (with a 'Copy to Clipboard' button)

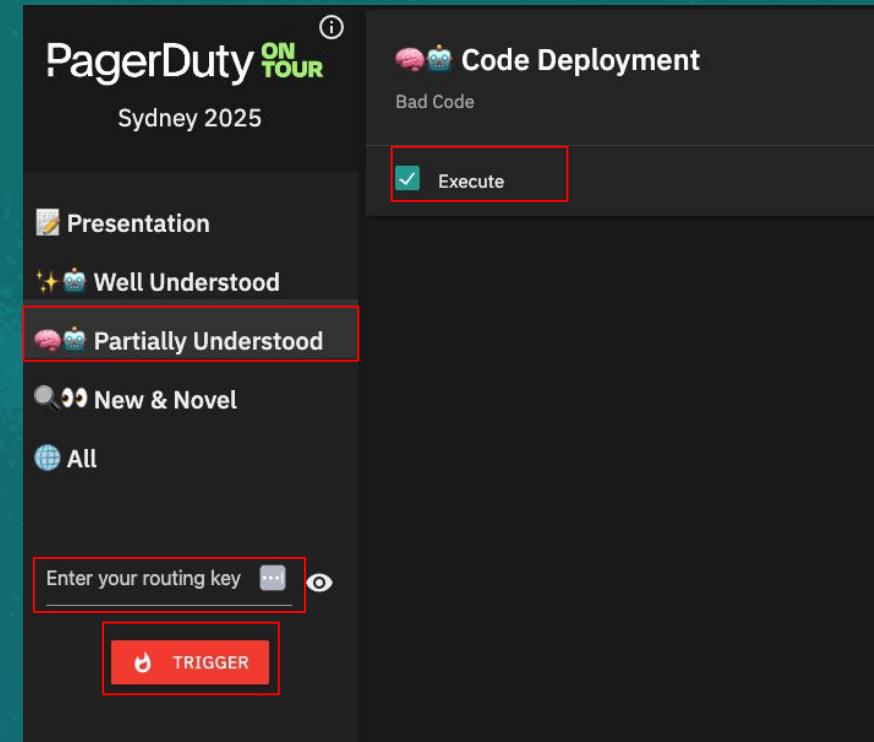
At the bottom of the integration card is another blue '+ New Integration' button.

## 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 is a sidebar titled 'RESPONSE' with options: 'All Incidents' (which is highlighted with a red box), 'Alerts', and 'Visibility Console'. The main content area is titled 'Incident' and displays summary statistics: 'Your open incidents' (0 triggered, 0 acknowledged). At the bottom of the sidebar, there's a link 'Jedi-Doc Incident Review'. The main table lists incidents with columns for Status (Triggered), Priority (High), Urgency (High), Type (Base Incident), and Title ('Unexpected System Failure Detected'). A red box highlights the 'Title' column for the single listed incident. There are also buttons for 'SHOW DETAILS' and '1 triggered alert'.

Status	Priority	Urgency	Type	Title
Triggered	--	High	Base Incident	Unexpected System Failure Detected SHOW DETAILS (1 triggered alert)

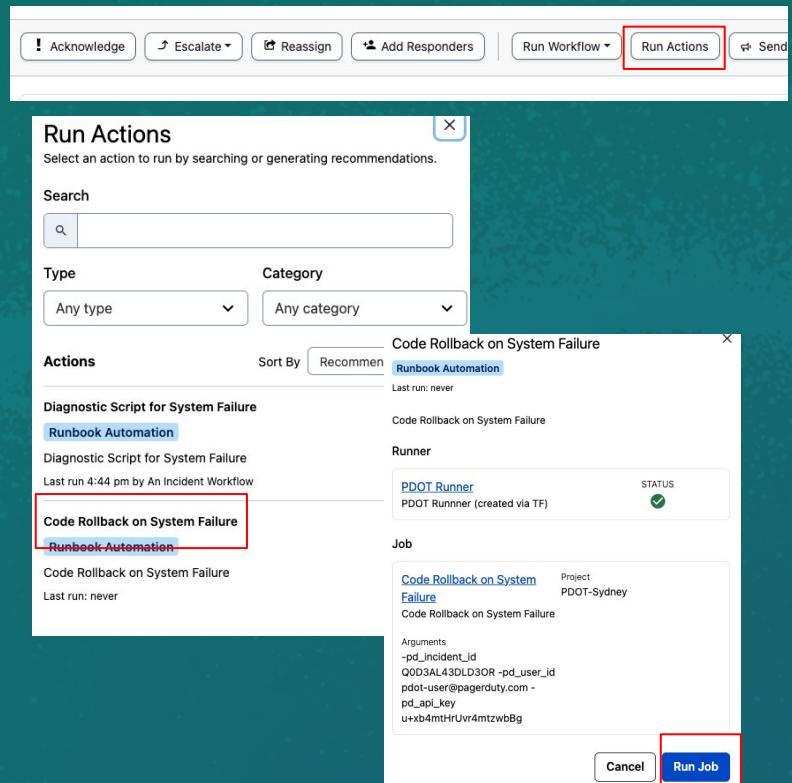
## Do this

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 interface. 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 content area displays incident details:

