PagerDuty ON TOUR

To-Do: Getting ready

AP: Doltone House Events

Password: DHCorpClient

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$"

PagerDuty ON TOUR

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

PagerDuty ON TOUR

Before we Begin

PagerDuty ON TOUR

When you see this - People need to catch up!

PAUSE

PagerDuty ON TOUR

Do this

SKIPPING ONBOARDING

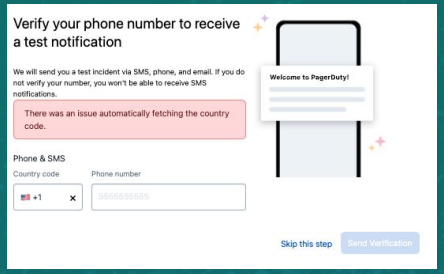
➡️ **SKIP THIS STEP**

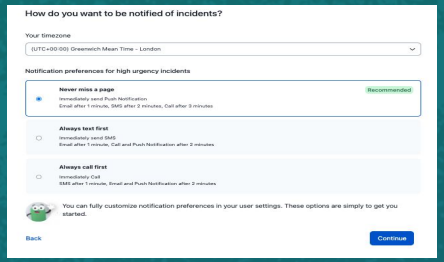
➡️ **NEVER MISS A PAGE**

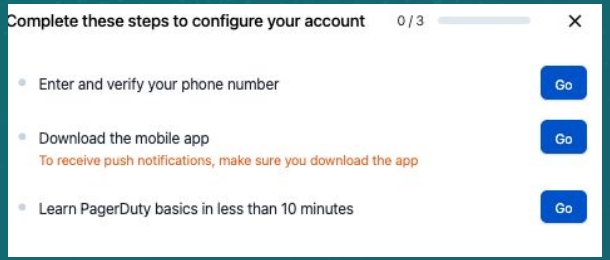
➡️ **CLOSE THE ONBOARDING STEPS**

CONTINUE TO THE MAIN INCIDENT PAGE

See this







Let’s get started!

PagerDuty ON TOUR

1. Well Understood

100% AI and Automation

PagerDuty ON TOUR

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

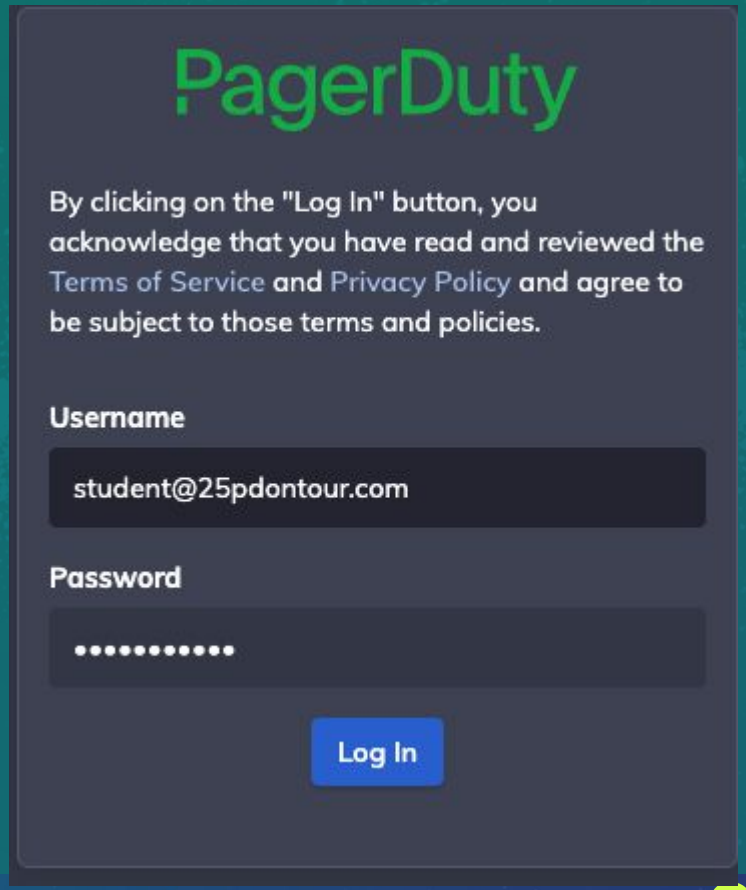
pdot-xxxx

User Name

pdot@sydney$

Hit Log In to join

See this



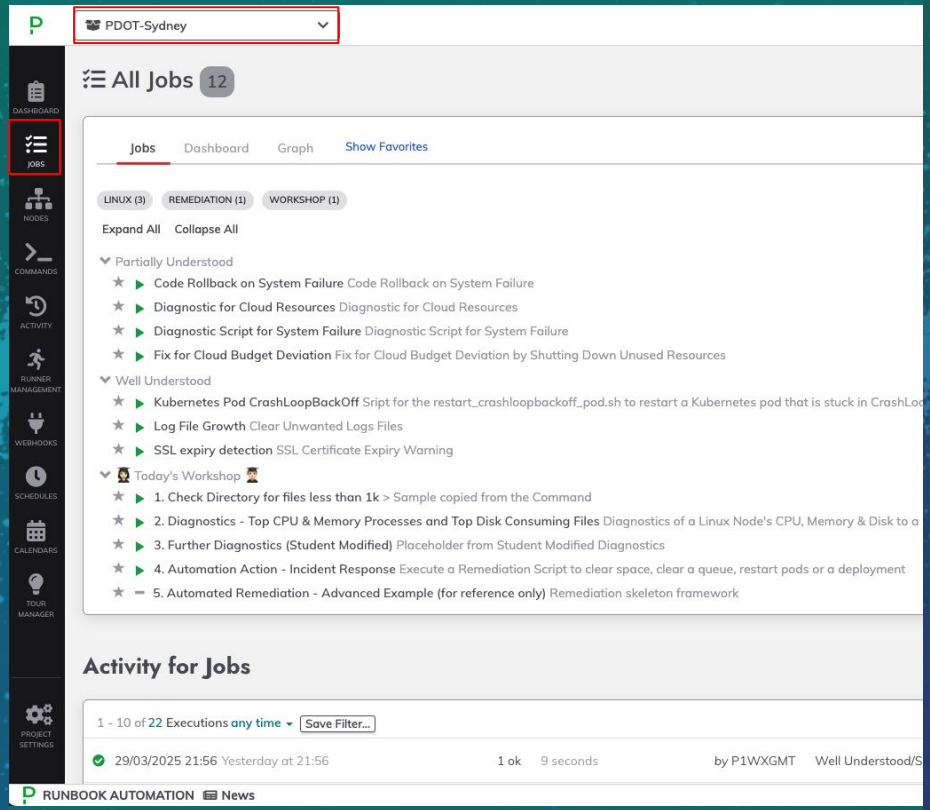
Do this

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**

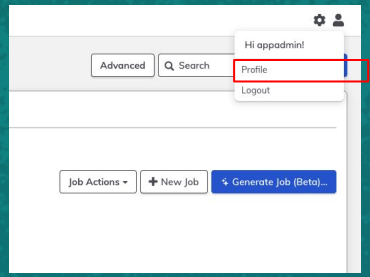
See this



Do this

Lets connect these runbooks to PagerDuty (Step 1)

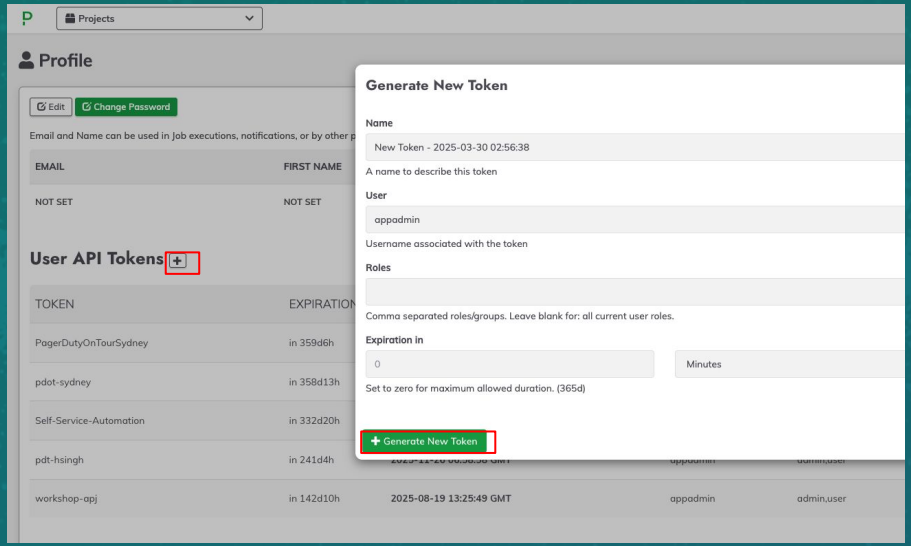
➡️**Click your profile silhouette and access “Profile”**



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

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

See this



Do this

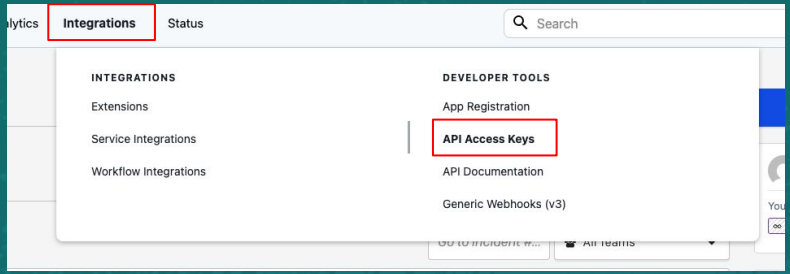
Lets connect PagerDuty to these Runbooks (Step 1)

