SCALYR®

WHITE PAPER

Starting With Logs:

An Application Note

August 2019

Your most potent monitoring tool is already part of your application.

Building software is just the beginning. You're now on the hook for deploying it into production and keeping it running, too. Every release means new fixes, new features, and new risks. When you roll out a new version of your application, you need to take advantage of every resource at your disposal. So, how are you going to get the most from your logs?

Today's architectures are dynamic and distributed. Transactions span many containers, microservices, and serverless functions. How can you manage the chaos? Use your logs. They tie these disparate services together. With a comprehensive log strategy, you can create a holistic end-to-end view. But you can't do this if you treat logs as an afterthought.

Observability starts with your code.

APM and metrics can help you head some problems off before they occur. But nothing gives you a better picture of what's happening inside your code than logs. Of course, you'll only get this clear picture with carefully-crafted messages and the right tools. If you're not managing your logs well, you're leaving one of your most powerful resources on the table. You need a strategy.

STRATEGIC LOGGING

When most people think of logs, they think of errors. But that's just the tip of the iceberg. A comprehensive logging strategy encompasses errors, events, and metrics. First, you record enough information—and the right information. Then, you turn your logs into your primary troubleshooting and monitoring tool.

ERROR LOGGING

Experienced developers understand the necessity of logging errors. But are you logging enough information to isolate problems quickly?

- **Where am I?** Where did the error occur? An exception provides a stack trace, but if you're recording an error, make sure you log where it happened.
- **How did I get here?** The location of the error is only the start. How did your code get there? If it's in response to a request, log the request parameters. For events, include the event. If it's the result of a state change, include the states.
- **What am I doing?** Log the data involved in the error, not just the operation that failed.
- **Who was I talking to?** Today's complex systems rely on communication between different services and systems. Errors caused or related to failures to send or receive messages between entities deserve special attention.

EVENT LOGGING

While it's good to talk about errors and the events that might lead to them, a complete logging strategy captures events before they become errors. Maintaining a record of events is a tactic that transforms your logs into a critical resource. Here's how to leverage yours:

- Proactively logging events gives you context for errors when they finally do occur. As discussed later in this white paper, Scalyr places events in context so that you can compare or overlay them with error messages.
- By recording the right information with each event, you can use logs to track data about your system.
- If you don't want to log every event, log periodic or random samples.
- The events that are distributed across multiple containers or servers together with a shared ID.

QUANTITATIVE AND QUALITATIVE LOG METRICS

Strategic logging means logging complete information about errors and events so you can put that information to work. If you have the right tools and set them up correctly, you can track performance and user experience metrics with your logs. APM tools suffer from integration and performance problems. Logs, which you have complete control over, don't.

Here are some ways you can log strategically:

- Log metrics about data and events. How long did it take to process a request or receive a response? How many bytes was the response? How many records did it contain?
- Log events where items are capped, limited, or discarded. Depending on your application, you might not consider these events errors. But they may be indicative of future problems or opportunities to improve your code.

SCALYR: CRITICAL TOOLS FOR YOUR LOGGING STRATEGY

Logging errors and events is only half of your logging strategy. You also need tools that can transform data into intelligence. When you can do this, you can actually start troubleshooting with your logs, which are the single source of truth, rather than using them as a last resort.

In order to have this intelligence at your engineers' fingertips, your logging solution must ingest your logs in real time and make the content available in seconds. You can then use this data stream to monitor application state and detect performance problems before they become issues.

Scalyr is built to help your devops teams troubleshoot fast so that you can deliver a better customer experience and focus their skills on continued innovation for your business. By leveraging the value of your logs at the beginning of the troubleshooting process, rather than the end, you save critical time and resources.

LIVE TAIL

The first step to taking a strategic approach to managing your logs is to centralize them. It is nearly impossible to maintain logs when they're scattered across an array of systems or containers. Scalyr is your centralized log system.

If you've developed or supported server-side software before, you've followed a log with **tail** or **less** in a terminal window. You may have even pinned up two terminal sessions and tried to follow two logs side by side. It was difficult to do that when we deployed software on rack-mounted servers. Now, with the hoops that containers and cloud security put in front of you, it cannot be done.

Scalyr's Live Tail displays a dynamically updated list of log events in response to a query. You watch your logs at a central location, in real time. Need to watch a new deployment? Give Scalyr the right search criteria and view the logs for any further issues.

NO-INDEX PARSING LOG DATA

Scalyr is not limited by the constraints of keyword indexes—we are built for speed. Scalyr parses your logs into structured fields that are stored in a NoSQL database. Each field is available for searches, queries, graphs, and alerts. So your efforts to log useful information about errors and events lead to a real-time stream of actionable data.

