



Creating alerts with PagerDuty

Tracetest is a testing tool based on OpenTelemetry that permits you to test your distributed application. It allows you to use your trace data generated on your OpenTelemetry tools to check and assert if your application has the desired behavior defined by your test definitions. It also supports **Synthetic Monitoring**

PagerDuty is a powerful tool to let you know when things go wrong and require manual intervention to fix it.

By using Tracetest with PagerDuty, you can ensure your application runs as you expect and also get notified when it doesn't work.

Quickstart

In this guide, we will show how to use Tracetest's Synthetic Monitoring to run these tests and use its webhook mechanism to alert you when things fail.

Prerequisites

- A Tracetest account
- A PagerDuty account

Create a PagerDuty Service

NOTE

For a detailed guide, navigate to PagerDuty's support page [Services and Integrations](#).

To create a service in the PagerDuty web app:

1. Go to **Services** -> **Service Directory** and click **New Service**. On the next screen you will

be guided through several steps.

- i. Enter **Name**. We suggest to put a cluster name, for example: "Production EU".
- ii. Click **Next** to continue.

2. Select Escalation Policy that suits you best. Click **Next** to continue.

3. Select Reduce Noise policy that suits you best. Click **Next** to continue.

4. Select the **Events API V2** integration.

Create a Service

✓ Name ——— ✓ Assign ——— ✓ Reduce Noise ——— **4 Integrations**

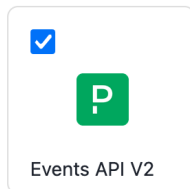
Integrations

Alert feeds can come into PagerDuty from a number of sources. We apply our AI to these alerts and can trigger incidents and notify the right people at the right time.

Select the integration(s) you use to send alerts to this service

Search for an integration(s) ▼

Your selections (1)



5. Click **Create Service**.

6. Now you will be in the service's Integrations tab, where you'll find the **Integration Key**.

7. Copy the **Integration Key** and proceed with the next section.

Events API v2 Overview

The Events API v2 is a highly reliable, highly available asynchronous API that ingests machine events from monitoring tools and other systems like code repositories, observability platforms, automated workflow tools, and configuration management systems. Events sent to this API are ultimately routed to a [PagerDuty service](#) and processed.

Event Types

The Events API v2 can ingest multiple types of events. Each event type is described below.

Event Type	Description	Example Events	Notifications can be sent?
Alert	A problem in a machine monitored system. Follow up events can be sent to acknowledge or resolve an existing alert.	High error rate CPU usage exceeded limit Deployment failed	Yes
		Pull request merged	

Integration Name
Events API v2

Integration Key
R03E2XK1VZNQOE0ST8BG0PCEDLGIIBP

Integration URL (Change Events)
https://events.eu.pagerduty.com/v2/change/enqueue

Integration URL (Alert Events)
https://events.eu.pagerduty.com/v2/enqueue

cURL command example

```
curl --request 'POST' \
  --url 'https://events.eu.pagerduty.com/v2/enqueue' \
  --header 'Content-Type: application/json' \
  --data '{
    "payload": {
      "summary": "Test alert",
      "severity": "critical",
      "source": "Alert source"
    }
  }'
```

Create a Monitor

1. Go to [Tracetest](#)
2. Access your account
3. Click on "Monitors"
4. Click on "Create"
5. Setup your monitor as you like
6. On the "Alert" tab, add the information provided by PagerDuty
 - i. For the method, use "POST"
 - ii. Set the URL to "<https://events.pagerduty.com/v2/enqueue>"
 - iii. Select "JSON" in the "Body" dropdown
 - iv. Paste the suggested body from the PagerDuty page and adjust the messages as needed, it should look like this:

```
{
  "payload": {
    "summary": "Title of my alert",
    "severity": "critical",
    "source": "Alert source"
  },
  "routing_key": "<routing_key>",
  "event_action": "trigger"
}
```

- v. Click on the play button to test your alert, PagerDuty should be notified by this action

7. Click on "Create" to finish the Monitor setup

 [Edit this page](#)