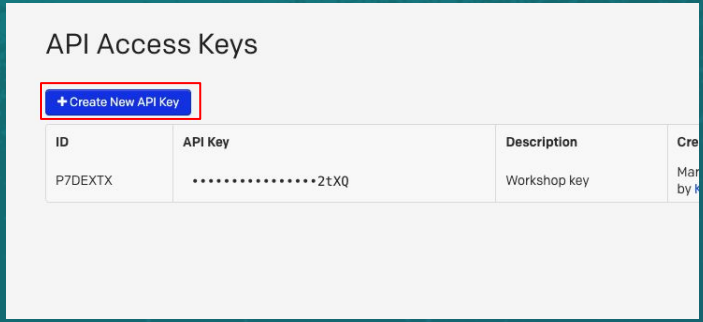
➡️**Switch to your PagerDuty account and goto “Integrations” → “API Access Key”**

➡️ **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

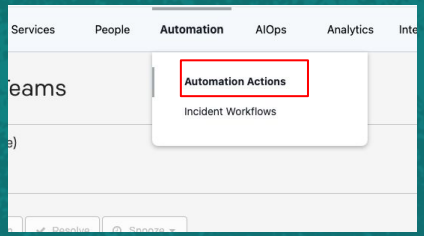




Do this

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

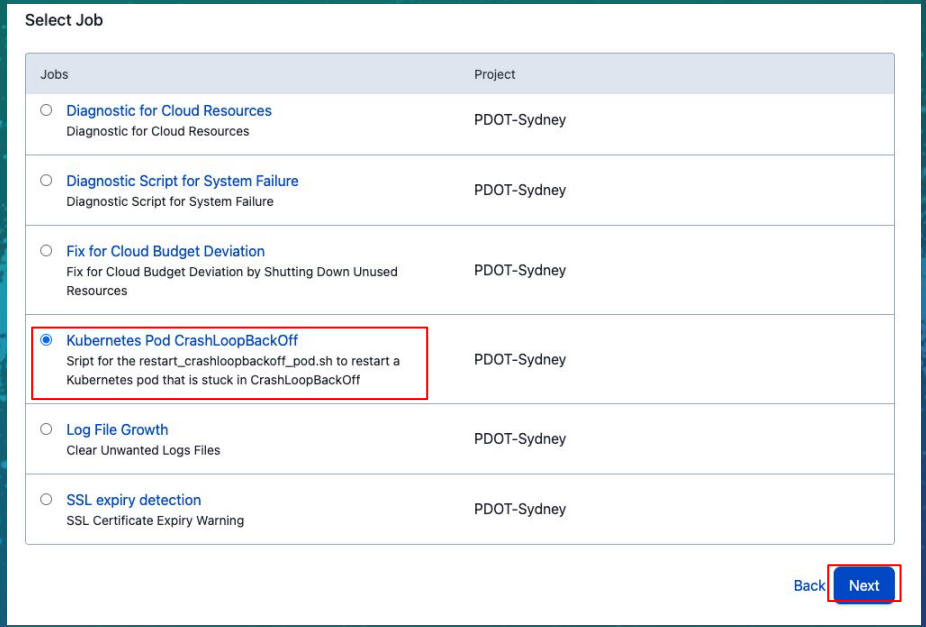
➡️ **On PagerDuty goto "Automation" → "Automation Actions"**



➡️ **Click on the "Add Action" button and select the "PDOT Runner"**

➡️ **Select the "Kubernetes Pod CrashLoopBackOff" job and click next.**

See this



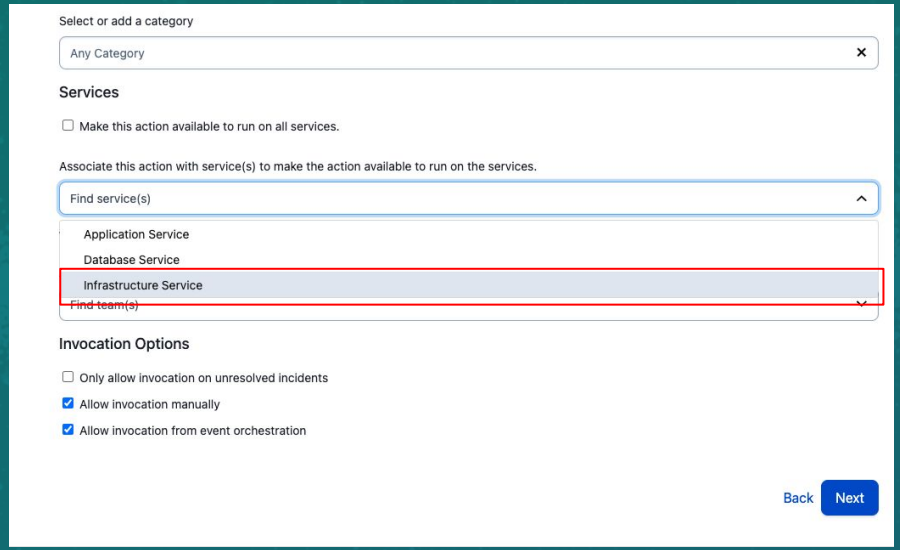
Do this

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



Do this

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

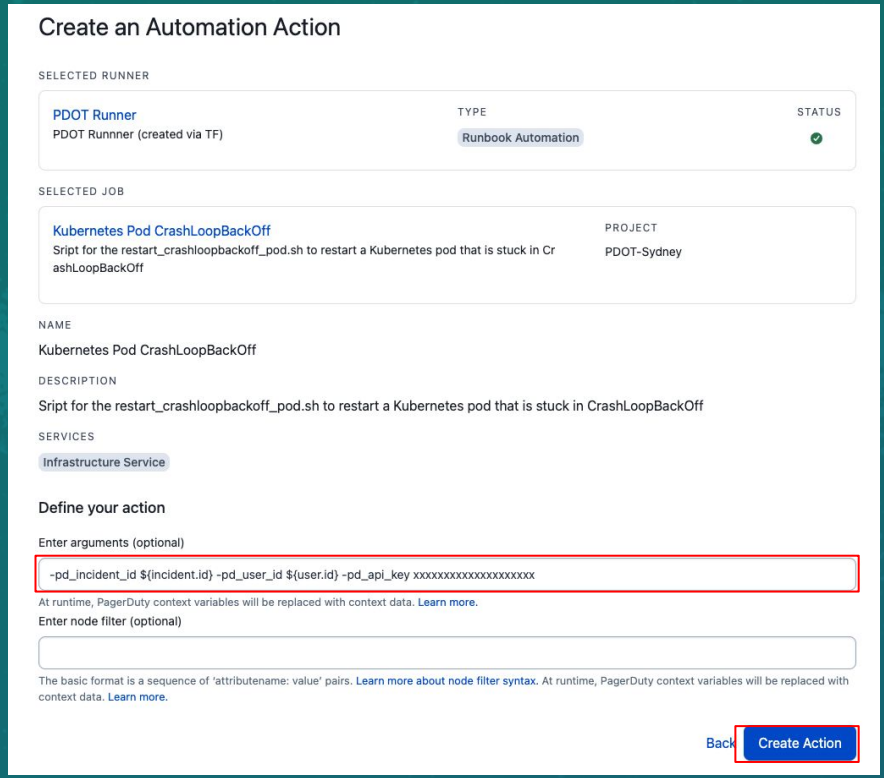
➡️ **Scroll down to the "Enter argument (optional)" and enter below with spaces in between.**

-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



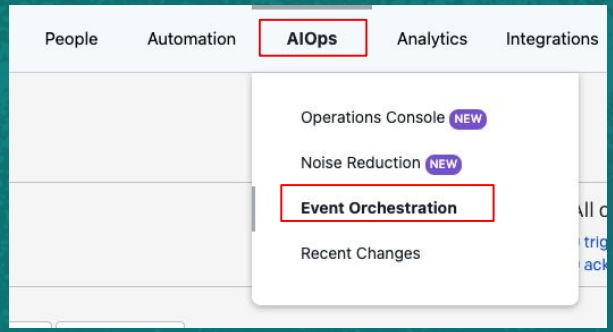
PAUSE

PagerDuty ON TOUR

Do this

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

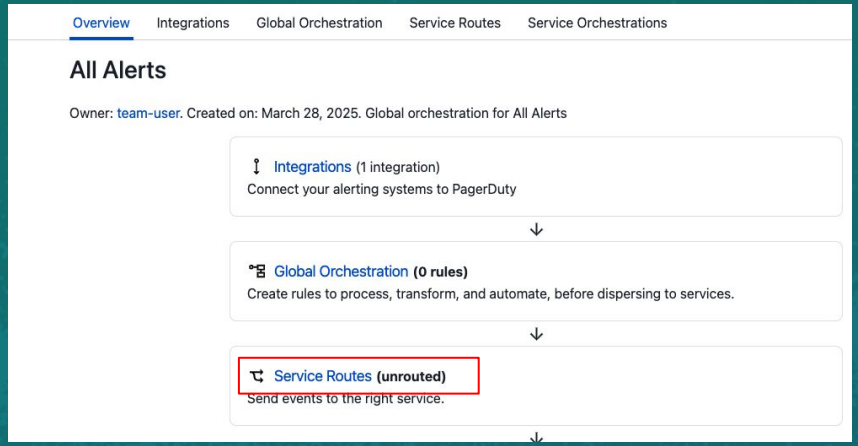
➡️ **Goto "AIOps" → "Event Orchestration"**

