MEZMO EBOOK

# Logging 101: Logging Fundamentals for the Cloud Native Era

mezmo

# INTRODUCTION

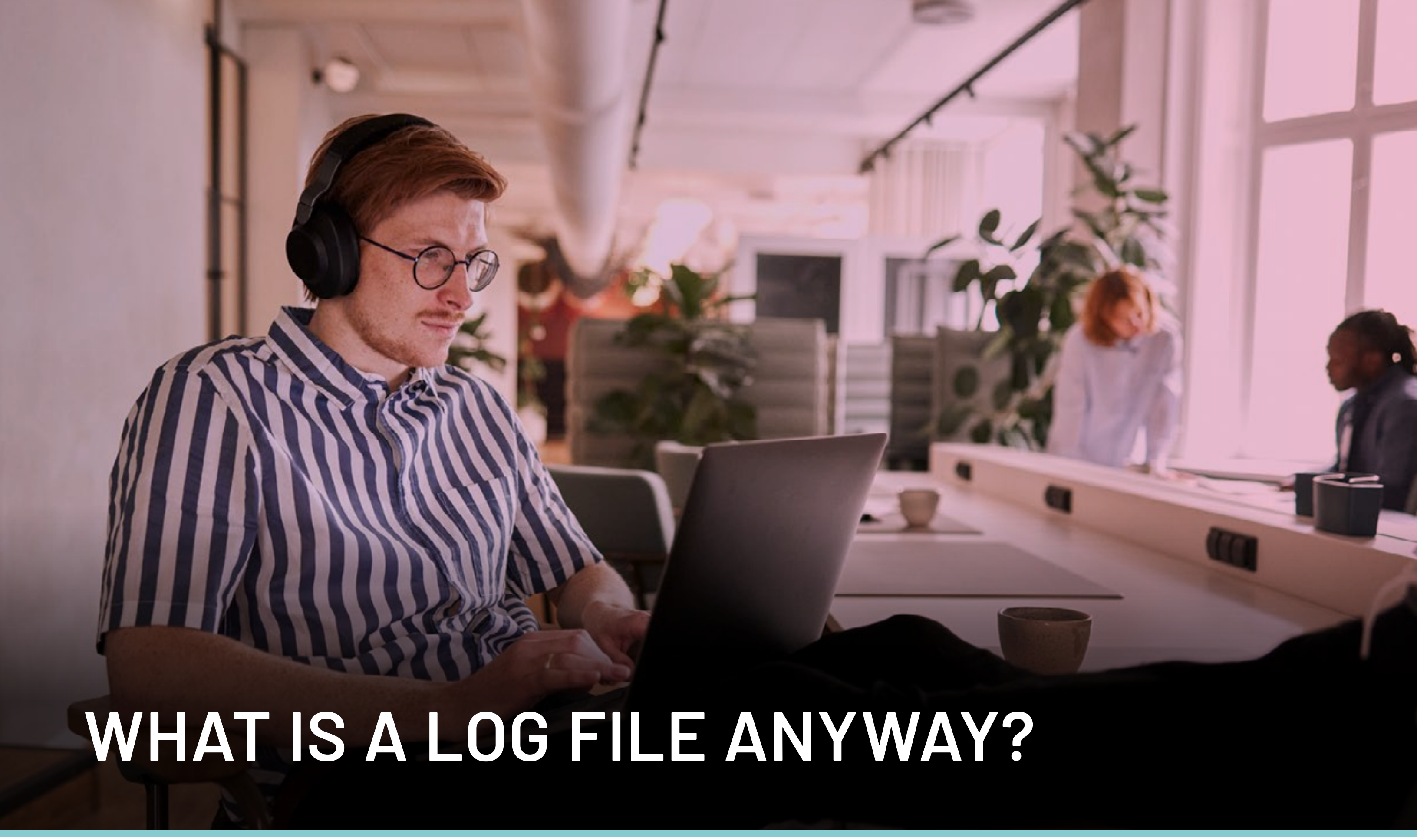
For IT professionals, logging can feel akin to saving for retirement or eating fruits and vegetables: You know it's something you should do. But understanding exactly why and how to do it — and finding the motivation to ensure you do it properly – is where things get complicated.

