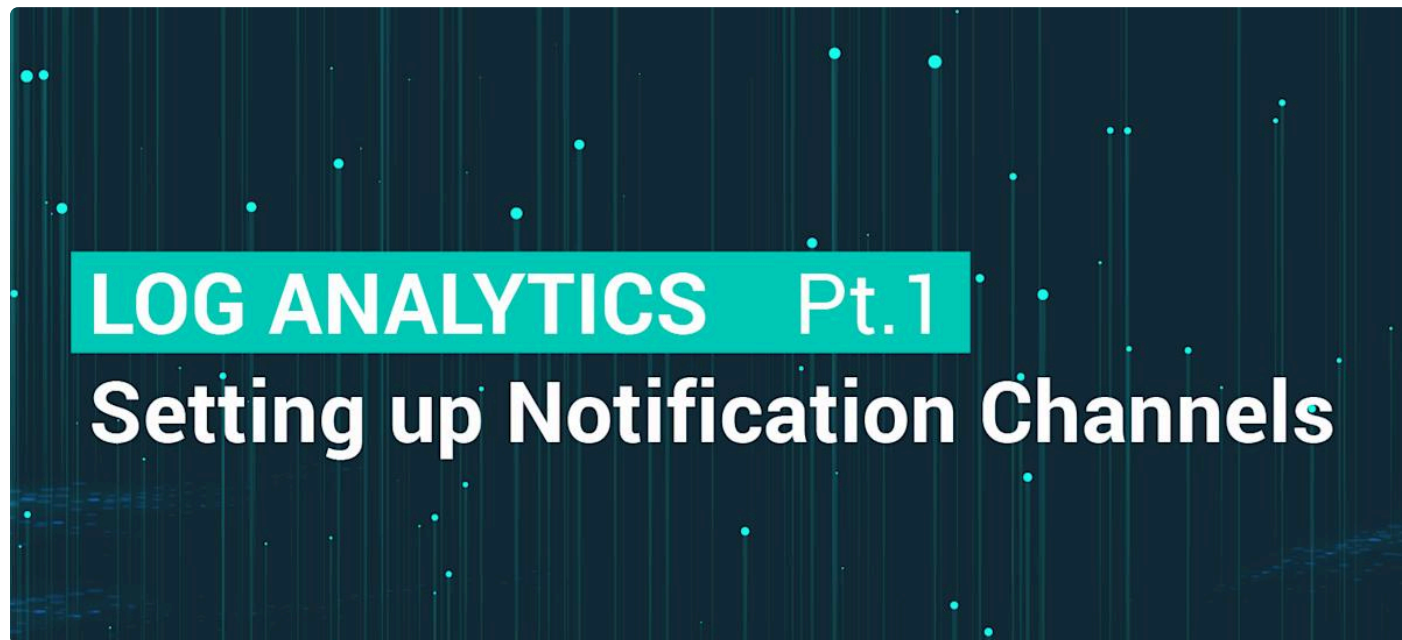


Log Monitoring with Signals Pt.1: Setting up Notification Channels

Jochen Kressin · 2024-07-18



Search Guard Signals is a powerful monitoring and alerting solution designed for Elasticsearch environments. It enables users to proactively monitor their data by setting up notifications for specific events or anomalies detected. With a range of supported notification channels like Email, Slack, and PagerDuty, Search Guard Alerting ensures timely awareness and response to critical system events.

In this article series, we will show you how to use Signals to monitor ingested log data for anomalies and send notifications on various channels if such an anomaly is detected.

Our first objective is to set up email notifications when errors spike in the logs. Additionally, we'll configure a Slack channel to receive these notifications in parallel. Later, we'll use *Severity Levels* to determine whether to send notifications to Slack or Email based on the severity of the detected errors.

This article will walk you through preparing sample log data and setting up notification channels. In the next part, we'll create a simple watch that monitors the log files index for errors.