SEARCH AND QUERIES

No doubt you have used log aggregation tools with keyword search facilities. Searching for log entries with keywords and text expressions is useful—even essential. But you need more. Your logs contain critical metrics about your system. Scalyr's structured approach to managing logs takes full advantage of the data they contain . Instead of searching logs, you query them.

Scalyr's query language lets you group, sort, manipulate, and display log information. Then, you can take the result of any query and use it in graphs or alerts. Let's take a quick look at an example.

Imagine a set of web server access logs. You want to view access errors, grouped into the top four pages. This query will count errors based on HTTP status code and display it in tabular format.

```
$logfile contains 'access'
```

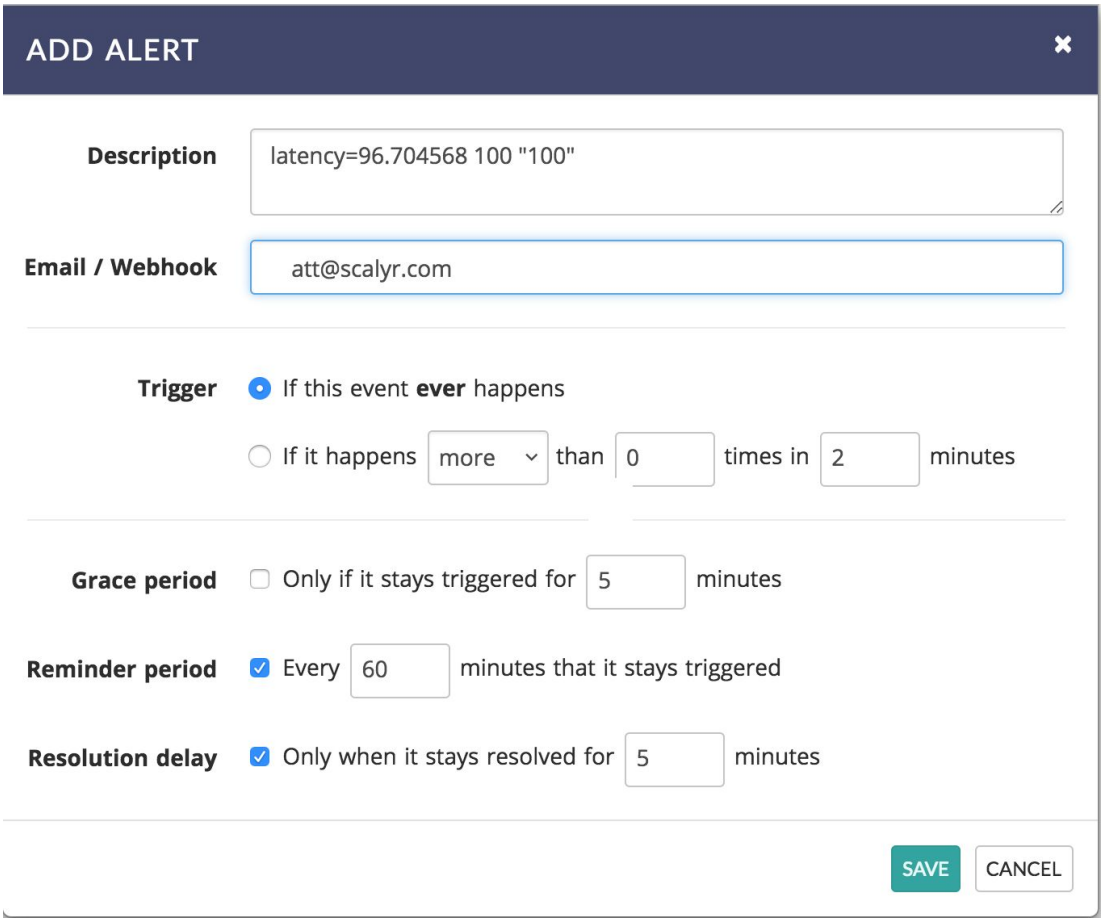
```
| group total = count(),
    clientErrors = count(status >= 400 && status <= 499),
    serverErrors = count(status >= 500 && status <= 599)
    by uriPath
| sort -total
| limit 5
```

uriPath	total	clientErrors	serverErrors
/home	8319	2	6
/news	6214	108	39
/blog	1125	31	0
/login	538	14	2

You can apply Scalyr's query language to searches, alerts, and graphs. In other words, you can analyze or track any piece of structured data from your logs.