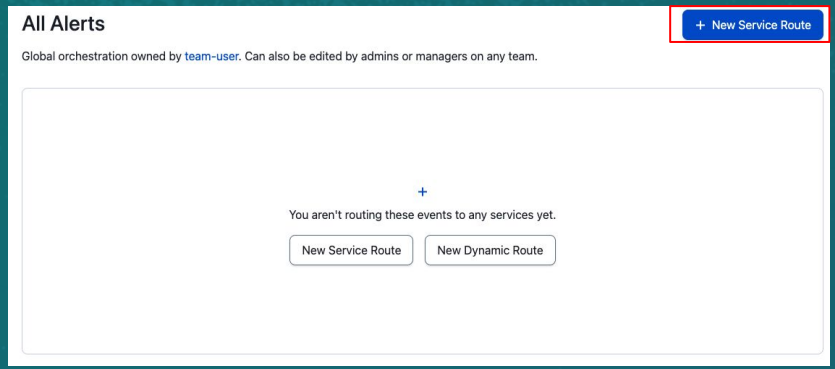


➡️ **Select the "All Alerts" orchestration and go into "Service Routes"**

➡️ **Click on "New Service Route"**

See this

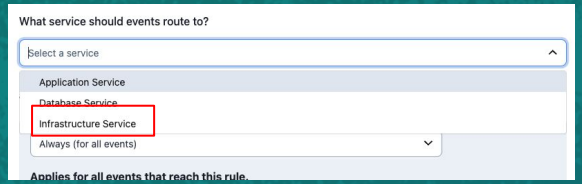




Do this

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

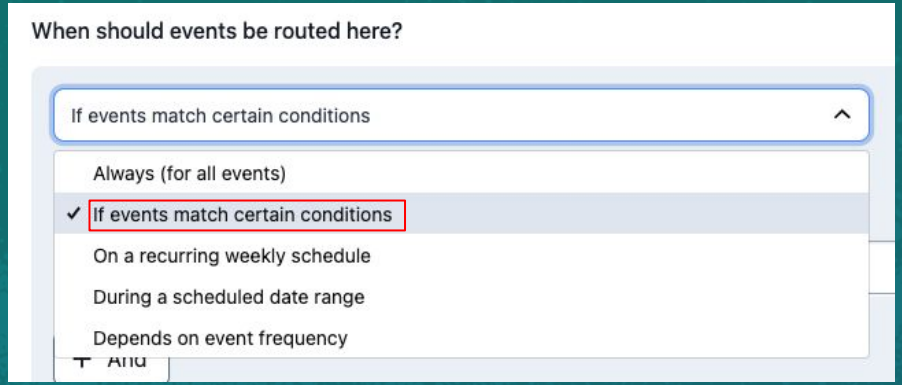
➡️ **For “What service should events route to?” choose “Infrastructure Service”**

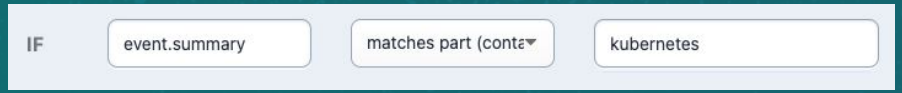


➡️ **For “When should events be routed here?” Select “If events match certain conditions”**

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

See this





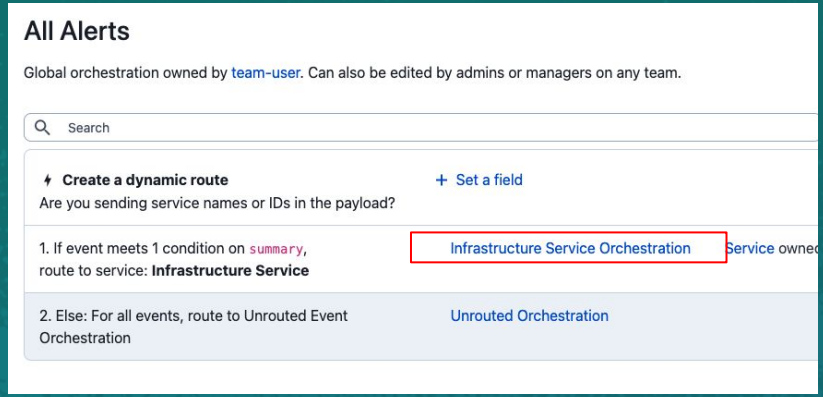
Do this

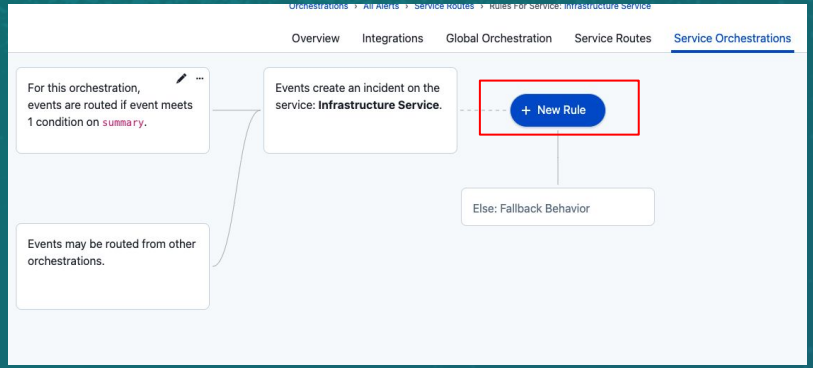
Lets orchestrate an alert to use the Automation Action (Step 3)

➡️ **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





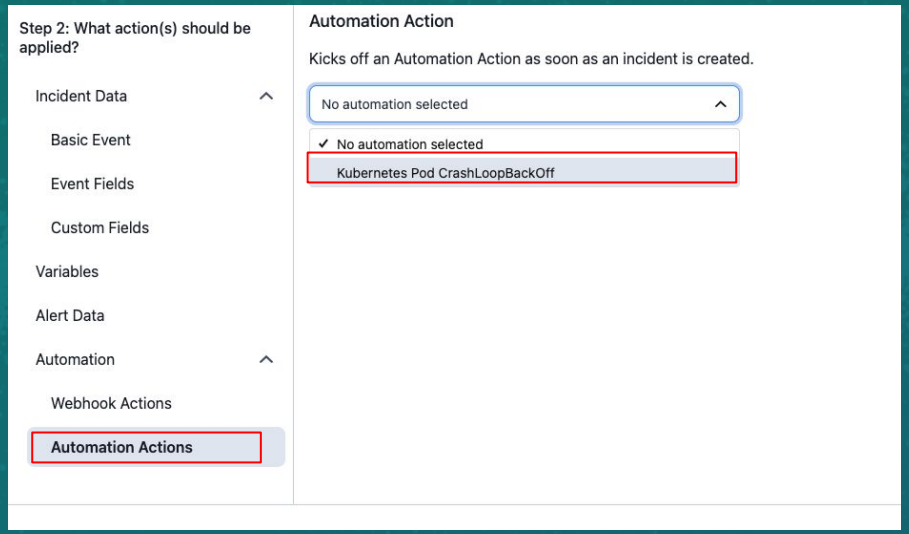
Do this

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

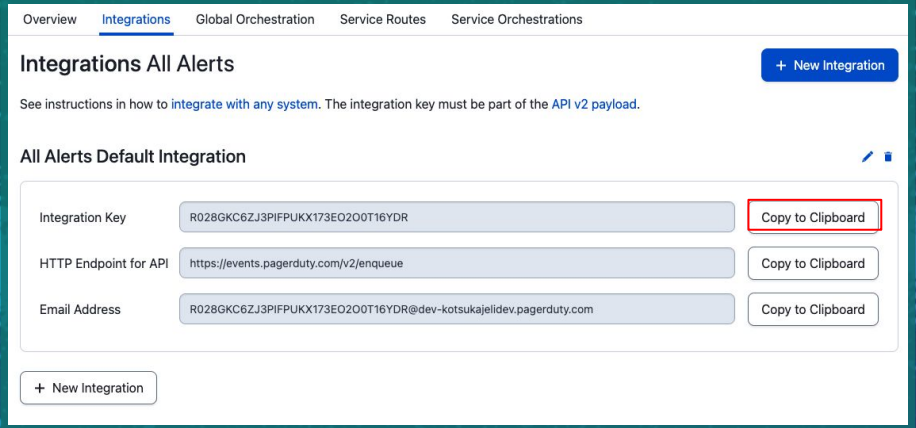


Do this

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



Do this

Lets simulate the alert! (Step 2)

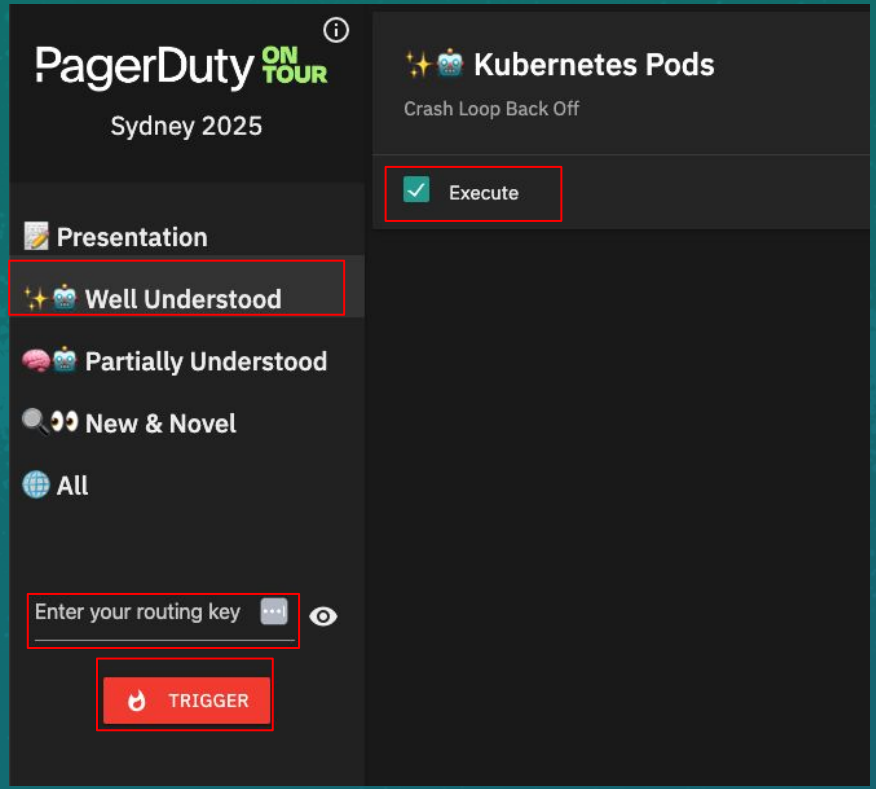
➡️ **Open a new browser tab and goto “https://press4ack.com/”**

➡️ **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.