Importing the Sample Data

First, we'll import the Kibana sample log data. On the home screen, click on "Add sample data" and select the "Sample web logs" data:

Welcome home



Observability

Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.



Security

Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.



Analytics

Explore, visualize, and analyze your data using a powerful suite of analytical tools and applications.

Get started by adding integrations

To start working with your data, use one of our many ingest options. Collect data from an app or service, or upload a file. If you're not ready to use your own data, play with a sample data set.

[+ Add integrations](#)[Try sample data](#)[Upload a file](#)

Try managed Elastic

Deploy, scale, and upgrade your stack faster with Elastic Cloud. We'll help you quickly move your data.

[Move to Elastic Cloud](#)

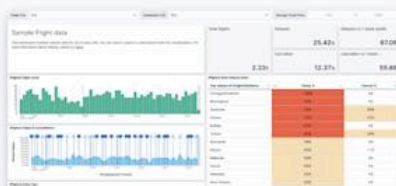
Explore our live demo environment

Browse real-world data in a demo environment where you can explore search, observability, and security use cases like yours.

[Start exploring](#)[Other sample data sets](#)

Sample eCommerce orders

Sample data, visualizations, and dashboards for tracking eCommerce orders.

[Add data](#)

Sample flight data

Sample data, visualizations, and dashboards for monitoring flight routes.

[Add data](#)

Sample web logs

Sample data, visualizations, and dashboards for monitoring web logs.

[Add data](#)

This will import sample weblog data and set up a dashboard with some visualizations:

Dashboards

 Search...

