

OpsGenie to Spike integration guide

Every monitoring tool that sends alerts to OpsGenie needs to send them to Spike. Your OpsGenie API keys and URLs do not carry over, because each Spike integration has its own webhook URL.

UPDATED OCTOBER 2026

Prefer code?

The `spike_integration` resource in the [Terraform provider](#) returns each webhook URL, so you can create integrations and pass the URLs to your monitoring config in one apply.

The same steps for every tool

- 1 **Create the integration in Spike.** Add a service, add the integration for your tool, and copy its webhook URL. See [creating integrations and services](#).
- 2 **Add the Spike webhook to the tool.** Keep the OpsGenie target in place for now.
- 3 **Send a test alert.** Use the tool's test notification, and check that an incident appears in Spike with the right title and priority.
- 4 **Check that it closes.** Send a recovery, and confirm the incident resolves in Spike.
- 5 **Remove the OpsGenie target.** Do this when you cut over.

Keep OpsGenie in place while you add Spike

Run both tools side by side with the [parallel run plan](#).

Tool by tool

Datadog

- In Datadog, open Integrations, then Webhooks, and create a webhook with your Spike URL.
- Use the payload from the Spike guide, and keep `$AGGREG_KEY` and `$ALERT_TRANSITION` in it. They let incidents resolve on their own.
- Add the webhook to your monitors, for example `@webhook-spike`.

See the [Datadog guide](#).

Prometheus and Alertmanager

- Add a webhook receiver to `alertmanager.yml` with your Spike URL, and add a route that sends alerts to it.
- To send alerts to both tools for now, give the OpsGenie route `continue: true`.
- Incidents resolve on their own.

See the [Prometheus guide](#).

AWS CloudWatch

- Create an SNS topic, and add an HTTPS subscription that uses your Spike webhook. Spike confirms the subscription for you.
- Add the topic as an action on your CloudWatch alarms.
- Add an OK state action as well, so incidents resolve on their own.

See the [AWS guide](#).

Grafana

- Create a webhook contact point (called a notification channel in older versions), paste your Spike URL, and set the content type to `application/json`.
- Choose the alerts that should use it. Incidents resolve on their own, and repeated alerts are grouped.

See the [Grafana guide](#).

New Relic

- Create a webhook notification channel, and paste your Spike URL as the base URL.
- Paste the custom payload from the Spike guide, and do not edit it. Changes can stop incident messages from parsing.

See the [New Relic guide](#).

Sentry

- Install the Spike app from Sentry's integrations page, which works for one project, or add Spike as a webhook, which works for any number of projects.
- In your alert rules, choose Send a notification via an integration, and select Spike.
- Resolving an incident in Spike does not resolve the issue in Sentry.

See the [Sentry guide](#).

Azure Monitor

- Create an alert rule in Azure Monitor, and add an action that sends to your Spike webhook.
- In the advanced settings, turn on enable upon creation and automatically resolve alerts.

See the [Azure guide](#).

Google Cloud Monitoring

- Add a webhook notification channel with your Spike URL. Basic authentication is optional.
- Select Test Connection, which creates a test incident, and resolve that incident right away.
- Choose the channel in your alerting policies.

See the [Google Cloud guide](#).

One webhook for each tool

Use each integration's own webhook in its own tool. Using it elsewhere can break incident messages and grouping.

Not on this list?

Spike has 80+ integrations, and every one works the same way. Find yours in the [integration guides](#). If a tool is not there, you have two options:

- Send alerts from your own scripts or tools with the [webhook integration](#).
- Use the [email integration](#) for tools that can only send email. Incidents from email do not resolve automatically.

If you need a new integration, write to support@spike.sh, and we will build it.

Need help with your migration?

Write to support@spike.sh, or book a call with our team at spike.sh/demo, and we will help you map your setup.