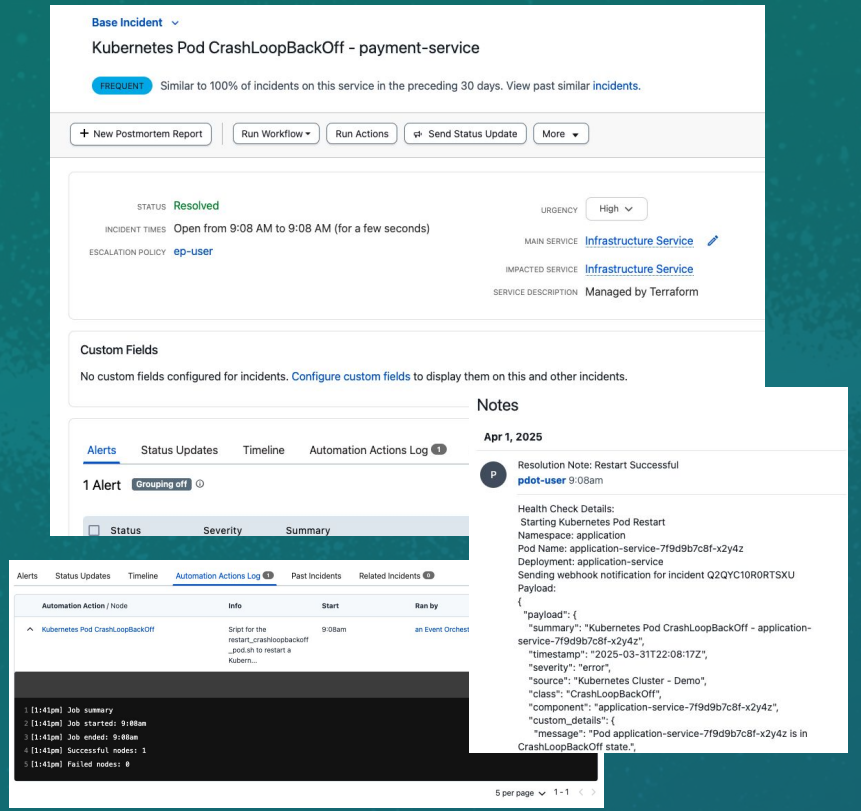
This of course is core Incident Response functionality.

Look at the incident resolve automatically!

Check the below to see what happened:

- Notes
- Timeline
- Automation Action Log

See this



2.Partially Understood

AI & Automation + Responder assisted

PagerDuty ON TOUR

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

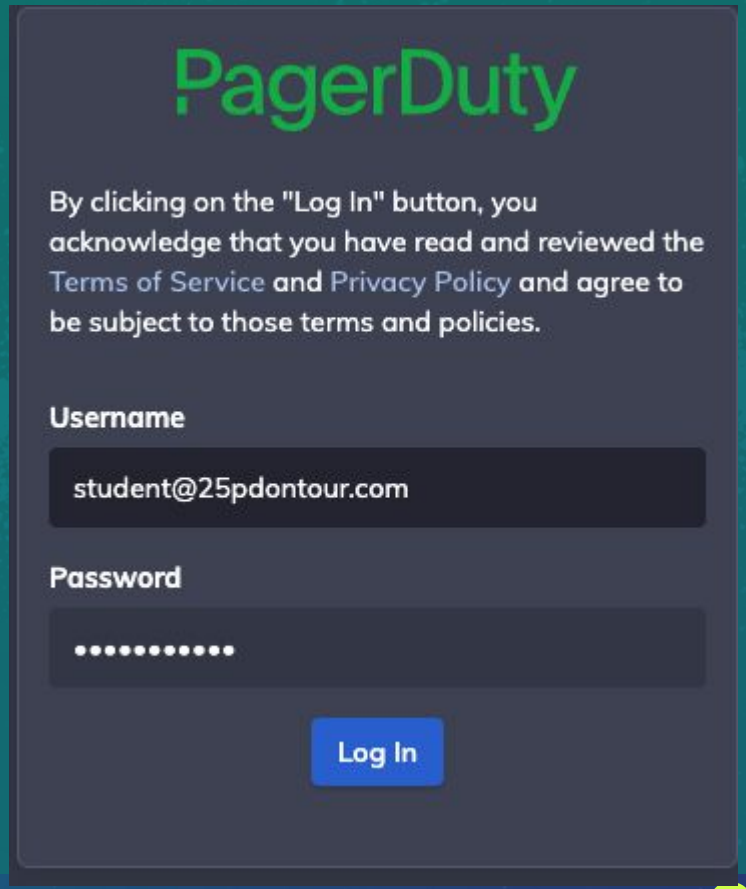
xxxxxx

User Name

pdot@sydney$

Hit Log In to join

See this



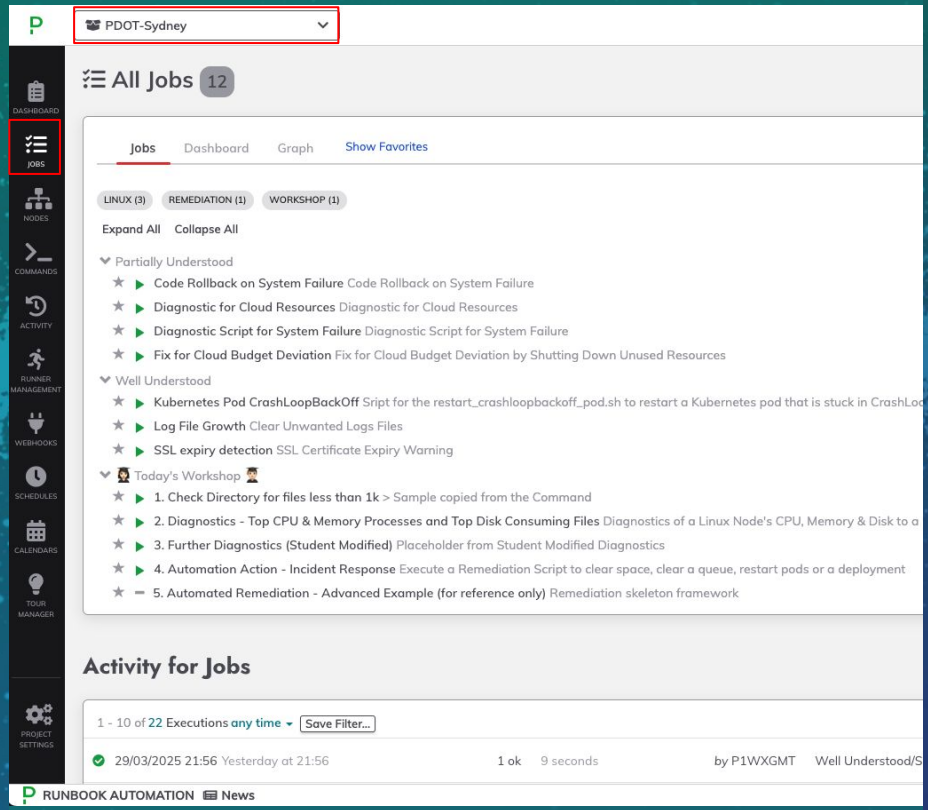
Do this

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



Do this

Lets connect PagerDuty to these Runbooks (Step 1)

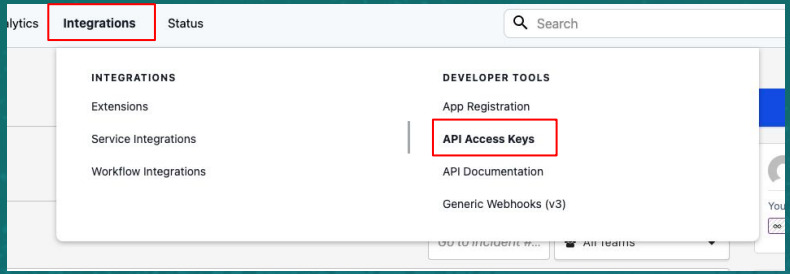
➡️**You can skip this step if you have the API key created on the "Well Understood" use case.**

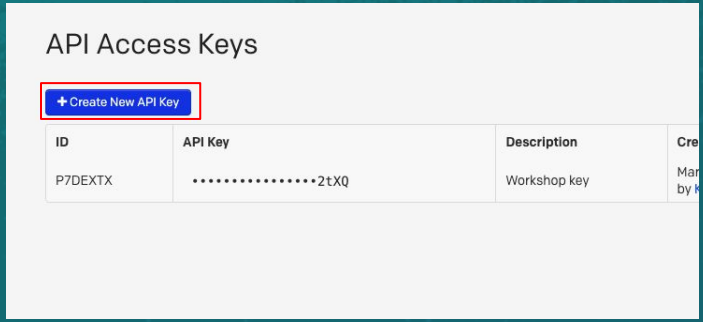
➡️**Switch to your PagerDuty account and goto "Integrations" → "API Access Key"**

➡️ **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

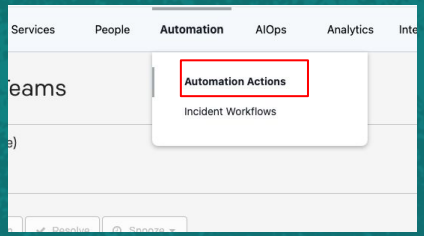




Do this

Lets connect PagerDuty to these Runbooks (Step 2)