Recently updated ▾

Tags ▾

☐ Name, description, tags

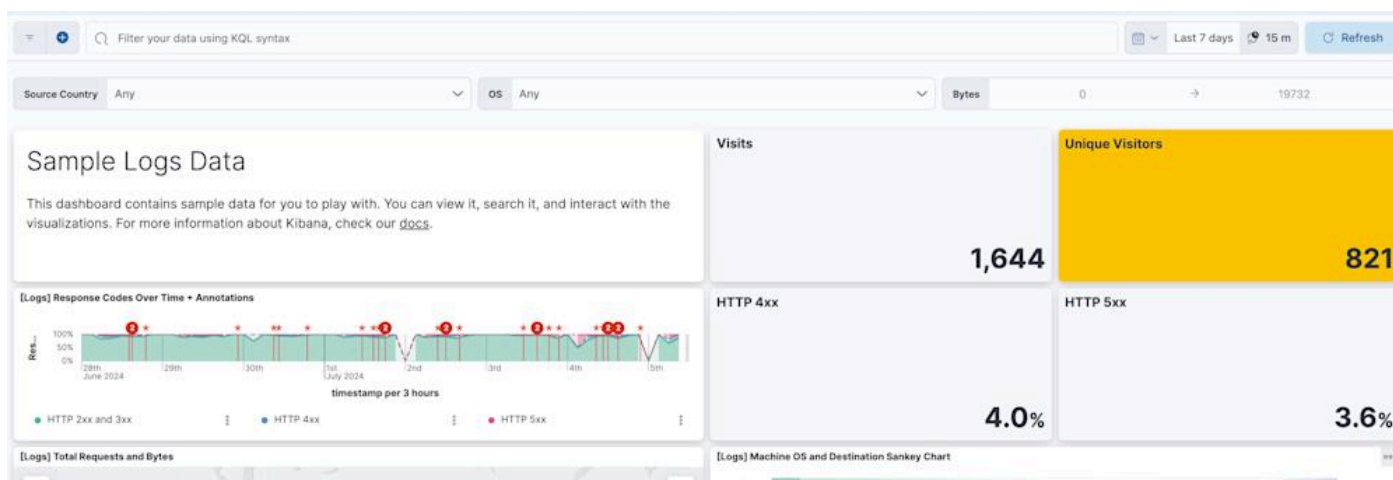
Last updated ▾

Actions

☐ [Logs] Web Traffic
Analyze mock web traffic log data for Elastic's website

6 seconds ago

Rows per page: 20 ▾

 1


A sample document in our logs index looks roughly like this (some fields have been omitted for brevity):

```
{
  "agent": "Mozilla/5.0 (X11; Linux x86_64; rv:6.0a1) Gecko/20110421 Firefox/6.0a1",
  "bytes": 5297,
  "clientip": "212.46.165.124",
  "extension": "deb",
  "host": "artifacts.elastic.co",
  "index": "kibana_sample_data_logs",
  "ip": "212.46.165.124",
  "machine": {
    "ram": 5368709120,
    "os": "osx"
  },
  ...,
  "memory": null,
  "referer": "http://twitter.com/success/charles-fullerton",
  "request": "/elasticsearch/elasticsearch-6.3.2.deb",
  "response": 200,
  "@timestamp": "2024-07-07T18:40:10.094Z",
  "url": "https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.3.2.deb",
  "utc_time": "2024-07-07T18:40:10.094Z",
}
```

The document contains a *response* field with the request's HTTP response code. We'll use this field to check for errors and the *timestamp* field to check for errors within a specific time period.

Adding an Email Notification Channel

Signals supports several notification channels:

- Email
- Slack
- PagerDuty
- JIRA
- Webhooks
- Index (to write data back to an Elasticsearch index)

In this example, we'll set up an email notification for unusual error rates in our logs. While everything can be configured via the [Signals API](#), we'll use the Signals Kibana UI for simplicity.

First, select "Alerting" from the Dashboards Navigation:

D

GI

Dashboard

[L

Home

Recently viewed

[Logs] Web Traffic

Timelines

Cases

Explore

Intelligence

Manage

Management

Dev Tools

Integrations

Fleet

Osquery

Stack Monitoring

Stack Management



Search Guard



Configuration

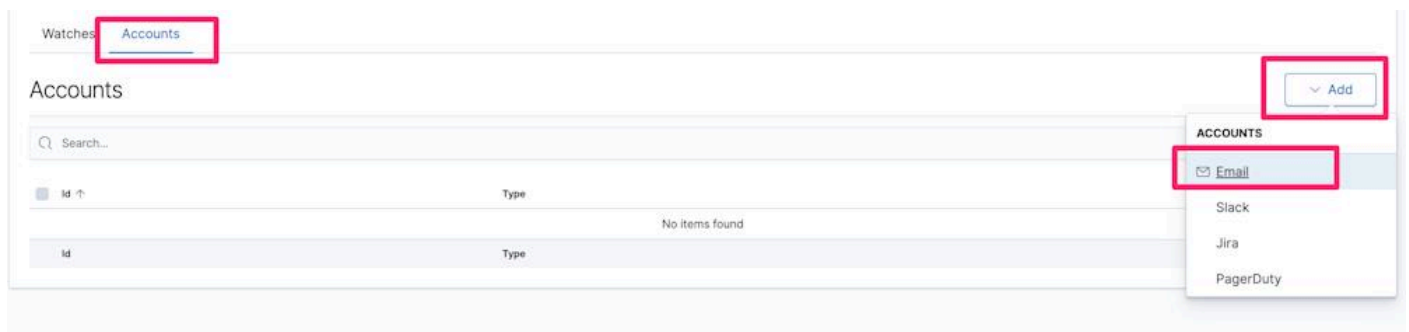
Auth Tokens

Signals

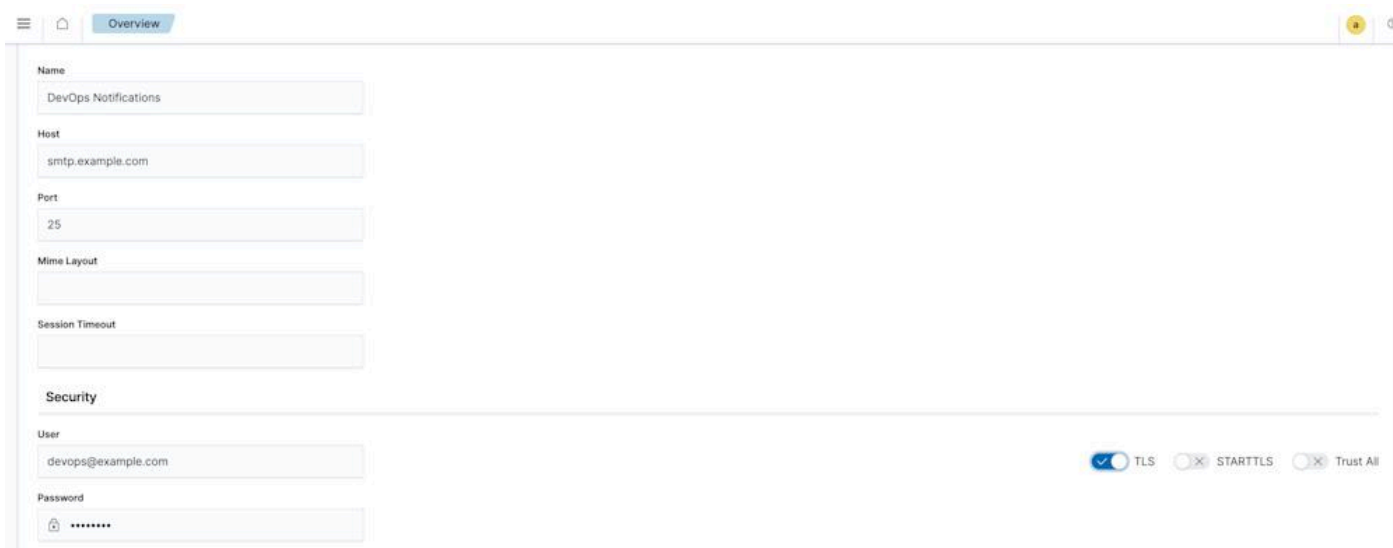


Add integrations

To set up an email notification channel, click on "Accounts" → "Add" → "Email".



On the next screen, configure your Email server connection details. The minimal fields required are the hostname, port, and the credentials of the Email account:



The screenshot shows the 'Overview' tab of a configuration interface. It contains several input fields for email server settings:

- Name:** DevOps Notifications
- Host:** smtp.example.com
- Port:** 25
- Mime Layout:** (empty)
- Session Timeout:** (empty)
- Security:**
 - User:** devops@example.com
 - Password:** (masked with dots)

On the right side of the Security section, there are three toggle switches:

- TLS:** Enabled (blue checkmark)
- STARTTLS:** Disabled (grey X)
- Trust All:** Disabled (grey X)

We want to verify our notifications before sending actual emails. Enable the *Simulate* mode and set *Debug* to true. *Simulate* mode provides information about the Email that would be sent without actually sending it, which is great for testing the setup. We can disable these options after setting up and testing the watch.

Defaults

From

notifications@example.com

To

devops@example.com ×



Cc



Bcc



Debug



Debug



Simulate

Adding a Slack Notification Channel

Next, we'll set up a Slack notification channel. First, add a new channel to Slack called *alerting*. To send notifications to this channel, add a Slack App. This can be done with just a few clicks.

From the Slack Admin UI, add a new App, give it a name, and define the Slack Workspace to add the App to:

Name app & choose workspace



App Name

Don't worry - you'll be able to change this later.

Pick a workspace to develop your app in:

Keep in mind that you can't change this app's workspace later. If you leave the workspace, you won't be able to manage any apps you've built for it. The workspace will control the app even if you leave the workspace.

[Sign into a different workspace](#)

By creating a **Web API Application**, you agree to the [Slack API Terms of Service](#).

CancelCreate App

Next, choose "Add features and functionality" and select "Incoming Webhook":

Add features and functionality

Choose and configure the tools you'll need to create your app (or review all [our documentation](#)).