What's more, given that the complexity of applications and applications architectures has evolved significantly over the past decade, understanding how best to approach logging has become even more mysterious for many teams. Logging may have been straightforward enough in the age when we deployed applications as monoliths running directly on the host operating system. But today, the typical application is composed of a series of microservices deployed in containers and orchestrated by a platform like Kubernetes. Add multiple clouds, service meshes, and APIs to the mix, and figuring out what to log and how to log it becomes challenging indeed.

The good news is that teams can master this complexity. Doing so starts with understanding the fundamentals of logging in the cloud-native era – which is what this eBook will help you do.

The following pages explain why logging is important, how logging works and how to get the most out of log data through centralized logging and analysis. Although many takeaways apply to any application or environment, the discussion focuses on modern, cloud-native applications. By reading through the following chapters, then, you'll learn the core building blocks of an effective logging strategy for any software environment you might manage today.

# TABLE OF CONTENTS



# WHAT IS A LOG FILE ANYWAY?

Let's start with the very basics: Defining what a log file is, where you find log files in modern systems and how you can ingest them into log analysis and observability tools.

## Log Files in the Cloud Native Age

As IT systems become more sophisticated and the volume and velocity of data grow, it's more important than ever to have a logging system that can provide valuable insights in a real-time, time-series fashion. Identifying which log files provide the most value may be challenging, so you may wonder which ones to ingest. For example, some of these logs might include operating system metrics that your applications run on or appliance data that secures your internal systems. They might not seem valuable as independent data points, but they can collectively tell a much bigger story when aggregated together.

## What Is a Log File?

A log file represents a record of an action taken on a system within the environment. These actions can include rebooting the server or consuming resources such as CPU and memory. Log files can range in size from kilobytes to gigabytes, and the events they record typically have a timestamp so that you can identify when they occurred.

As your system grows, these log files will become impossible to manage without a centralized logging system. Companies scale horizontally by adding more machines to the resource pool to get better throughput — a process that exponentially increases the number of log files that the logging system generates.

## What Types of Log Files Are There?

Log files come in all shapes and sizes, including XML, JSON, CSV, and simple key-value pairs. They can be operating system logs, application logs, performance monitoring logs, firewall logs, ticketing system logs, or audit trails logs. Since they vary in format, it's challenging to identify timestamps and line breaks when onboarding them to your logging system. When aiming to decrease time to value (TTV), choosing a logging system that can structure your data in a machine-parsable format is essential.

## Will Log Ingestion Continue to Increase in the Coming Years?

As competition grows, companies have to keep innovating to stay relevant in the marketplace. Therefore, businesses often take on more projects and initiatives to capture market share, increasing log volume and velocity.

Many companies struggle with forecasting the optimal way to account for this growth, so they determine which logs have the most significant value and then onboard only those logs. Since companies are often unaware of other options, it's understandable to take this approach.

There are, however, ways to be clever, such as preprocessing data that are already in motion to reduce log size and collect more data that will provide deeper insight into what the environment is doing. Preprocessing, in this case, includes flattening XML files, removing whitespaces, and converting to clean key-value pairs.

## What Is Observability?

We can define observability to measure how well we can infer a system's internal state based on external outputs. Observability is different from monitoring in that it allows you to ask questions based on a hypothesis. In contrast, monitoring is more passive and will only alert when it sees a defined problem.

Most companies quantify their success according to service availability, and they usually compare current availability to past availability to understand how well they are doing. It can become exponentially more challenging to improve these availability numbers over time without innovative ideas. By utilizing an observability platform, you can ask questions from the outside and tune internal processes to get the desired result, resulting in reduced Mean Time to Resolution (MTTR) for critical business incidents.