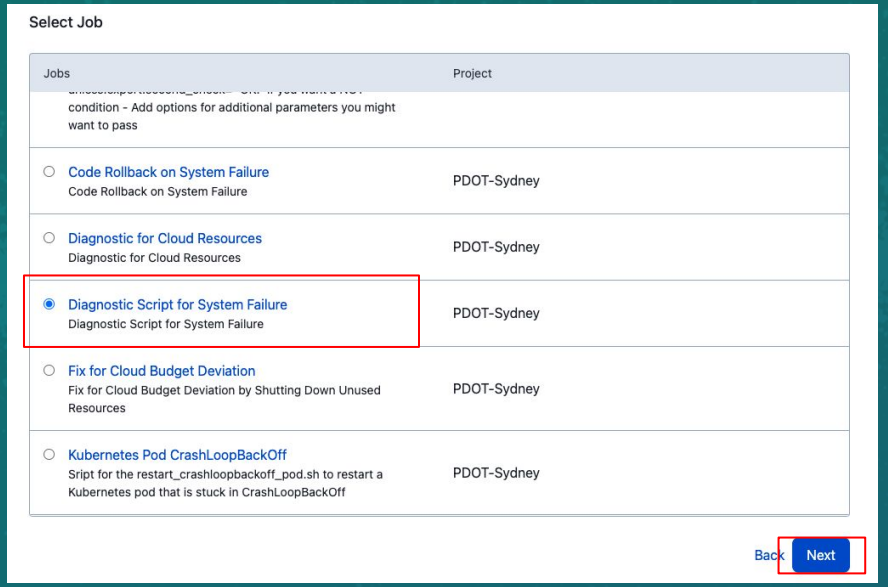
➡️ **Goto “Automation” → “Automation Actions”**



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

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

See this



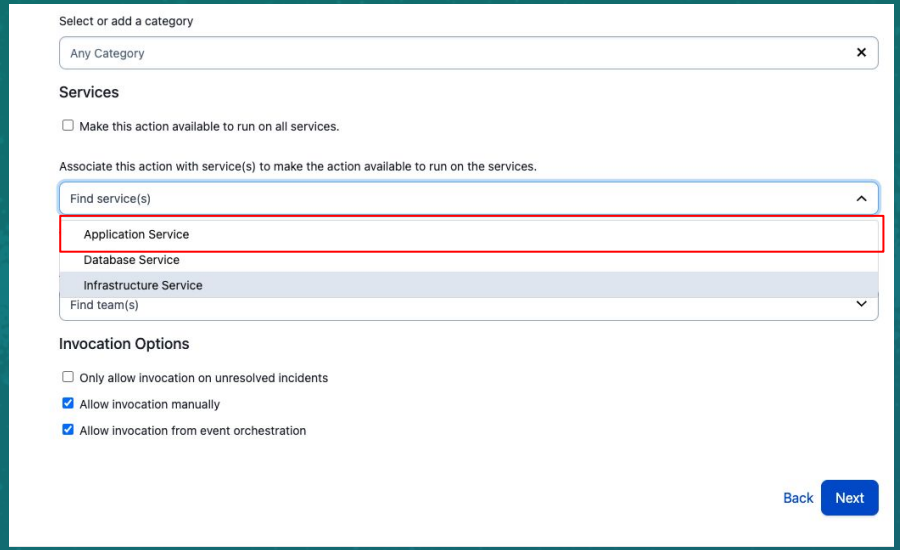
Do this

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



Do this

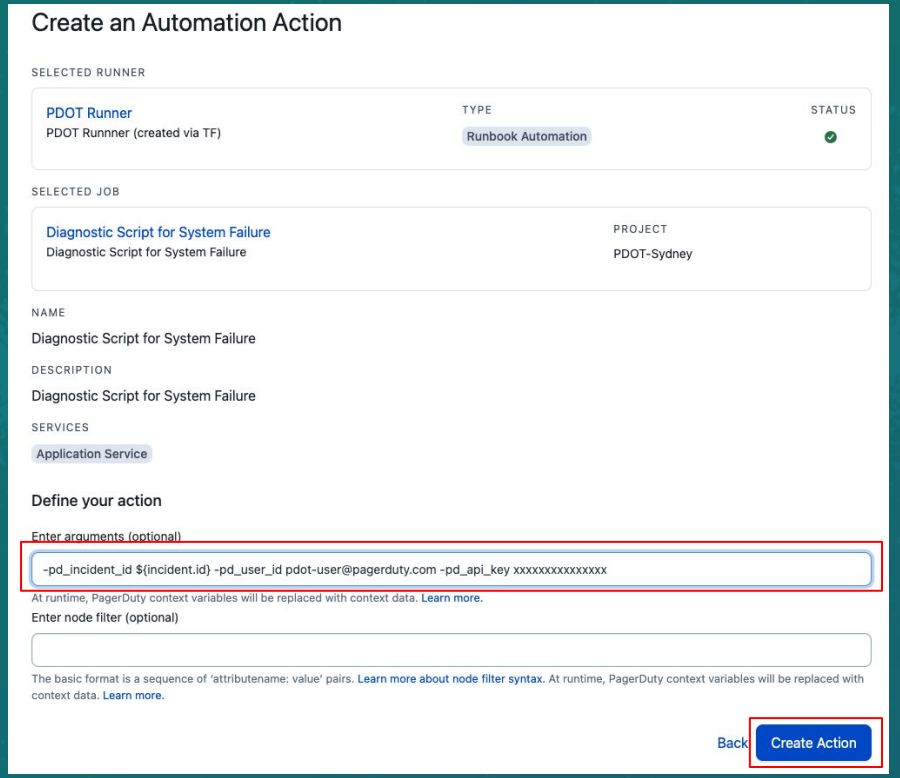
Lets connect PagerDuty to these Runbooks (Step 4)

➡️ **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

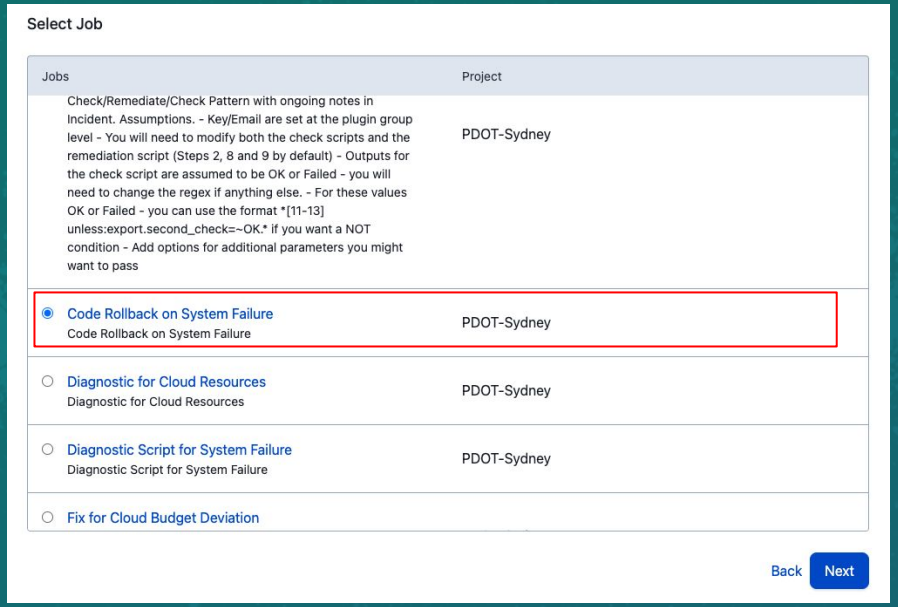


Do this

Lets connect PagerDuty to these Runbooks (Step 5)

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

See this



PAUSE

PagerDuty ON TOUR

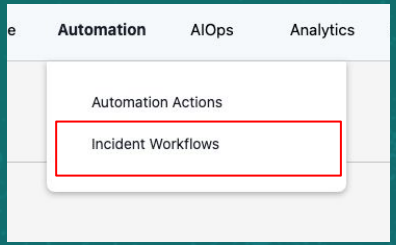
Do this

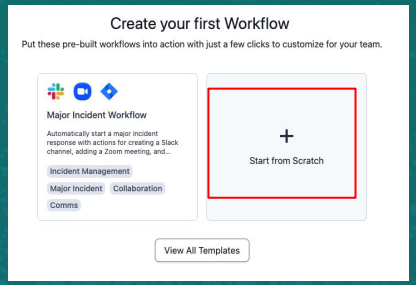
Lets build the diagnostic job into the incident as a workflow (Step 1)

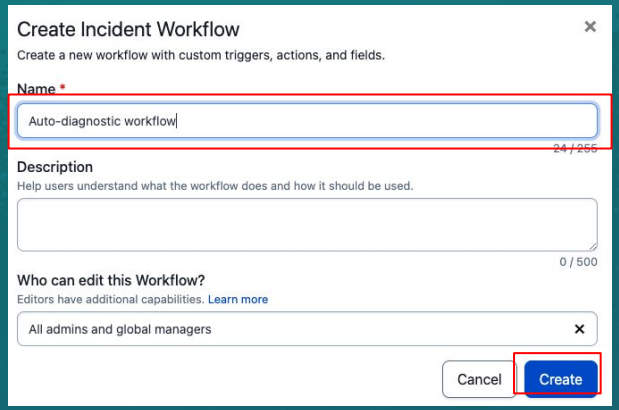
➡️ **Goto “Automation” → “Incident Workflows”**

