logdna.com

logdna

LOGDNA EBOOK

Using LogDNA to Troubleshoot In Production

LogDNA gives users access to the information they need to effectively troubleshoot in production.

INTRODUCTION

Traditionally, logging was most commonly associated with the post-deployment part of the software development lifecycle, or SDLC. Logs typically served first and foremost to help IT engineers find and troubleshoot problems that arose in production.

Today, however, logging can help teams optimize much more than just production-environment application management. And indeed, logging needs to be leveraged across all stages of the SDLC in order to ensure the reliable, continuous delivery of software. Developers, testing teams, and anyone else involved in software delivery must make use of logs and log analysis as one way to ensure the smooth flow of code across the entire SDLC.

With that reality in mind, we've prepared this guide to showcase practical approaches to log analytics at different stages of the SDLC.

In our series of eBooks, you'll find an explanation of why logging across the SDLC is essential in modern software delivery chains, as well as real-world examples of how teams can use LogDNA to streamline three distinct stages in the SDLC: Development, QA and staging, and production troubleshooting. This eBook is focused on production troubleshooting.

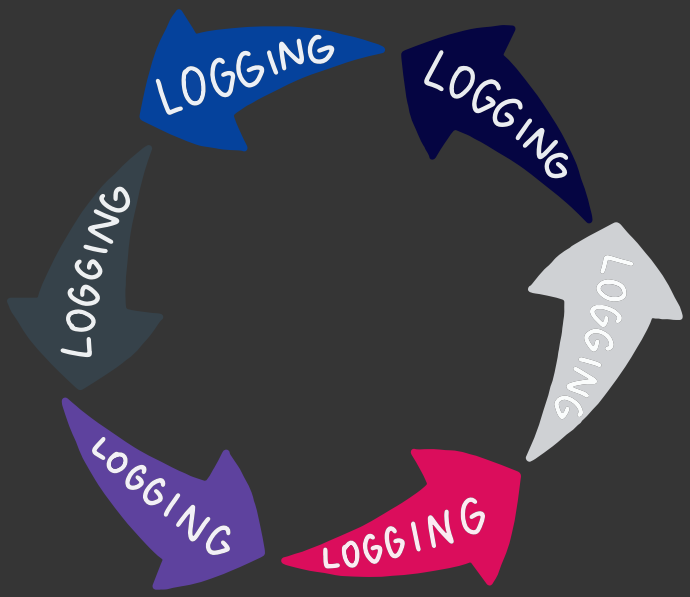


TABLE OF CONTENTS

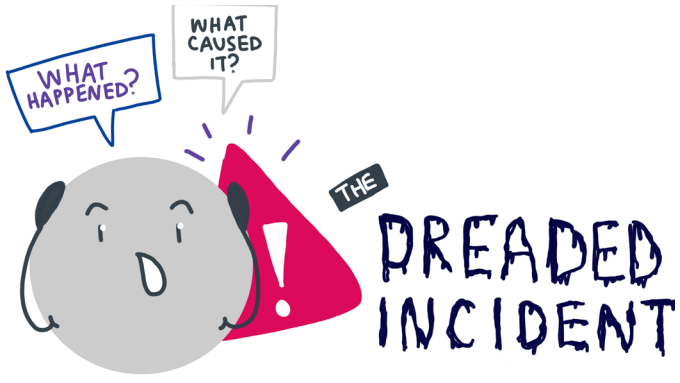
USING LOGDNA TO TROUBLESHOOT IN PRODUCTION

In 1946, a moth found its way to a relay of the Mark II computer in the Computation Laboratory where Grace Hopper was employed. Since that time, software engineers and operations specialists have been plagued by "bugs." In the age of DevOps, we can catch many bugs before they escape into a production environment. Still, occasionally they do, and they can spawn all kinds of unexpected problems when they do.

In addition to software bugs, today's modern systems encounter other problems as well. And these problems collectively manifest as issues in production. Or, as some might refer to them, the dreaded incident!

When an incident occurs, it's not always readily apparent why it happened or what caused it in the first place. This chapter explores some of the different situations that can result in a production incident, and we'll investigate how you can uncover these situations using LogDNA. Troubleshooting a production incident shares many similarities with how a medical professional might approach a patient's diagnosis. We look at the symptoms, we run tests, and we reach a diagnosis by drawing on our experience, the experience of others, and sometimes even relying on the process of elimination.

Detection and Recovery



Production incidents have two distinct phases: detection and recovery. Many teams track ***Mean Time To Detection (MTTD)*** and ***Mean Time To Recovery/Resolution (MTTR)*** metrics. MTTD is a measure of how long it takes for a problem or incident to be identified and acknowledged by the responsible team. MTTR then measures the amount of time before the team can identify and rectify the underlying cause.

This eBook is not about tracking metrics; however, it is essential to measure and track their effect on our results when we're investigating and investing in new ideas and tools. Metrics such as MTTD and MTTR provide visibility into how effectively the organization manages the incident process and how valuable different approaches and toolsets are within that process.

The Complexity of Modern System

Modern computer systems have begun to rely on microservice architectures, patterns for high-availability, and public cloud offerings. We build highly scalable systems that take advantage of ever-improving connectivity and the availability of third-party infrastructure and services. While the benefits are apparent, we find ourselves managing increasingly complex systems. Let's consider the example of a web application that becomes unresponsive.