## Which Log Files Do You Need to Build an Observability Platform?

When building an observability platform, the type of log data you're currently ingesting affects the success of your project. It may seem like some types of logs are not crucial to digest because you don't think they're impactful. Still, when you tie them all together into a single solution, those supposedly low-value logs can paint a high-value picture that improves your overall monitoring posture. Having structure (schemas) gives your organization the building blocks it needs to build an observability platform. So, if you're wondering which log files you should ingest, the answer is all of them.

# THE KEY BENEFITS OF LOG DATA

Now that we've discussed what log data entails, let's move onto the second key question that teams must be able to answer to build an effective logging strategy today: Why does log data matter?

When utilized effectively, the data collected by logging can be of great value to developers, IT personnel, and business folks alike. Log data can help organizations find problems within their applications at the earliest possible moment. Additionally, it can enable developers and incident responders to resolve issues more quickly, and it can provide critical insight into how people are using applications.

Below, we discuss these benefits and delve into what makes each of them extremely valuable to DevOps teams.

## Log Data Can Help Reduce MTTA

Log data can play a crucial role in enabling DevOps teams to identify problems within their applications quickly. Accomplishing this is often made possible by using log management platforms that feature real-time log analysis capabilities and alert functionality. The platform ingests the log data and analyzes it immediately while configuring alerts to notify the necessary personnel if the resulting analysis reveals an issue. For example, an organization might encounter an uptick in the error rate for a web application. In this case, their web server logs might indicate higher than average volumes of HTTP status codes indicative of failure. Using log management software and real-time log analysis, this rise in error rate can trigger an alert, enabling the response process to begin at the earliest possible point.

Leveraging log data in this manner leads to reduced Mean Time To Acknowledgement (MTTA), a key metric for measuring the effectiveness of an incident response strategy. In other words, by reducing the time it takes to realize that a problem exists, log data enables responders to begin working to resolve the issue earlier. Downstream, this limits application downtime and the overall impact of an incident on end-users.

## Leveraging Log Data Helps Teams Optimize Root Cause Analysis

Log data can be beneficial for pinpointing the root cause when problems occur within an application. For instance, helpful information such as the full stack trace is typically recorded in the application's error log when the system throws an exception. This data enables developers to trace through the method calls that led to the problem and identify the exact line of code that triggered the exception, making the issue easier to research and the problematic scenario easier to reproduce. Thus, responders can understand the problem, which enables them to reach a complete and permanent solution.

Efficient root cause analysis is a critical component of an effective incident response process. So, by serving as a valuable resource for determining root cause, log data assists in minimizing another key incident response metric: Mean Time To Resolution (MTTR). Similarly, reducing MTTR further helps in limiting the impact that incidents have on end-users. Moreover, when it takes engineers less time to remediate problems, they can focus more on building new and exciting functionality, which further increases the product's value to the customer.

## Log Data Helps Teams Gain a Better Understanding of Application Usage

While log data is a valuable asset for resolving failures within applications, it can also be beneficial for other reasons. For example, developers can use log data to gain complete insight into how users use an application. In a web application, developers can analyze request logs to reveal significant key trends and patterns for the organization, such as determining when a web application is subject to higher traffic. In addition, this analysis can reveal information as far as which content your users are accessing most often and which browsers they are using most often to access it.

