CHAOSSEARCH

DEVOPS FORENSIC FILES:

Using Log Analytics to Increase Efficiency

Apply log analytics to discover trends and solve problems within your applications and infrastructure

CONTENTS

INTRODUCTION

DevOps adoption has nearly doubled in the last five years, with 74% of organizations using a DevOps methodology in some form to accelerate delivery of software and applications.

Organizations that combine once-distinct development and operations roles into a continuous integration/continuous delivery (CI/CD) model have discovered that they can deliver quality software much faster and more consistently than previous, siloed approaches.

Along with the swift rise of DevOps, there has been an equally pervasive adoption of cloud, containers, and microservices architecture. Cloud application deployment architectures are much more diverse than traditional architectures, they're increasingly distributed, and they consist of dynamically changing, cloud-native components and services. In other words, the ways in which an application interacts with its environment have changed dramatically. That's made it more complicated to optimize the user experience or troubleshoot issues with applications, systems and infrastructure. To understand these complex environments, DevOps and Data Engineering teams must combine log data from a variety of sources. (This process is easier said than done, as we'll explore later.)

In response to these challenges, DevOps teams rely on a host of management tools for use cases including:

- Infrastructure monitoring
- Network performance monitoring (NPM)
- Application performance monitoring (APM)
- Full-stack observability
- … and more

DevOps teams rely on APM and observability tools, in particular, to get real-time alerts. And, as their names imply, these tools provide visibility into the full stack (from infrastructure to code) so teams can isolate and respond to issues. However, the strength of APM and observability tools is also their weakness. They are excellent for real-time visibility into short-term operational data. They have good workflows and UIs. But they were not designed for log analytics at scale. Today's modern architectures and log volumes make analytics at scale a complex and expensive problem to solve.

That's why DevOps teams should look to a dedicated log analytics solution to meet these unique requirements. Log analytics are one of the first practices developers and engineers turn to when they need to investigate an outage or performance issue. However, today's volume and variety of log data is overwhelming infrastructure and APM/observability tools. They simply weren't designed for long-term retention of high-volume log data. In fact, the volume is too much for platforms originally designed for logs, such as Elasticsearch. Without a good solution for log analytics at scale, DevOps teams either sacrifice adequate log coverage or pay exorbitantly for it. A simple, scalable log analytics solution is a must-have to increase log coverage, improve DevOps efficiency, and isolate the root cause of app and service issues.

First, let's dive into a few typical use cases for log data in DevOps.

DEVOPS USE CASES FOR LOG DATA FOR SECURITY

Long-term log data can provide a wealth of information for DevOps teams, specifically when it comes to troubleshooting, improving the software development process, and tracking trends in application performance. Log analytics solutions provide detailed records of events that take place within the application and throughout the IT environment. While metrics and traces help DevOps teams determine what is broken or where an error is happening, logs reveal a deeper level of detail that can help developers troubleshoot and debug their applications.

Let's take a look at how log data can be valuable to DevOps, and what types of logs are typically the most useful.

Troubleshooting infrastructure and applications

As organizations adopt more cloud services and their cloud environments grow in complexity, they naturally produce more telemetry data—including application, system, and security logs that document all types of events. All cloud services and infrastructure components generate their own, distinct logs.

As mentioned previously, APM and observability tools are well-designed for short-term operational data, but struggle with higher log volumes and long-term retention. Because of this, it's common for DevOps teams to only retain a few weeks to around 30 days of data.

While 30 days of history is sufficient in many instances, even this volume can be high enough for data retention to become an issue. Why? APM and observability platforms typically "enrich" data during ingest, adding metadata that helps with alerting, operational tasks, and analysis. Unfortunately, this metadata also adds to data size. The volume of data being generated can mean even a month's worth of retention gets pricey.

JSON files can also add to this complexity and data explosion. JSON files with nested fields and arrays need to be "flattened" to organize the data into a tabular format for analysis. This can quickly increase database size. To keep data sizes manageable, DevOps teams face a choice: eliminate fields for analysis, or narrow their queries to a certain window of time. In either case, they're losing insights due to the poor log coverage.

FASTER TROUBLESHOOTING IMPROVES INCIDENT MANAGEMENT KPIS, SUCH AS:

- **Mean time to know (MTTK)**
- **Mean time to repair (MTTR)**
- **Mean time between failure (MTBF)**

Inadequate log coverage makes it difficult to determine the root cause of a persisting issue in the forensics process. Scalable log analytics, however, reduces some of the headaches associated with troubleshooting common cloud infrastructure and services issues. Uncovering these issues faster improves incident management KPIs, which include mean time to know (MTTK), mean time to repair (MTTR), and mean time between failure (MTBF), among others.