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

See this







Do this

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

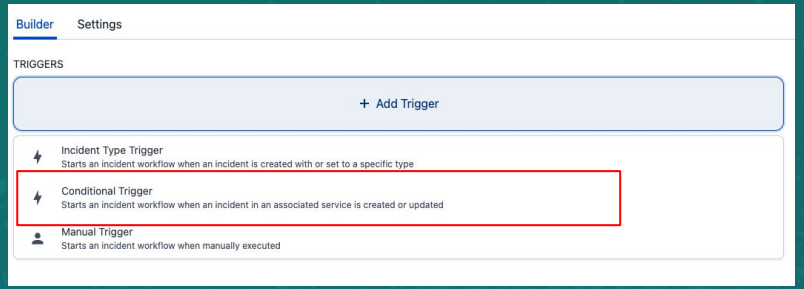
➡️ **In the builder click “+ Add Trigger” and select “Conditional Trigger”**

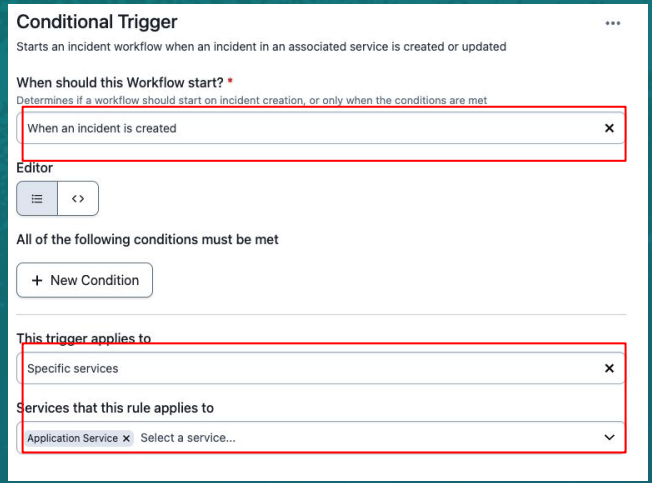
➡️ **On the right side of the editor, for “When should this Workflow start?” select “When an incident is created”**

➡️ **For “This trigger applies” to select “Specific services” and choose “Application Services”**

➡️ **Click “Save”**

See this





Do this

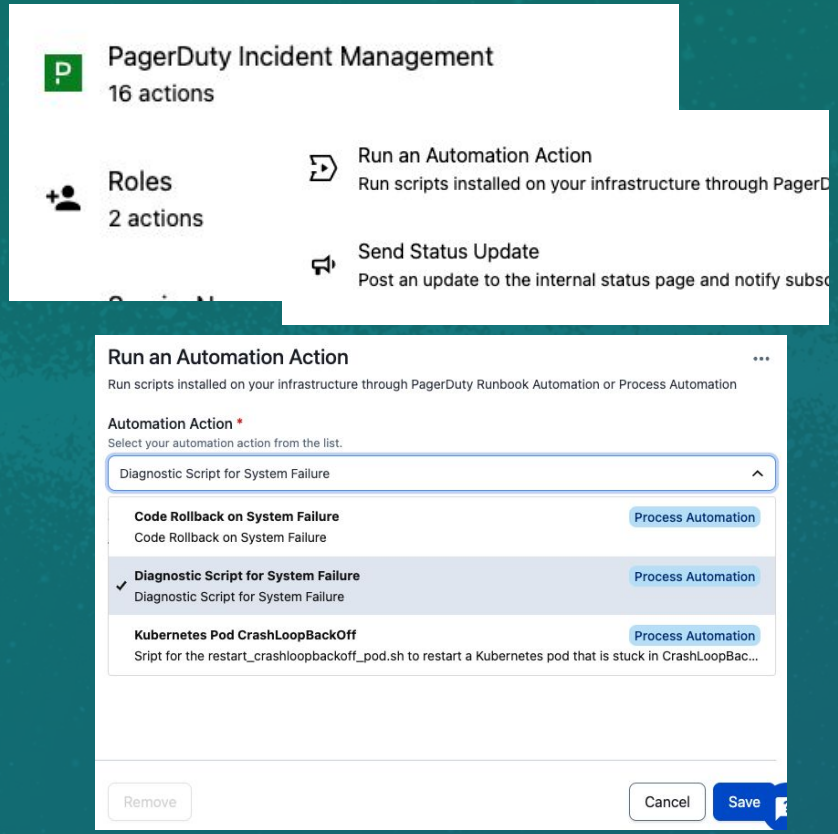
Lets build the diagnostic job into the incident as a workflow (Step 3)

➡️ **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



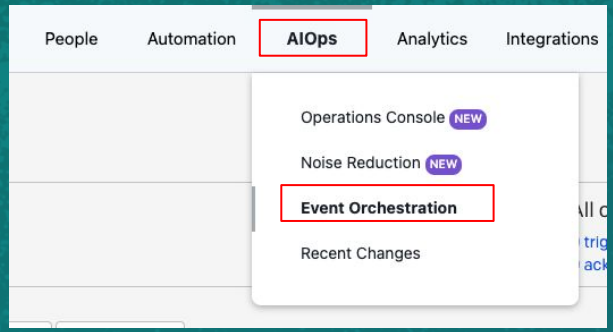
PAUSE

PagerDuty ON TOUR

Do this

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

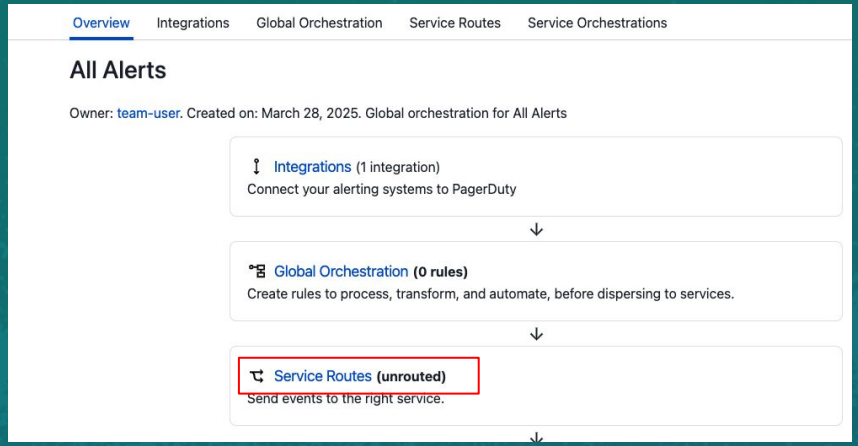
➡️ **Goto “AIOps” → “Event Orchestration”**

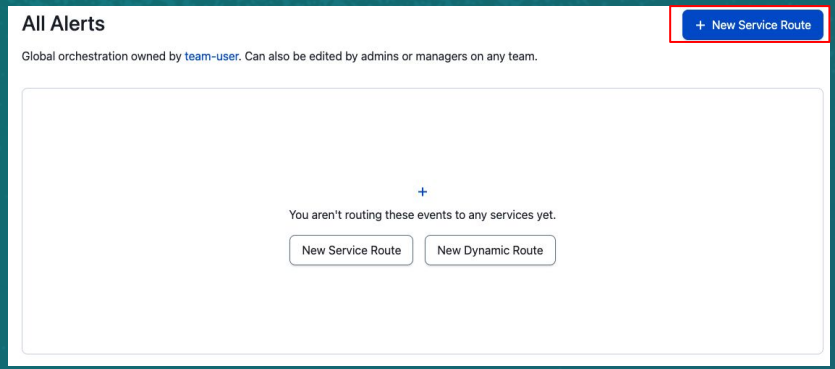


➡️ **Select the “All Alerts” orchestration and go into “Service Routes”**

➡️ **Click on “New Service Route”**

See this

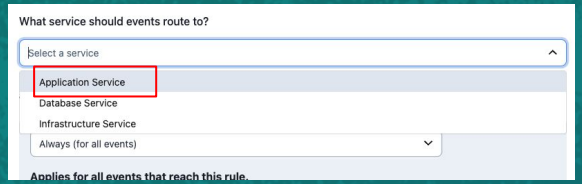




Do this

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

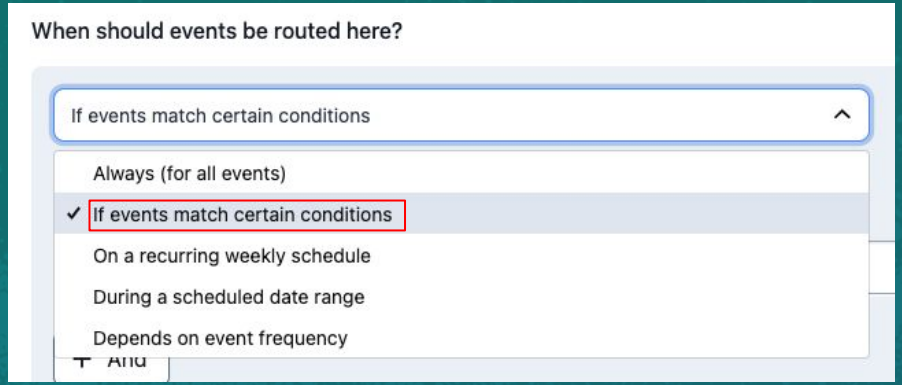
➡️ **For “What service should events route to?” choose “Application Service”**

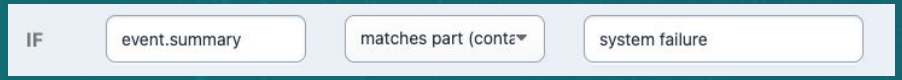


➡️ **For “When should events be routed here?” Select “If events match certain conditions”**

