+++ title = “Alert notifications” description = “Alerting notifications guide” keywords = [“Grafana”, “alerting”, “guide”, “notifications”] weight = 100 aliases = [“/docs/grafana/latest/alerting/notifications/”] +++

Alert notifications

When an alert changes state, it sends out notifications. Each alert rule can have multiple notifications. In order to add a notification to an alert rule you first need to add and configure a `notification` channel (can be email, PagerDuty, or other integration).

This is done from the Notification channels page.

> **Note:** Alerting is only available in Grafana v4.0 and above.

Add a notification channel

1. In the Grafana side bar, hover your cursor over the **Alerting** (bell) icon and then click **Notification channels**.
2. Click **Add channel**.
3. Fill out the fields or select options described below.

New notification channel fields

Default (send on all alerts)

- **Name -** Enter a name for this channel. It will be displayed when users add notifications to alert rules.
- **Type -** Select the channel type. Refer to the List of supported notifiers for details.
- **Default (send on all alerts) -** When selected, this option sends a notification on this channel for all alert rules.
- **Include Image -** See Enable images in notifications for details.
- **Disable Resolve Message -** When selected, this option disables the resolve message [OK] that is sent when the alerting state returns to false.
- **Send reminders -** When this option is checked additional notifications (reminders) will be sent for triggered alerts. You can specify how often reminders should be sent using number of seconds (s), minutes (m) or hours (h), for example `30s`, `3m`, `5m` or `1h`.

Important: Alert reminders are sent after rules are evaluated. Therefore a reminder can never be sent more frequently than a configured alert rule evaluation interval.

These examples show how often and when reminders are sent for a triggered alert.

Alert rule evaluation interval	Send reminders every	Reminder sent every (after last alert notification)
`30s`	`15s`	~30 seconds
`1m`	`5m`	~5 minutes
`5m`	`15m`	~15 minutes
`6m`	`20m`	~24 minutes
`1h`	`15m`	~1 hour
`1h`	`2h`	~2 hours

List of supported notifiers

Name	Type	Supports images	Supports alert rule tags
DingDing	`dingding`	yes, external only	no
Discord	`discord`	yes	no
Email	`email`	yes	no
Google Hangouts Chat	`googlechat`	yes, external only	no
Hipchat	`hipchat`	yes, external only	no
Kafka	`kafka`	yes, external only	no
Line	`line`	yes, external only	no
Microsoft Teams	`teams`	yes, external only	no
Opsgenie	`opsgenie`	yes, external only	yes
Pagerduty	`pagerduty`	yes, external only	yes
Prometheus Alertmanager	`prometheus-alertmanager`	yes, external only	yes

Name	Type	Supports images	Supports alert rule tags
Pushover	`pushover`	yes	no
Sensu	`sensu`	yes, external only	no
Sensu Go	`sensugo`	yes, external only	no
Slack	`slack`	yes	no
Telegram	`telegram`	yes	no
Threema	`threema`	yes, external only	no
VictorOps	`victorops`	yes, external only	yes
Webhook	`webhook`	yes, external only	yes

Email

To enable email notifications you have to set up [SMTP settings]({{< relref "../../administration/configuration/#smtp" >}}) in the Grafana config. Email notifications will upload an image of the alert graph to an external image destination if available or fallback to attaching the image to the email. Be aware that if you use the `local` image storage email servers and clients might not be able to access the image.

Note: Template variables are not supported in email alerts.

Setting	Description
Single email	Send a single email to all recipients. Disabled per default.
Addresses	Email addresses to recipients. You can enter multiple email addresses using a ";" separator.

Slack

{{< figure class="float-right" max-width="40%" src="/static/img/docs/v4/slack_notification.png" caption="Alerting Slack Notification" >}}

To set up Slack, you need to configure an incoming Slack webhook URL. You can follow Sending messages using Incoming Webhooks on how to do that. If you

want to include screenshots of the firing alerts in the Slack messages you have to configure either the external image destination in Grafana or a bot integration via Slack Apps. Follow Slack's guide to set up a bot integration and use the token provided, which starts with "xoxb".