Building an internal app locally or behind a firewall?

To receive your app's payloads over a WebSockets connection, enable [Socket Mode](#) for your app.

Incoming Webhooks

Post messages from external sources into Slack.

Interactive Components

Add components like buttons and select menus to your app's interface, and create an interactive experience for users.

Slash Commands

Allow users to perform app actions by typing commands in Slack.

Event Subscriptions

Make it easy for your app to respond to activity in Slack.

Bots

Allow users to interact with your app through channels and conversations. ✨

Permissions

Configure permissions to allow your app to interact with the Slack API.

Activate Incoming Webhooks

On

Incoming webhooks are a simple way to post messages from external sources into Slack. They make use of normal HTTP requests with a JSON payload, which includes the message and a few other optional details. You can include [message attachments](#) to display richly-formatted messages.

Adding incoming webhooks requires a bot user. If your app doesn't have a [bot user](#), we'll add one for you.

Each time your app is installed, a new Webhook URL will be generated.

If you deactivate incoming webhooks, new Webhook URLs will not be generated when your app is installed to your team. If you'd like to remove access to existing Webhook URLs, you will need to [Revoke All OAuth Tokens](#).

Webhook URLs for Your Workspace

To dispatch messages with your webhook URL, send your [message](#) in JSON as the body of an `application/json` POST request.

Add this webhook to your workspace below to activate this curl example.

Sample curl request to post to a channel:

```
curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello, World!"}' YOUR_WEBHOOK_URL_HERE
```

Webhook URL	Channel	Added By
No webhooks have been added yet.		

Add New Webhook to Workspace

On the next screen, select the channel to post to, in this case, *alerting*.

This app was created by a member of your workspace, eliatra.



Alerting Plus Notifications is requesting permission to access the eliatra Slack workspace

Where should Alerting Plus Notifications post?

Alerting Plus Notifications requires a channel to post to as an app

alerting-plus-notifications

Cancel

Allow

After setting up the webhook, copy the webhook URL provided:

Webhook URLs for Your Workspace

To dispatch messages with your webhook URL, send your [message](#) in JSON as the body of an `application/json` POST request.

Add this webhook to your workspace below to activate this curl example.

Sample curl request to post to a channel:

```
curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello, World!"}'
https://hooks.slack.com/services/XXXXXXXXXXXX/XXXXXXXXXXXX/XXXXXXXXXXXX
```

Copy

Webhook URL	Channel	Added By
https://hooks.slack.com/serv Copy	#alerting	 

Add New Webhook to Workspace

The URL will be in the format:

`https://hooks.slack.com/services/XXXX/YYYY/ZZZZ`

The URL will be in the format: `https://hooks.slack.com/services/XXXX/YYYY/ZZZZ`

Copy this URL, go to the Signals Accounts page, and add a new Slack channel. The required entries are the channel name and the Webhook URL:

Overview

Home / Accounts / Create Account

Create Account

Account slack

Name

Alerting

Url

`https://hooks.slack.com/services/T01DBFKLDCU/B05HAC`

Cancel Create

That's it for this post. In the next article, we'll set up a watch to monitor our log data and send notifications to the channels we've created.

Articles in this series

- Log monitoring with Signals, Pt.1: Setting up Notification Channels (this article)
- Log monitoring with Signals, Pt.2: Setting up a Watch in Blocks Mode
- Log monitoring with Signals, Pt.3: Sending Notifications
- Log monitoring with Signals, Pt.4: Implementing Escalation Levels

--- Image source: Sutterstock-Vektorgrafik ID: 1007122360 [Garry Killian](#)

This article was published on the Search Guard blog: <https://search-guard.com/blog/log-monitoring-signals-pt-1-notification-channels/>

Search Guard is Security and Alerting for Elasticsearch and Kibana — search-guard.com · [Start a free trial](#)