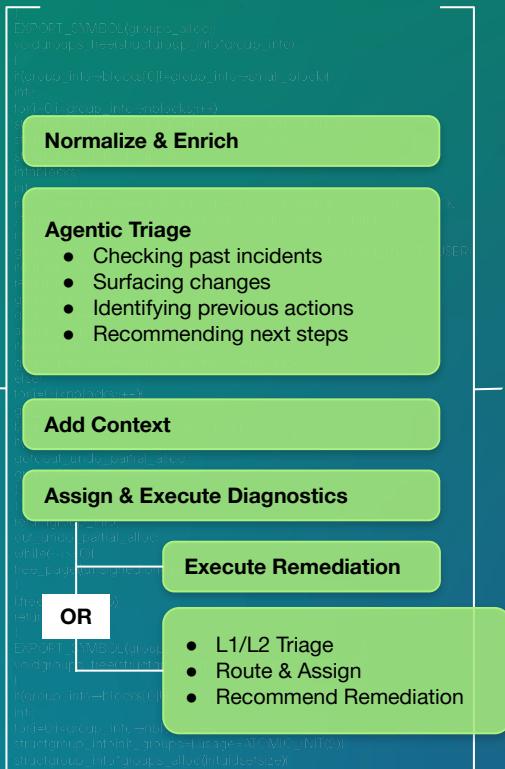
- STATUS:** Resolved
- INCIDENT TIMES:** Open from 9:08 AM to 9:08 AM (for a few seconds)
- URGENCY:** High
- MAIN SERVICE:** Infrastructure Service
- IMPACTED SERVICE:** Infrastructure Service
- SERVICE DESCRIPTION:** Managed by Terraform

  
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 "dot-user" at 9:08am. It includes "Health Check Details" and a "Payload" object containing a "script" for restarting a Kubernetes pod.  
  
At the bottom, there's a "Automation Actions Log" tab showing a single entry for a "Kubernetes Pod CrashLoopBackOff" automation action. The log entries show the script running and a successful restart of the pod.  
  
Page navigation at the bottom right indicates "5 per page" and "1 - 1".

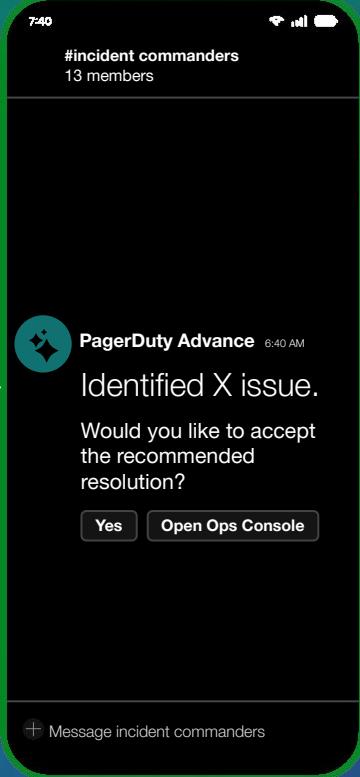


## Partially understood incident

Events  
CMDB Data  
Customer Incidents  
Telemetry  
Diagnostics  
Logs



## Post Incident Learning



### Human Response



**PagerDuty** **ON TOUR**

# Scenario



## New & novel

Brand new incidents, or incidents requiring expert attention



Responder-led +  
AI & Automation

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

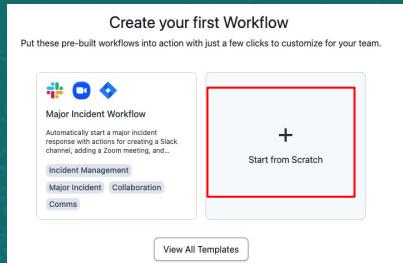
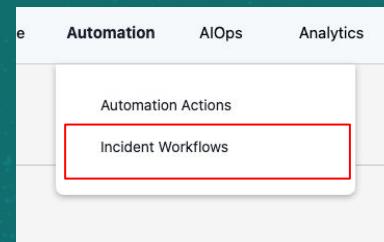
## Do this

Let's build a war room scenario process as a workflow (Step 1)

→ Goto “Automation” → “Incident Workflows”

→ Click “Start from Scratch” and name your workflow “`<username> War Room`” and “Create”