Setting	Description
Url	Slack incoming webhook URL, or eventually the chat.postMessage Slack API endpoint.
Username	Set the username for the bot's message.
Recipient	Allows you to override the Slack recipient. You must either provide a channel Slack ID, a user Slack ID, a username reference (@<user>, all lowercase, no whitespace), or a channel reference (#<channel>, all lowercase, no whitespace). If you use the `chat.postMessage` Slack API endpoint, this is required.
Icon emoji	Provide an emoji to use as the icon for the bot's message. Ex :smile:
Icon URL	Provide a URL to an image to use as the icon for the bot's message.
Mention Users	Optionally mention one or more users in the Slack notification sent by Grafana. You have to refer to users, comma-separated, via their corresponding Slack IDs (which you can find by clicking the overflow button on each user's Slack profile).
Mention Groups	Optionally mention one or more groups in the Slack notification sent by Grafana. You have to refer to groups, comma-separated, via their corresponding Slack IDs (which you can get from each group's Slack profile URL).
Mention Channel	Optionally mention either all channel members or just active ones.
Token	If provided, Grafana will upload the generated image via Slack's file.upload API method, not the external image destination. If you use the `chat.postMessage` Slack API endpoint, this is required.

If you are using the token for a slack bot, then you have to invite the bot to the channel you want to send notifications and add the channel to the recipient field.

Opsgenie

To setup Opsgenie you will need an API Key and the Alert API Url. These can be obtained by configuring a new Grafana Integration.

Setting	Description
Alert API URL	The API URL for your Opsgenie instance. This will normally be either `https://api.opsgenie.com` or, for EU customers, `https://api.eu.opsgenie.com`.
API Key	The API Key as provided by Opsgenie for your configured Grafana integration.
Override priority	Configures the alert priority using the `og_priority` tag. The `og_priority` tag must have one of the following values: `P1`, `P2`, `P3`, `P4`, or `P5`. Default is `False`.
Send notification tags as	Specify how you would like [Notification Tags]({{< relref "create-alerts.md/#notifications" >}}) delivered to Opsgenie. They can be delivered as `Tags`, `Extra Properties` or both. Default is Tags. See note below for more information.

> **Note:** When notification tags are sent as `Tags` they are concatenated into a string with a `key:value` format. If you prefer to receive the notifications tags as key/values under Extra Properties in Opsgenie then change the `Send notification tags as` to either `Extra Properties` or `Tags & Extra Properties`.

PagerDuty

To set up PagerDuty, all you have to do is to provide an integration key.

Setting	Description
Integration Key	Integration key for PagerDuty.
Severity	Level for dynamic notifications, default is `critical` (1)
Auto resolve incidents	Resolve incidents in PagerDuty once the alert goes back to ok
Message in details	Removes the Alert message from the PD summary field and puts it into custom details instead (2)

> **Note:** The tags `Severity`, `Class`, `Group`, `dedup_key`, and `Component` have special meaning in the Pagerduty Common Event Format - PD-CEF. If an alert panel defines these tag keys, then they are transposed to the root of the event sent to Pagerduty. This means they will be available within the Pagerduty UI and Filtering tools. A Severity tag set on an alert overrides the global Severity set on the notification channel if it's a valid level.
>
> Using Message In Details will change the structure of the `custom_details` field in the PagerDuty Event. This might break custom event rules in your PagerDuty rules if you rely on the fields in `payload.custom_details`. Move any existing rules using

`custom_details.myMetric` to `custom_details.queries.myMetric`. This behavior will become the default in a future version of Grafana.

Note: The `dedup_key` tag overrides the Grafana-generated `dedup_key` with a custom key.

Note: The `state` tag overrides the current alert state inside the `custom_details` payload.

Note: Grafana uses the `Events API V2` integration. This can be configured for each service.

VictorOps

To configure VictorOps, provide the URL from the Grafana Integration and substitute `$routing_key` with a valid key.

Note: The tag `Severity` has special meaning in the VictorOps Incident Fields. If an alert panel defines this key, then it replaces the `message_type` in the root of the event sent to VictorOps.

Pushover

To set up Pushover, you must provide a user key and an API token. Refer to What is Pushover and how do I use it for instructions on how to generate them.

Setting	Description
API Token	Application token
User key(s)	A comma-separated list of user keys
Device(s)	A comma-separated list of devices
Priority	The priority alerting nottifications are sent
OK priority	The priority OK notifications are sent; if not set, then OK notifications are sent with the priority set for alerting notifications
Retry	How often (in seconds) the Pushover servers send the same notification to the user. (minimum 30 seconds)
Expire	How many seconds your notification will continue to be retried for (maximum 86400 seconds)
Alerting sound	The sound for alerting notifications
OK sound	The sound for OK notifications

Webhook

The webhook notification is a simple way to send information about a state change over HTTP to a custom endpoint. Using this notification you could

integrate Grafana into a system of your choosing.