Cloud monitoring services often capture metrics, metadata and events that can help inform DevOps teams about the status of their applications and infrastructure. There are many types of logs used for troubleshooting cloud services and infrastructure.

Among them include:

- **Event logs:** These provide information about network traffic, usage and more. For example, event logs can capture login sessions, track activity on a network, and record application errors.
- **Transaction logs:** These log files list changes to a database or cloud storage environment, and are commonly associated with SQL Server transactions.
- **Message logs:** These logs document activity from messages such as email and chat.
- **Audit logs:** Audit logs may vary between applications and systems but typically capture events that show who did what, and how the system responded.

Optimizing DevOps efficiency

Log analytics can increase DevOps teams' efficiency by helping developers understand how their software is acting in the context of its environment. Rather than focusing on troubleshooting alone, developers armed with log analytics can detect anything from minor inefficiencies to major bugs that would impact the user experience.

As with the troubleshooting use case, log analytics can be combined with APM features including digital experience monitoring and real user monitoring (RUM). Log analytics enables developers to layer in historical data and understand persistent trends, empowering them with insights such as how many visitors are using the app, where they're spending the most time, and where they're encountering friction or experiencing bottlenecks.

In addition to identifying issues, log analytics helps teams predict how an application is expected to perform in the future, as the user base or other requirements continue to change and expand. This is particularly important for fast-growing SaaS companies that need to understand the interrelated components of their cloud-native applications and infrastructure.

As more SaaS companies turn to product-led growth (PLG) strategies to attract and retain users, log analytics can help speed up the software development lifecycle (SDLC) and help prioritize new features, functionality and fixes that have the biggest impact on the customer experience.

LOG COVERAGE GAPS ACROSS OBSERVABILITY TOOLS

If you're a site reliability engineer (SRE) or hold a similar role on a DevOps team, you know there's an observability tool for every purpose. From APM, NPM and infrastructure monitoring to purpose-built security information and event monitoring (SIEM)—these tools were created to give teams real-time visibility into their applications and environments.

Observability tools like APM, as one example, give DevOps teams the ability to measure the performance of application services, dependencies (databases, web services, caching, etc.), requests, and transactions, using telemetry data from installed agents that collect metrics and traces.

But despite their ability to monitor application performance and user experience, APM solutions aren't the only tool that DevOps teams need to achieve comprehensive application observability, monitoring, and analytics.

As containerized microservice architectures, such as Kubernetes (K8s), have rapidly become commonplace, log data volumes have exploded and APM tools have been exposed for their deficiencies in delivering log analytics at scale.

Here's where DevOps teams are identifying gaps in their APM tools:

Limited Data Retention: Since APM tools are real-time or near-real-time operational solutions, they are designed for fast analysis of recent (less than 30 days old) and smaller data sets. If a DevOps team wants longer retention, it becomes expensive. Some APM tools even cap retention at 12 to 13 months. While DevOps teams can use APM tools for short-term analysis of operational data, their limited data retention creates analytical blind spots that negatively impact use cases like root cause and long-term trend analysis.

High Storage Costs: APM tools often add metadata and parse log data into record fields to support analytics initiatives. This practice inflates the aggregate size of log data, leading to higher data storage costs. DevOps teams are ultimately forced to choose between shouldering these inflated costs or reducing them by decreasing log retention.

Data Movement + Multiple Data Copies = Increased Costs: APM tools often require DevOps teams to capture and aggregate logs in cloud storage before sending them to a SaaS APM tool. This results in data storage and egress costs from the cloud service provider, plus data ingest and storage costs from the APM SaaS provider. As log volumes grow, these costs become prohibitive, which eventually leads to data retention trade-offs.

Data Privacy and Security Compliance: DevOps teams that are required to retain ownership and control of sensitive data can encounter compliance challenges that result from moving data between cloud storage and a SaaS APM platform.

When it comes to log analytics at scale, APM tools simply aren't optimized to efficiently aggregate, query, and analyze the massive volume of logs generated in today's complex application environments.

To cover these gaps in log coverage and better support use cases like root cause and long-term trend analysis, DevOps teams can implement a log analytics solution that supports cost-effective storage, querying, and analysis of log data. These solutions can complement existing observability tools, and provide deeper insights into performance issues, user behavior, and trends over time.