## See this

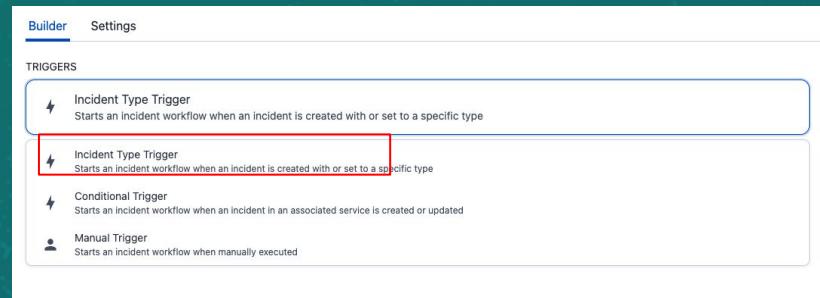
A screenshot of a "Create Incident Workflow" dialog box. It includes fields for "Name" (containing "War Room workflow"), "Description" (with a note "Help users understand what the workflow does and how it should be used."), and "Who can edit this Workflow?" (set to "All admins and global managers"). At the bottom right, there are "Cancel" and "Create" buttons, with the "Create" button highlighted with a red box.

## Do this

Let's 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

Let's 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

Let's 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 form titled "Create Incident". The fields are as follows:

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

At the bottom right are two buttons: "Cancel" and "Create Incident" (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 'Major Incident' details page for a 'New Service Outage'. The incident was triggered at 1:52 PM (a minute ago) and is currently 'Open'. It is assigned to 'pdot-user' and has an escalation policy for 'ep-user' with a conference number '+61 404370845' and URL 'https://example.meeting.com'. The incident key is 'dcf521105fa14a95935e25529fce84f0'. The main service is 'Database Service' and the impacted service is also 'Database Service'. The service description is '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 'Responders' section shows 'pdot-user' as assigned, 'Unassigned Incident Commander', and 'Unassigned Customer Liaison'. The 'Status Updates' tab is active, showing a status update from 'pdot-user' at 1:52 PM. The 'Subscribers' section shows 1 team and 0 users subscribed. The 'Status Dashboard' indicates the incident is not visible on the status dashboard. Buttons for 'Add Responders' and 'Assign Roles' are visible at the bottom right.

# PAUSE and Demo PD Advance and Jeli

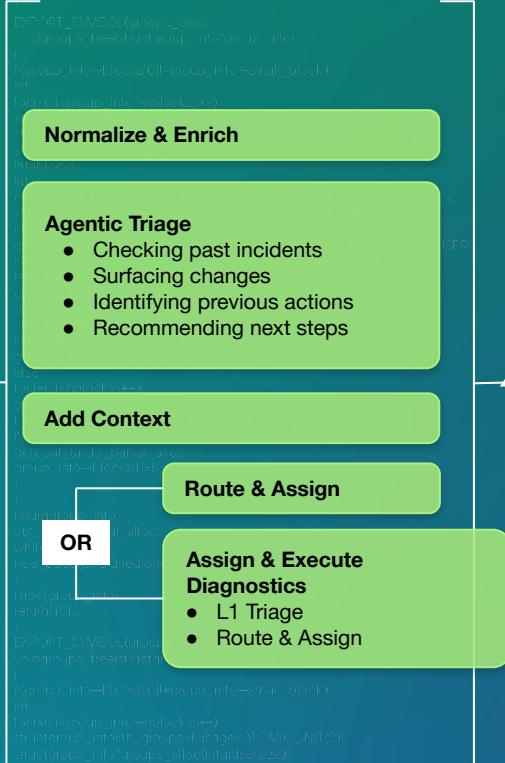


PagerDuty **ON TOUR**

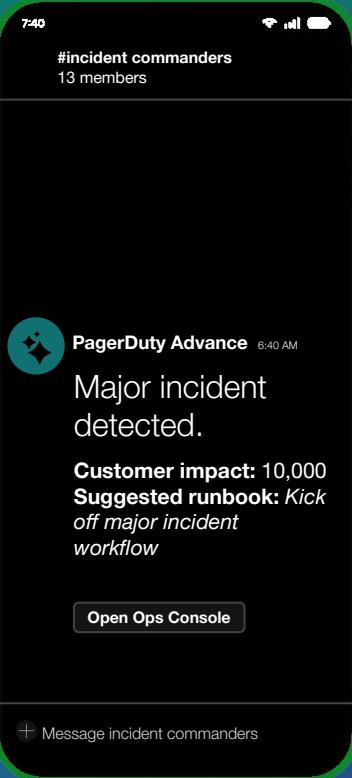


## New, novel, major incident

Events  
CMDB Data  
Customer Incidents  
Telemetry  
Diagnostics  
Logs



Push Incident Detection



**Human Response**

- Mobilize teams
- Response
- Guided Incident Remediation
- Maintain Vital Comms



PagerDuty **ON TOUR**

# Thank You!

PagerDuty **ON TOUR**