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

See this



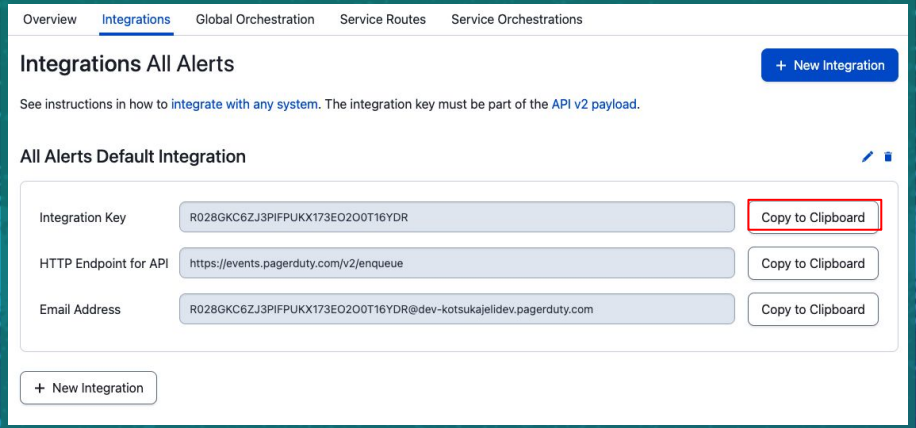


Do this

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



Do this

Lets simulate the alert! (Step 2)

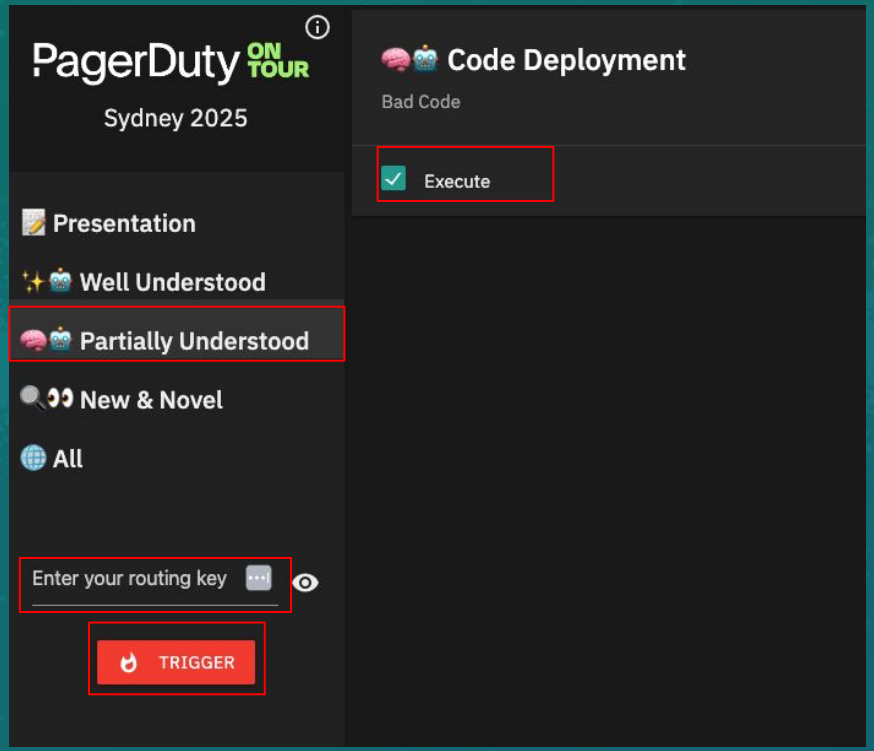
➡️ **Open a new browser tab and goto “https://press4ack.com/”**

➡️ **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

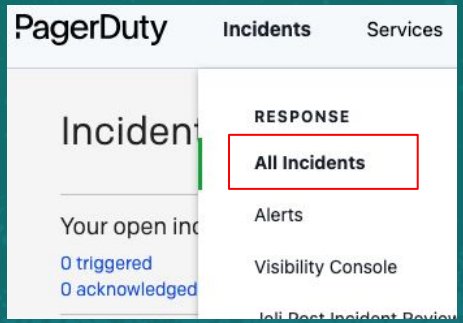
Lets simulate the alert! (Step 3)

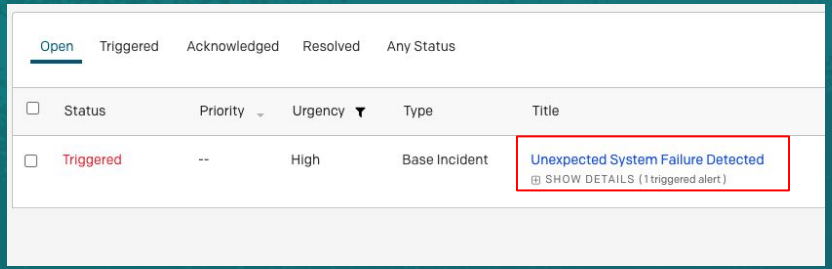
➡️ **On PagerDuty click on “Incidents” → “All Incidents”**

➡️ **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





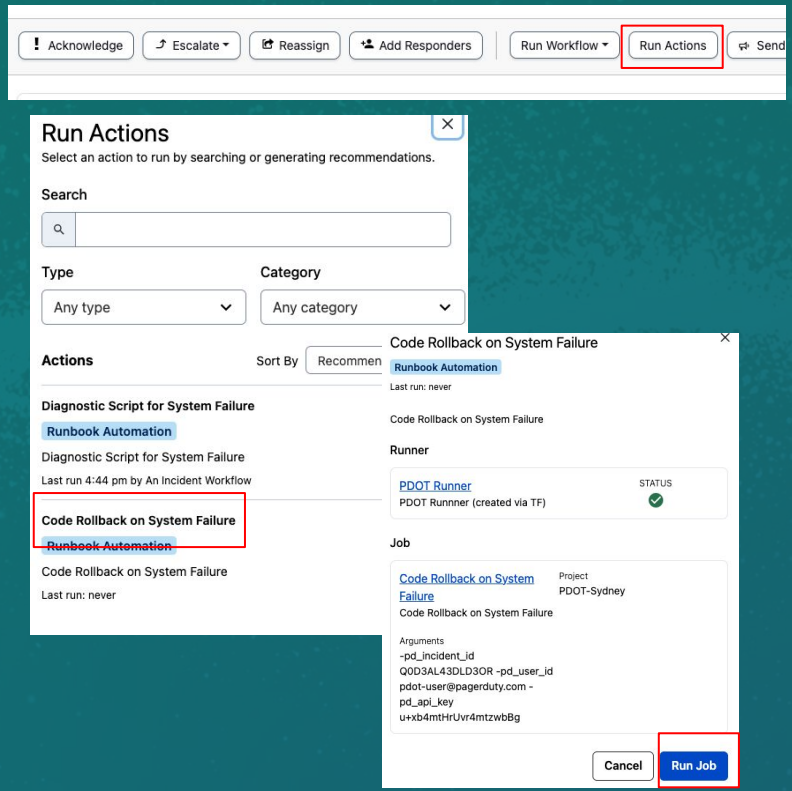
Do this

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.

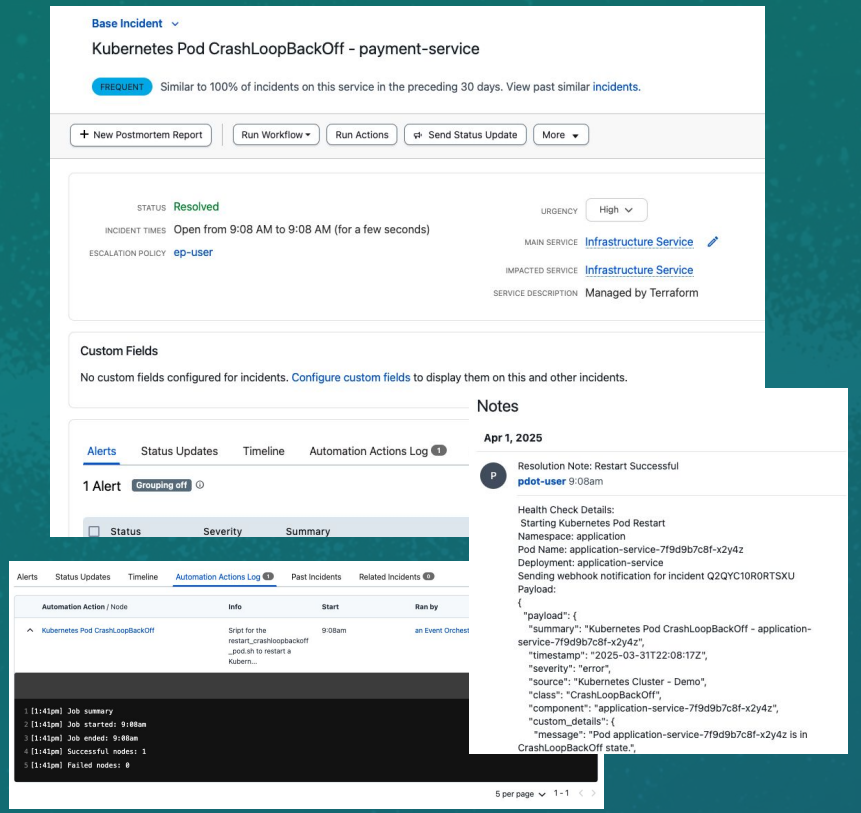
This of course is core Incident Response functionality.