CURRENT CHALLENGES WITH ELASTICSEARCH FOR LOG ANALYTICS

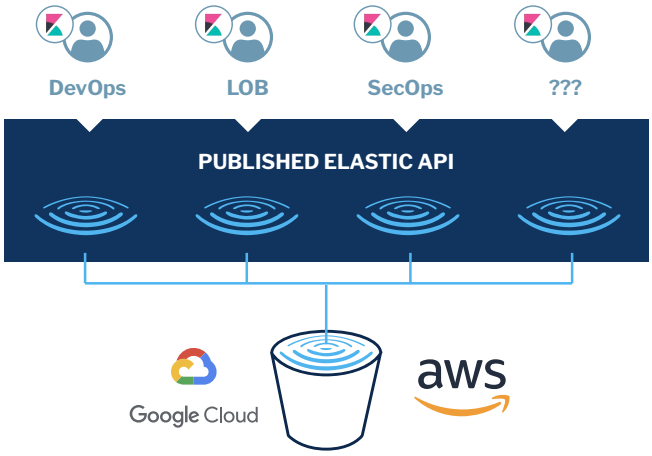
The explosion of log volume and variety requires a dedicated log analytics platform, rather than stretching the limits of an APM/observability platform's retention capabilities (and paying exorbitant costs). For the last decade plus, Elasticsearch has been the platform of choice. Whether an on-premises ELK stack (Elasticsearch, Logstash, Kibana) or a managed Elasticsearch service, the older technology foundation makes Elasticsearch difficult to manage and expensive at scale. Here's why:

- **Painful cost vs. retention tradeoffs:** Elasticsearch, whether it's self-managed or managed service, tightly couples compute and storage. Customers are forced to either reduce log volume or shorten retention windows to keep costs down. For DevOps teams, this can result in poor log coverage.
- **Management burden and availability/durability issues:** Managed service or not, the tight coupling of compute and storage means support teams must plan for and constantly adjust their Elasticsearch cluster. The complex architecture causes availability issues. Not to mention, support from service providers is often lacking. It can also take a long time to bring a failed cluster back up, which frustrates users who can't access data when an issue arises.
- **Complex pipelines:** To work around Elasticsearch's architectural issues, admins will:
  - **a:** reduce the amount of data they store (either fields or retention);
  - **b:** store data in different storage tiers (such as warm and cold storage, typically in S3) which affects performance; or
  - **c:** create a backup pipeline (typically to S3) to make sure there's no data loss. Managing these pipelines becomes complex and costly.
- **Limited data retention:** The tiered storage system described above leads to loss of insight because customers can't store all the logs (or attributes) they need. This creates different tiers of analytics and different classes of analytics users. Short-term data is kept in hot storage where it can be accessed with high performance, and long-term data is stored in slower, less accessible cold storage.

Teams often try to work around Elasticsearch's complexity and cost by deploying more and more separate clusters. The result is a complex and expensive architecture that satisfies no one.

Rather than relying on complex workarounds and data movement, there are ways to activate your existing cloud object storage as a hot analytics environment. Indexing your data within Amazon S3 or Google Cloud Storage can make your data fully searchable and available for analysis, using existing data tools such as Kibana. This approach empowers you to analyze data in place, without having to move data or create complex pipelines.

For instance, ChaosSearch offers a data lake platform for this purpose.



SIMPLIFYING LOG ANALYTICS WITH CHAOSSEARCH

The ChaosSearch platform activates cloud object storage for log search and analytics at scale. ChaosSearch creates a unified data lake on top of an existing AWS or GCP cloud object storage environment. Organizations can store and analyze all their data, without any transformation or data movement.

ChaosSearch users have reported up to **80% cost savings** over Elasticsearch and other, similar tools.

There are three core components of the platform:

Chaos Fabric®

A serverless computational fabric based on a containerized, distributed architecture that orchestrates the Chaos Index.

Chaos Index®

A distributed database that discovers, normalizes, and indexes data (by roughly 20x) without human intervention. Chaos Index uniquely supports both text search and relational queries, and is optimized for cloud storage.

Chaos Refinery®

An in-app tool that transforms data virtually and instantaneously, without needing dedicated resources or DBA skills.