Example json body:


```
{
  "dashboardId": 1,
  "evalMatches": [
    {
      "value": 1,
      "metric": "Count",
      "tags": {}
    }
  ],
  "imageUrl": "https://grafana.com/assets/img/blog/mixed_styles.png",
  "message": "Notification Message",
  "orgId": 1,
  "panelId": 2,
  "ruleId": 1,
  "ruleName": "Panel Title alert",
  "ruleUrl": "http://localhost:3000/d/hZ7BuVbWz/test-dashboard?fullscreen\u0026edit\u0026tab
  "state": "alerting",
  "tags": {
    "tag name": "tag value"
  },
  "title": "[Alerting] Panel Title alert"
}
```


- **state** - The possible values for alert state are: `ok`, `paused`, `alerting`, `pending`, `no_data`.

DingDing/DingTalk

DingTalk supports the following "message type": `text`, `link` and `markdown`. Only the `link` message type is supported. Refer to the configuration instructions in Chinese language.

In DingTalk PC Client:

1. Click "more" icon on upper right of the panel.
2. Click "Robot Manage" item in the pop menu, there will be a new panel call "Robot Manage".
3. In the "Robot Manage" panel, select "customized: customized robot with Webhook".
4. In the next new panel named "robot detail", click "Add" button.
5. In "Add Robot" panel, input a nickname for the robot and select a "message group" which the robot will join in. click "next".

6. There will be a Webhook URL in the panel, looks like this: https://oapi.dingtalk.com/robot/send?access_token=xxxxxxxxx. Copy this URL to the Grafana DingTalk setting page and then click “finish”.

Discord

To set up Discord, you must create a Discord channel webhook. For instructions on how to create the channel, refer to Intro to Webhooks.

Setting	Description
Webhook URL	Discord webhook URL.
Message Content	Mention a group using @ or a user using <@ID> when notifying in a channel.
Avatar URL	Optionally, provide a URL to an image to use as the avatar for the bot’s message.
Use Discord’s Webhook Username	Use the username configured in Discord’s webhook settings. Otherwise, the username will be ‘Grafana.’

Alternately, use the Slack notifier by appending `/slack` to a Discord webhook URL.

Kafka

Notifications can be sent to a Kafka topic from Grafana using the Kafka REST Proxy. There are a couple of configuration options which need to be set up in Grafana UI under Kafka Settings:

1. Kafka REST Proxy endpoint.
2. Kafka Topic.

Once these two properties are set, you can send the alerts to Kafka for further processing or throttling.

Google Hangouts Chat

Notifications can be sent by setting up an incoming webhook in Google Hangouts chat. For more information about configuring a webhook, refer to webhooks.

Prometheus Alertmanager

Alertmanager handles alerts sent by client applications such as Prometheus server or Grafana. It takes care of deduplicating, grouping, and routing them to the correct receiver. Grafana notifications can be sent to Alertmanager via a simple incoming webhook. Refer to the official Prometheus Alertmanager documentation for configuration information.

> **Caution:** In case of a high-availability setup, do not load balance traffic between Grafana and Alertmanagers to keep coherence between all your Alertmanager instances. Instead, point Grafana to a list of all Alertmanagers, by listing their URLs comma-separated in the notification channel configuration.

Sensu Go

Grafana alert notifications can be sent to Sensu Go as events via the API. This operation requires an API key. For information on creating this key, refer to Sensu Go documentation.

Enable images in notifications

Grafana can render the panel associated with the alert rule as a PNG image and include that in the notification. Read more about the requirements and how to configure [image rendering]({{< relref "../../image-rendering/" >}}).

You must configure an [external image storage provider]({{< relref "../../administration/configuration/#external-image-storage" >}}) in order to receive images in alert notifications. If your notification channel requires that the image be publicly accessible (e.g. Slack, PagerDuty), configure a provider which uploads the image to a remote image store like Amazon S3, Webdav, Google Cloud Storage, or Azure Blob Storage. Otherwise, the local provider can be used to serve the image directly from Grafana.

Notification services which need public image access are marked as 'external only'.

Configure the link back to Grafana from alert notifications

All alert notifications contain a link back to the triggered alert in the Grafana instance. This URL is based on the [domain]({{< relref "../../administration/configuration/#domain" >}}) setting in Grafana.

Notification templating

> **Note:** Alert notification templating is only available in Grafana v7.4 and above.

The alert notification template feature allows you to take the [label]({{< relref "../../basics/timeseries-dimensions.md#labels" >}}) value from an alert query and [inject that into alert notifications]({{< relref "./add-notification-template.md" >}}).