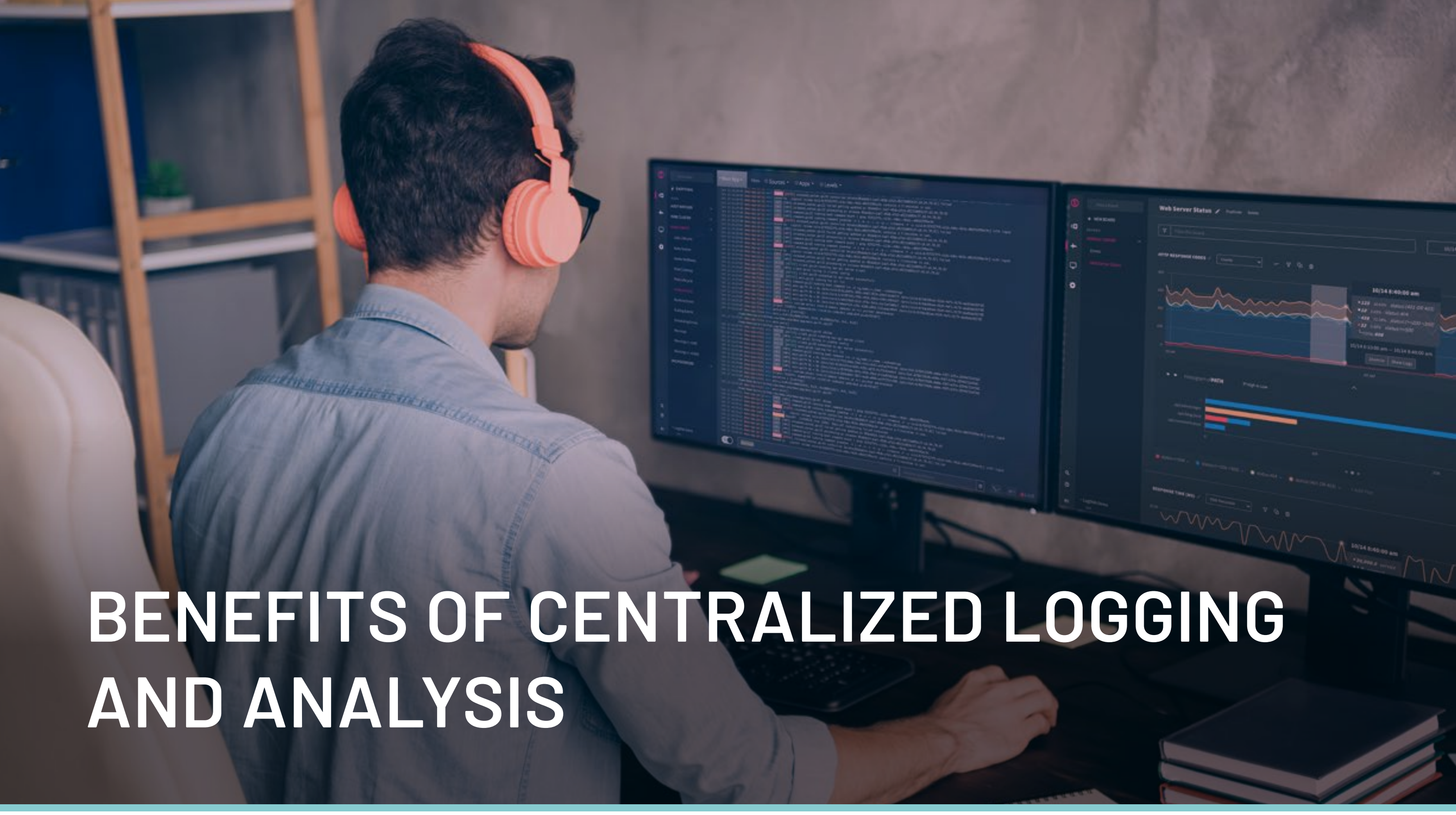
Another way log data can shed light on application usage is through the analysis of audit logs. Audit logging provides a way to evaluate user actions within an application, usually for security purposes. Audit logs typically record login and logout actions and details about when and how someone manipulates data within a system and who they are. This information gives organizations a significant advantage in that they now have a mechanism for identifying, tracking, and (potentially) reversing unauthorized data changes. In other words, organizations can leverage this mechanism to improve security and limit damage when incidents occur.

## Why Modern Teams Need Logs

Log data can be beneficial to DevOps organizations in several key ways, such as:

- By leveraging log management tools that feature real-time analysis capabilities and alert functionality, organizations can use log data to help streamline recognizing incidents at the earliest possible moment.
- Log data can be an invaluable resource for determining the root cause. Organizations can resolve incidents more quickly, allowing development teams to spend less time fixing problems and more time building new, valuable functionality.

Log data enables DevOps teams to gain crucial insight into how users use their applications, critical urgency for most businesses, you can't understate the value of centralized logging as a means of driving collaboration and shared visibility.