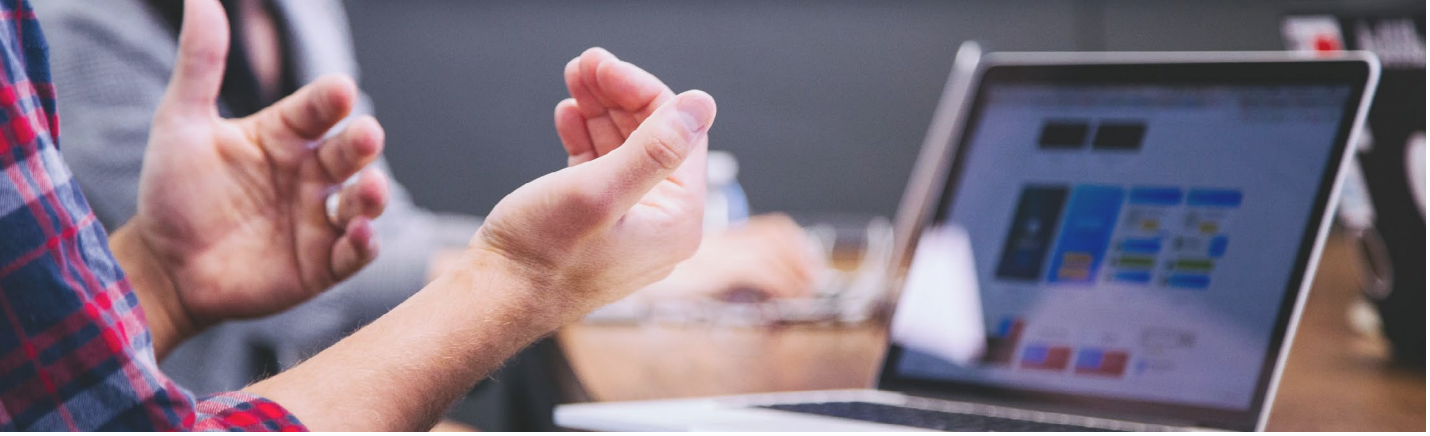
Getting connected to ChaosSearch to get value from the data in your data lake takes a few easy steps. Once you have data in an Amazon S3 or GCS storage bucket—which most organizations already do—with the click of a button, you can connect ChaosSearch and configure read-only access. With another click, Chaos Index will index your data with automatic schema detection and normalization.

Just as easily, Chaos Refinery will virtually aggregate and transform indices. You don't need to move or transform any data physically, and your full role-based access controls (RBAC) set within your cloud environment remain intact. ChaosSearch supports Elastic APIs so you can directly replace Elasticsearch without any change in user behavior, and includes embedded Kibana, or users can use other preferred tools via API access. In addition, ChaosSearch provides a SQL API for relational analytics via tools such as Looker or Tableau. This means that data consumers can continue working in their tools of choice, and no longer have to worry about multiple data tiers or waiting for access to certain datasets.

Collapse the pipeline with Chaos Index

JSON files, a common format for DevOps logs, can be composed of hundreds of nested fields, or attributes, within a single file. This makes JSON logs difficult to query, as data engineers often have to flatten these files into the typical structured databases made up of columns and rows. These databases can quickly become massive and costly to store. Either that, or a data engineer needs to write complex queries that can run against each nested file.

Chaos Index detects and indexes JSON data automatically, without any configuration required from the user. During indexing, field names from JSON objects are conceptually translated as columns of a table, while the field values are translated into row data. This gives DevOps teams unlimited flexibility to experiment freely with their JSON data, without having to request new queries or pipelines from the data engineering team every time.



THE BUSINESS CASE FOR LOG MONITORING AND ANALYTICS

Ultimately, optimizing your apps and infrastructure doesn't just impact the DevOps team. It impacts the entire organization. In PLG-driven companies where SaaS apps are central to the business, log analytics are business-critical.

For example, Marketing teams can correlate app usage with customer accounts to gain a better understanding of relevant campaigns or offers. Customer chat sessions with Support teams can reveal new data on why customer satisfaction scores have gone down, or why customer churn is happening. And, as previously mentioned, DevOps teams can speed up the SDLC and prioritize new features, functionality, and fixes that have the biggest impact on the customer experience.

Gaining these insights is a matter of simplifying the process of accessing and analyzing your log data at scale.

Want to learn more about the ChaosSearch platform?

Watch the demo

ABOUT CHAOSSEARCH

ChaosSearch enables customers to Know Better™, delivering data insights at scale while achieving the true promise of data lake economics. The ChaosSearch Data Platform connects to and indexes data within a customer's cloud storage environment, rendering it fully searchable and available for analysis with existing data tools—all with unlimited scale, industry-leading resiliency, and massive cost savings.

Based on these capabilities, ChaosSearch is an ideal replacement for the commonly deployed ELK stack today. With ChaosSearch, customers can perform scalable log analytics on AWS S3, using the familiar ElasticSearch API for queries, and Kibana for log analytics and visualizations while reducing costs and improving analytical capabilities.

We'd like to hear from you about your log management challenges and priorities. For any questions or requests, or to simply learn more, visit us online or send us an email.

info@chaossearch.com | **www.chaossearch.io**