Networking Problems

Between the user and our web application, there is a complex web of connections running on private and public infrastructure. Service outages and hardware failures can easily affect a user's ability to connect to our systems.

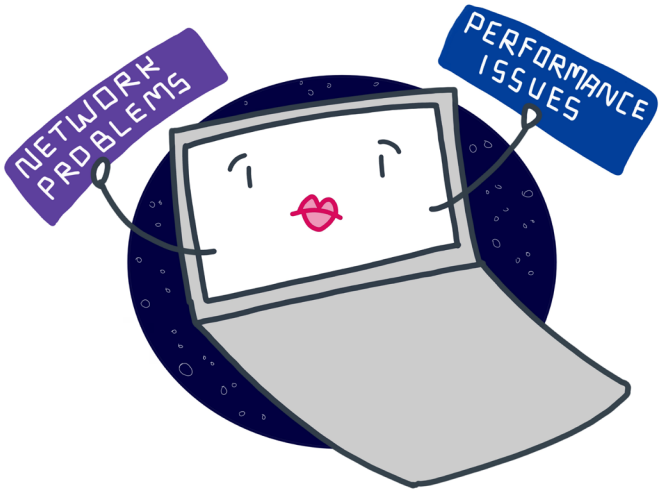
Even within our systems, network calls travel between different instances and services. Our applications and services may be running perfectly, but if the connections between them are broken or experience performance degradation, the system won't work as expected.

Performance Issues

Performance issues can occur due to hardware problems, constraints in frameworks and infrastructure we rely on, or poor design within the code itself. As the amount of traffic our applications handle increases, many of these problems grow exponentially, resulting in performance degradation or connection timeouts.

Why Centralized Log Management is Essential

Logging has always been a central component of



understanding and troubleshooting applications. System and application logs provide an essential window into how data is processed and transferred, as well as the infrastructure's performance. Distributed systems complicate log management because there are so many different places where logs are created and stored.

Centralized log management solutions, like LogDNA, give

you the ability to collect and aggregate all of your logs in a central location. However, collection and aggregation are just part of the solution. Search capabilities allow you to find relevant logs to help troubleshoot production problems and monitor the health of different parts of your application. Many systems also include monitoring and configurable alerts to automatically identify problems and anomalies, and automatically alert support personnel to address them.

Comprehensive Alerting

The most common types of alerts identify error conditions within your application. More advanced systems like LogDNA allow you to specify the type of error and its frequency as part of the alert. In addition to error states, you can also configure an alert based on the results of a predefined query against your logs.

In addition to alerts based on the presence of an error condition or log query, you can also configure alerts based on the absence of certain types of logs. An example might be an online commerce site that expects a minimal level of transactions each hour. A lack of these transactions might indicate an access problem within the application.

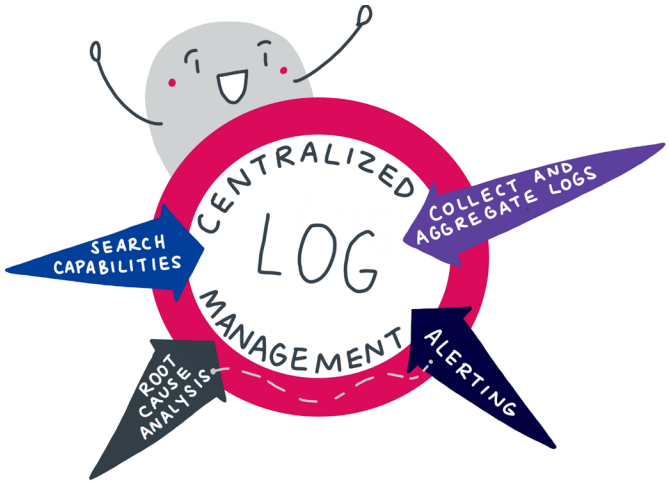
When used together, alerts for both the presence and absence of different conditions relieve your engineers of the responsibility to regularly review logs, and they can focus on building new features or improving their processes. You can learn more about alerting from the LogDNA Alerts Overview.

Root Cause Analysis

Once your support teams have received an alert from the log management system or a production incident has occurred, teams need to shift into resolution mode. The first step in the process is a root cause analysis. You can't resolve a problem until you have identified the reason it occurred and each of the components involved.

Identifying a problem in a distributed system can be incredibly challenging because you first need to identify which service is causing the problem and which services are affected by the situation. This investigation is where the ability to search through an aggregated collection of logs using request or transaction identifiers, time constraints, and additional filters is invaluable. You can learn more about how to search logs in the LogDNA Search Documentation.

Once you've identified the problem and worked on a solution, you'll typically want to deploy the solution while actively monitoring the logs. LogDNA's Live Tail lets you monitor logs as they are received from your application, giving you real-time visibility into the new deployment status and allowing you to validate its success or failure. You can also use the log management system to aggregate the logs and perform searches after time has passed.



CONCLUSION

In this eBook, we've shown how to leverage logs and LogDNA for production troubleshooting. LogDNA can help optimize other SDLC stages including development and QA and staging, which are discussed in the other eBooks in this series. No matter which stage of the SDLC you help manage, or which challenges you face, logs are one key resource to help you do your job better. And in a world where teams are expected to deliver new application releases multiple times per week, or even per day, engineers need every insight and data point available to them to keep the delivery pipeline flowing smoothly.

logdna

Thank You

Sales Contact: outreach@logdna.com
Support Contact: support@logdna.com
Media Inquiries: press@logdna.com