ALERTS

There are times when you need Live Tail's ability to watch your logs as events occur. But most of the time, you have other things to do. You need a monitoring system that can alert you when an event occurs or when the application breaks a threshold.

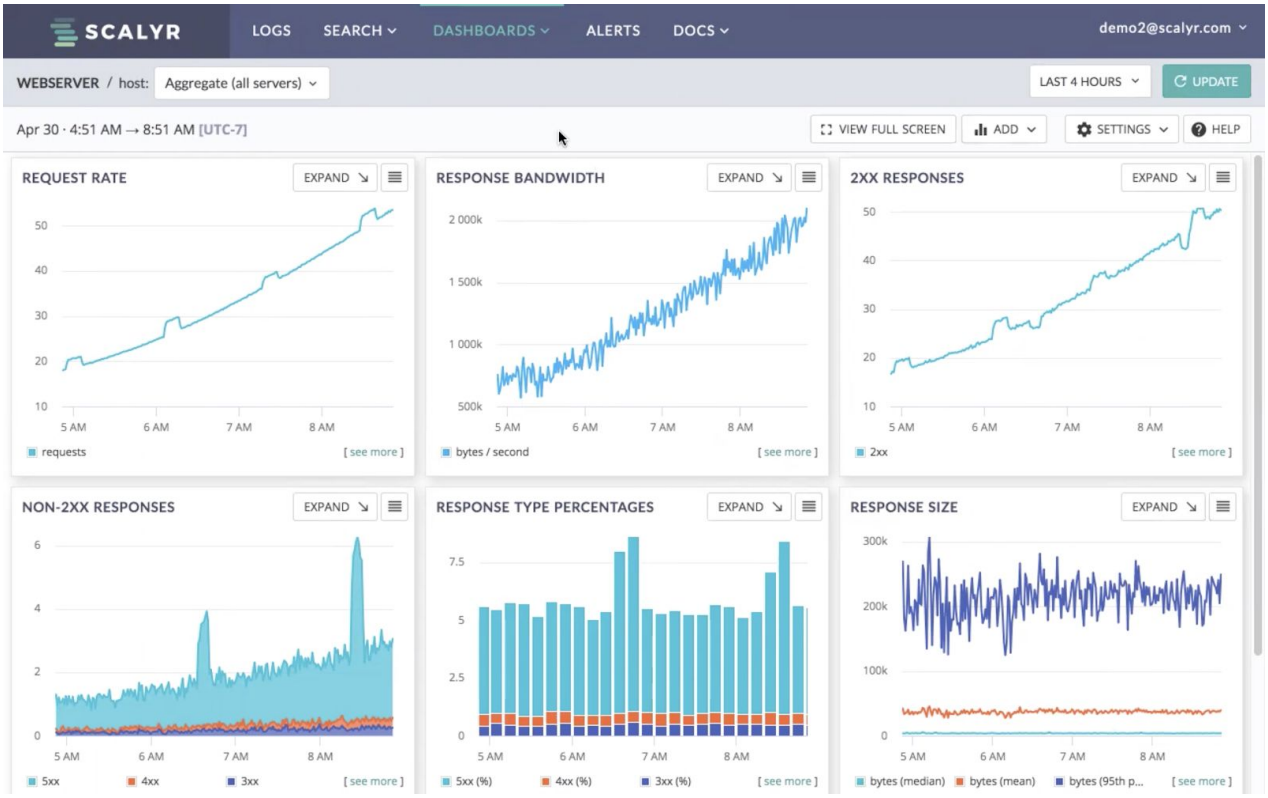


You can create alerts that watch any piece of data in your logs, or you can create an expression that combines several different values. Alerts work with triggers that are expressed in Scalyr's query language.

GRAPHS AND DASHBOARDS

Scalyr's graphing abilities complete the set of tools you need to make logs the linchpin of your monitoring platform. Don't wait for a problem to crop up and then search your logs. Instead, apply your logging strategy to watch your system proactively.

You can graph any structured field with Scalyr. Depending on the data, you can choose from line or bar charts. You can also break down graphs based on the system or roll up values into a consolidated view.



Scalyr's dashboards combine graphs into a composite view of your system. They provide some default layouts to start with, but you can customize them to suit your needs.

YOUR MOST IMPORTANT SOURCE OF DATA MONITORING

Today's distributed architectures are complex, and they're not going to change anytime soon. Instead of using a new monitoring stack to tie things together, the answer is right in front of you and entirely under your control.

If your logging isn't the foundation of your observability strategy, think again. Don't stream your log files into a server for keyword search and wait for a reason to look at them. Develop a plan for managing one of your most critical assets.