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



3. New & Novel

Responder-led + AI & Automation

PagerDuty ON TOUR

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)

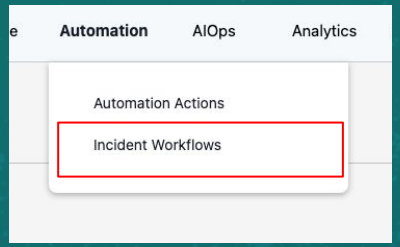
Do this

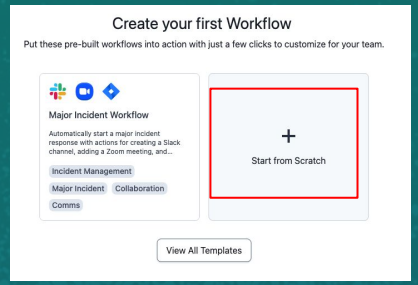
Lets build a war room scenario process as a workflow (Step 1)

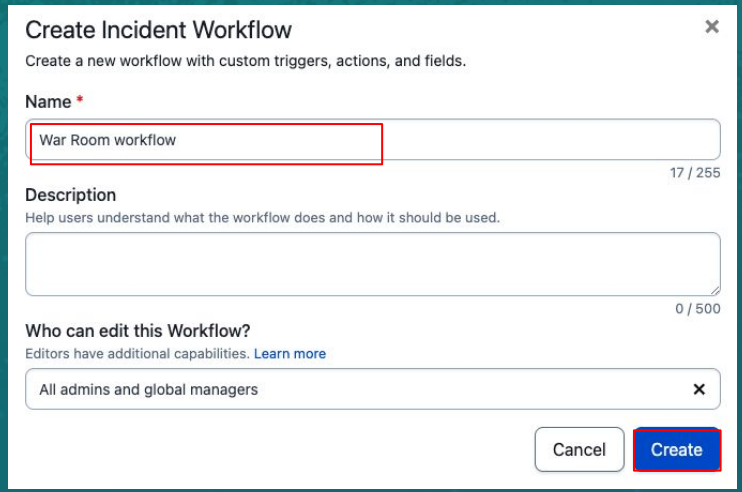
➡️ **Goto “Automation” → “Incident Workflows”**

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

See this







Do this

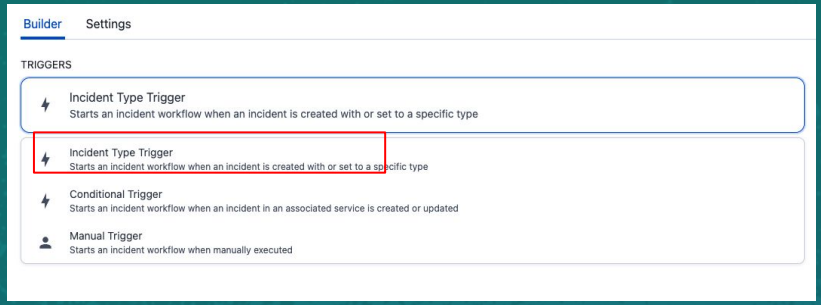
Lets build the diagnostic job into the incident as a workflow (Step 2)

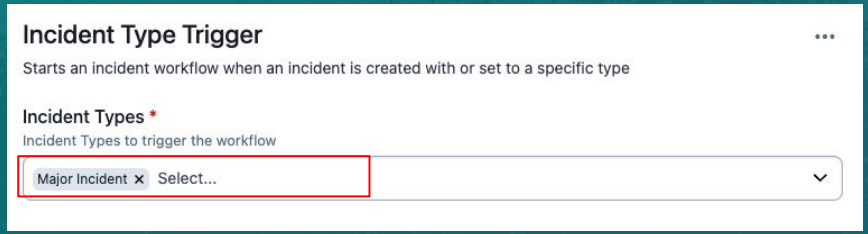
➡️ **In the builder click “+ Add Trigger” and select “Incident Trigger”**

➡️ **On the right side of the editor, for “Incident Types” select “Major Incident”**

➡️ **Click “Save”**

See this





Do this

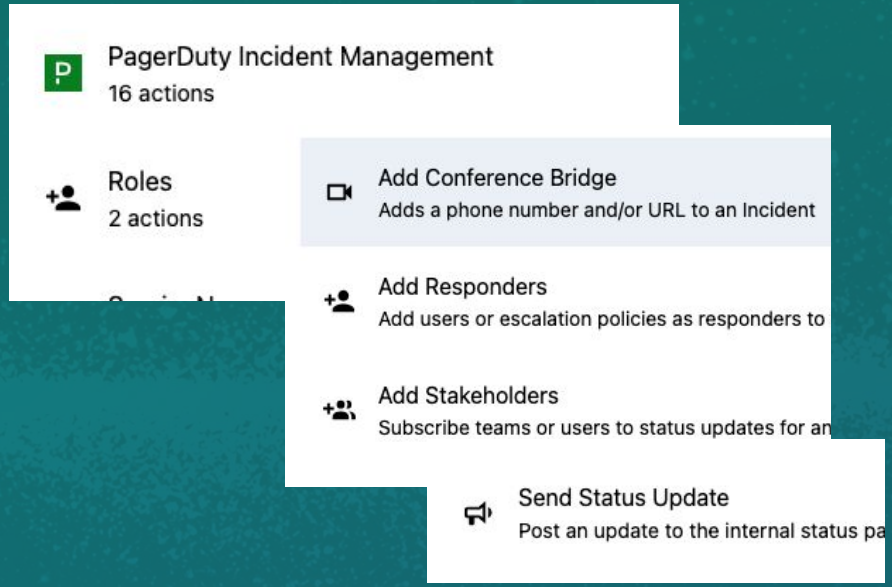
Lets build the diagnostic job into the incident as a workflow (Step 3)

➡️ **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



PAUSE

PagerDuty ON TOUR

Do this

Lets simulate the incident! (Step 1)

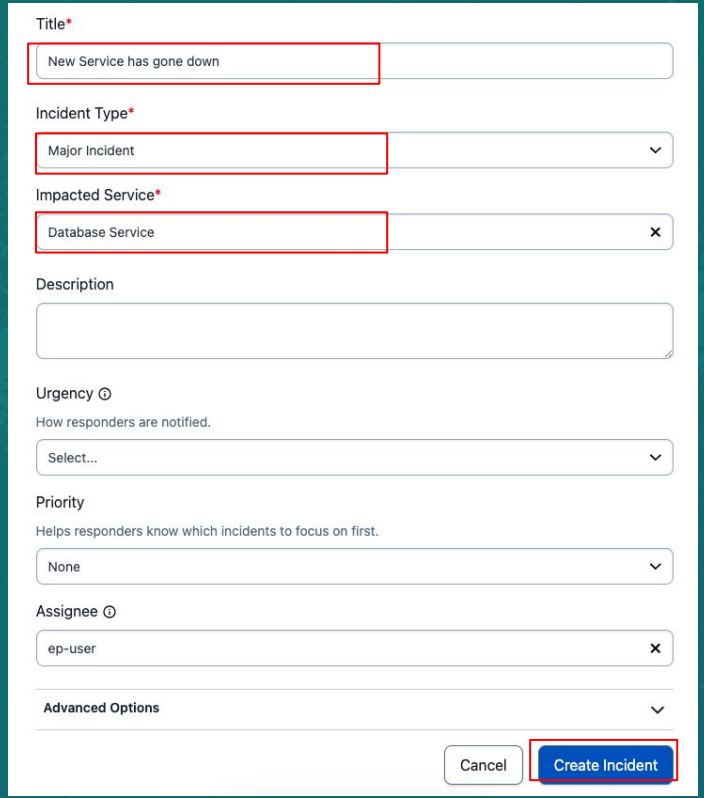
➡️ **Click on "+ New Incident" button and create a example incident. Give it a Title!**

➡️ **Select "Major Incident" as the Incident Type**

➡️ **Select the "Database Service"**

➡️ **Click on "Create Incident"**

See this



Do this

Your Incident is created.

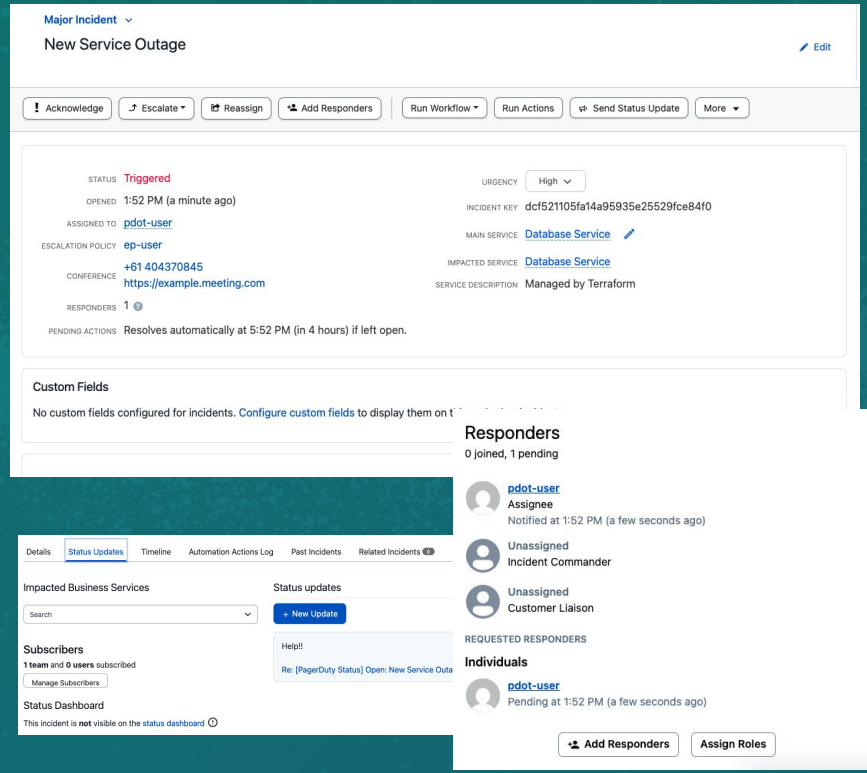
This of course is core Incident Response functionality.

Look at the incident resolve automatically!

Check the below to see what happened:

- Responders
- Timeline
- Status Update

See this



PAUSE and Demo PD Advance and Jeli

PagerDuty ON TOUR