# BENEFITS OF CENTRALIZED LOGGING AND ANALYSIS

So far, this eBook has explained why log data and analysis are essential in general. But the fact is that individual log files alone are of little value, especially in the context of modern, distributed applications. What teams need to achieve observability is centralized logging and analysis. By combining multiple log files and analyzing them from a central location, businesses double-down on log data's value.

Here are crucial benefits that teams can achieve by centralizing their approach to logging and analysis to prove the point.

## Correlating Events Between the Application Layer and Infrastructure Layer

We can generally break down software environments into two fundamental parts: applications and the infrastructure that hosts them.

Without centralized logging, it's hard to know how an event in one of those layers impacts the other layer. You might parse your infrastructure logs and detect that your server has maxed out its CPU, for example. But if you analyze the application logs separately, it's challenging to determine the impact of high CPU utilization on applications running on the server.

Sure, you could go and compare the infrastructure and application logs manually to correlate events. But that's inefficient, and it doesn't scale. If you centralize all of your logs automatically and by default, you can connect events across all layers of your environment instantly and continuously.

## Identifying Trends with Centralized Logging Software

How can you tell whether an event like an uptick in error rates in an application or a slowdown in response times is an isolated issue or part of a broader trend? Ideally, you'd look at all of your logs from a central location to determine whether similar events have occurred elsewhere.

You'd also look at historical log data centralized in the same place to compare current trends to historical baselines, which would provide you with additional context for distinguishing between random fluctuations and significant trends.

You can do both of these things when you centralize your logs automatically. Without centralization, it would be virtually impossible to identify trends efficiently.

## Using Centralized Logs to Identify Over-Allocation of Resources

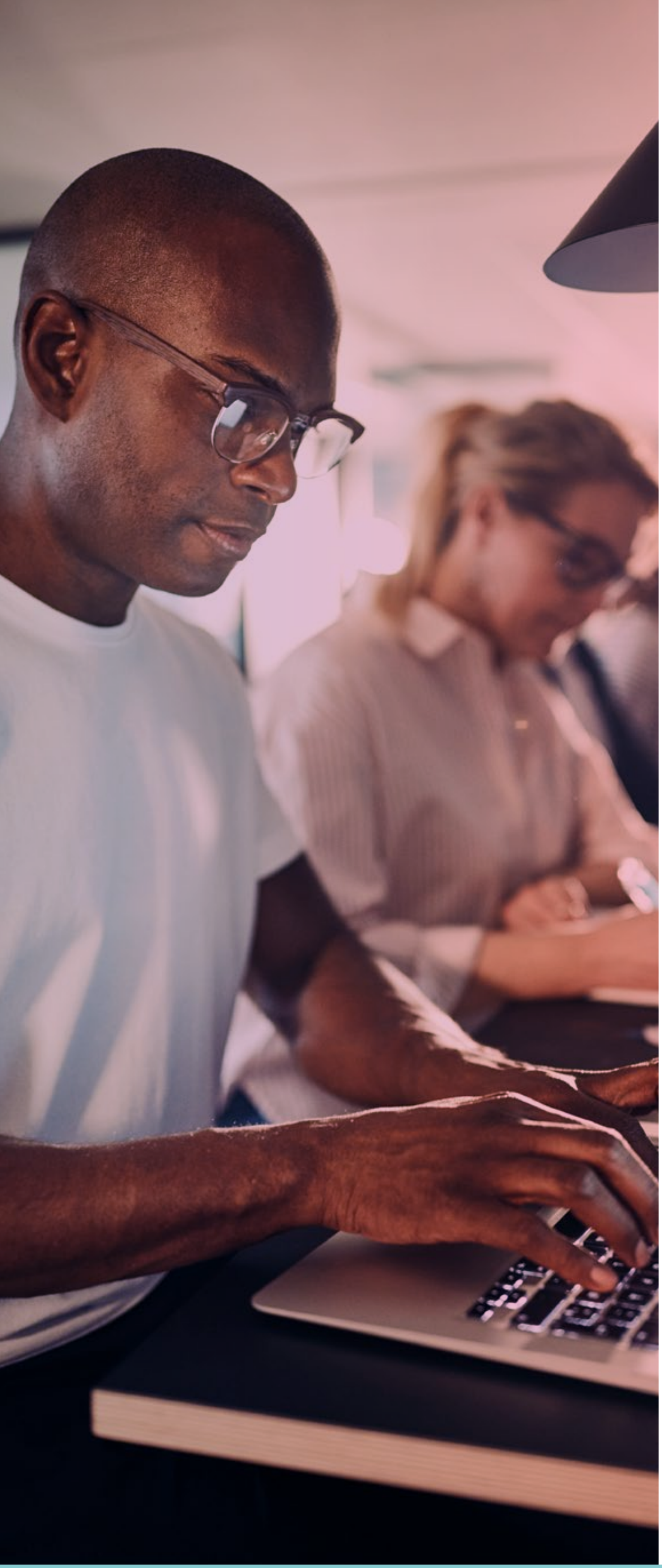
IT teams tend to spend most of their time worrying about applications that don't have enough resources to perform well. But an equally problematic event – especially in the age of the cloud, where most companies bill resources on a pay-as-you-go basis – is when you have allocated more of them than you require. In that case, you need to scale down to avoid wasting money.

Here again, centralized logging and analysis is the key to recognizing when to scale resource allocations down. Using features like Mezmo Kubernetes Enrichment and Presence Alerts, you can quickly view data about resource consumption alongside application performance data. You can also track resource consumption patterns over time, allowing you to safely scale back your allocations, all while continuing to follow application performance to verify that allocation changes don't harm your applications.

## Reducing MTTD and MTTR Through Centralized Log Analysis

The Mean Time to Detect (MTTD) and Mean Time to Resolve (MTTR) metrics are crucial for your customer experience. The longer it takes to find and fix problems, the less pleased your end-users will be.

When you have to toggle manually between multiple logs to confirm that an anomaly is a problem and then keep toggling as you investigate and respond to the issue, MTTD and MTTR are likely to remain high. But with centralized logging and analysis, you can interpret data and assess complex events efficiently, leading to lower MTTD and MTTR as well as happier customers.



## Ensuring Consistency and Collaboration Across Teams

Last but not least, consistent, standardized log centralization can help multiple teams work together uniformly.

For example, suppose your developers centralize logs from dev/test environments in the same place where IT teams manage logs from production (while differentiating between log sources, of course). In that case, it's easier for both teams to collaborate. Each group has visibility into what the other group sees in its part of the software delivery chain, and everyone can work toward shared goals and metrics tracked from the same centralized logs.

In an era when "breaking down silos" has assumed critical urgency for most businesses, you can't understate the value of centralized logging as a means of driving collaboration and shared visibility.

## The Many Benefits of Log Centralization

Log centralization is not something you do just because companies or myself told you to do it. Like brushing your teeth, it offers a range of critical benefits, albeit different ones than those associated with good oral hygiene. If you can't manage and analyze logs from across all layers of your environment centrally and automatically, your teams will struggle to operate efficiently and create business value.

# CONCLUSION

Logging is a complex topic, and this eBook has touched only on the most basic fundamental concepts and practices within the realm of logging. However, the basics are the best place to start when dealing with any complicated topic. By understanding, applying and building upon the lessons from the preceding chapters, modern development, IT and DevOps teams place themselves in the strongest position to gain deep observability over the complex applications that they manage in today's cloud-native world.

## About Mezmo

Mezmo is a comprehensive platform to control all of your log data. It enables teams to ingest and route massive amounts of log data, from any source to any source. This capability fuels enterprise-level application development and delivery, security, and compliance use cases, where the ability to use log data in real time.

mezmo

# Thank You

| | |
|---|---|
| Sales Contact: | outreach@mezmo.com |
| Support Contact: | support@mezmo.com |
| Media Inquiries: | press@mezmo.com |