LOGDNA EBOOK

The Tech Stack for Collaboration

logdna

INTRODUCTION

In recent years, companies have increasingly recognized collaboration as an essential ingredient in business success. By building organizations where employees from across departments and ranks can readily engage and collaborate, businesses can boost productivity, enhance employee morale and reduce friction within business operations.

If you follow the conversation surrounding collaboration within organizations, you're likely already familiar with cultural and structural techniques that companies can use to encourage collaboration across their teams as a whole.

But full collaboration requires more than just achieving a specific culture or structuring teams in a particular way. It also depends on giving your stakeholders the right tools to implement collaboration.

Chief among those tools is centralized logging. Although logging systems may not be one of the first items you consider when you think about collaboration, the ability to collect and share log data across teams is essential for sustaining a culture of collaboration – especially among technical employees.

This eBook walks through the role that logging and related DevOps technologies play in collaboration. It starts by explaining why achieving collaboration is so crucial for modern businesses. It then discusses the specific ways in which logging and other DevOps tools can enable a culture of collaboration at businesses of all types.

Although a complete discussion of collaboration in the modern organization is beyond the scope of this eBook, the following pages highlight one easy-to-overlook facet of collaboration tooling – centralized logging.

TABLE OF CONTENTS

WHY COLLABORATION TECHNOLOGY IS CRITICAL IN 2021

Which technical solutions do businesses need to implement to thrive in 2021? Ask most people in the IT industry that question, and you'll probably hear answers that align with trendy technology topics — like hybrid cloud, AI, and hyper-automation.

Those may well be among the keys to success in the future. But here's another critical area of focus: collaboration. Though it may seem mundane, collaboration has assumed unprecedented importance for a business's ability to thrive in the modern IT landscape.

To be sure, the importance of collaboration is nothing new in a conceptual sense. For decades, if not centuries, businesses have needed to ensure effective collaboration between employees, partners, customers and other stakeholders.

Today, however, the ability to collaborate efficiently and effectively has reached a new level of urgency. To prove the point, here's a look at several major IT trends that have made collaboration solutions a must-have asset for 2021 and beyond.

Permanent Distributed Workforces

In 2019, only about 3.4 percent of U.S. employees worked remotely. Two years and one pandemic later, analysts estimate that more than 20 percent of the workforce will work remotely well into the middle of this decade.

The takeaway here is clear: from now on, a significant number of employees will work remotely, at least part of the time.

In this context, companies that make collaboration tooling need to change the landscape.

They shouldn't assume that most employees work face-to-face and only need collaboration solutions to connect some of the time remotely. If anything, they should operate on the premise that employees are remote most of the time and only occasionally — if ever — on the same physical site.

Distributed workforces also set the bar higher for the performance of collaboration tools. When your employees spend all day in front of video conferencing software, it's not okay if call quality is just okay. Collaboration software vendors and the IT teams who support their platforms need to deliver top-notch user experiences routinely.

Ever-Increasing Customer Expectations

Customers are a demanding bunch, and they're growing increasingly exacting in their expectations.

A decade ago, five seconds was the baseline for how long users would wait on a website to load before becoming frustrated. Today, that number is closer to two seconds.

Likewise, according to Gartner, more than two-thirds of businesses now compete based on customer experience, compared to a little over one-third in 2010.

What do statistics like these have to do with collaboration for IT organizations? In short, everything. If developers can't collaborate effectively, they can't deliver software continuously to regularly roll out new and improved applications. Likewise, IT engineers and SREs (site reliability engineers) must collaborate seamlessly to maintain a smooth customer experience when problems arise within software environments.

More Complex Applications

Compounding the challenge of maintaining a positive customer experience, IT organizations now have to manage more complex software systems than ever.

Instead of monitoring a few virtual machines and monolithic apps, as was the norm about a decade ago, teams must now keep tabs on dozens of microservices spread across dozens (or possibly hundreds) of containers and servers. There are orchestration layers to manage now, too. Add what was mentioned above to how organizations may spread all of these IT resources across multiple clouds, on-premises servers, or both, and you start to appreciate just how complex software environments have become.

Collaboration is essential to keep track of what is happening within these complex environments, ensure that the right stakeholders have access to the correct data, solve problems, and improve the software over time. You can't survive — let alone thrive — in managing modern software architectures if your collaboration strategy consists of ad hoc email chains or bi-weekly sit-downs. You need fast, efficient, comprehensive collaboration solutions that allow team members to connect anytime and anywhere.

Pervasive Security Threats

As software has grown more complex, so have the challenges associated with keeping it secure. 2020 was a record-setting year for the volume of cyberattacks, and it looks like 2021 will be no different.

And it's not just the rate of cyberattacks that is increasing. The complexity of attacks is growing, too, as threat actors become ever more creative in the way they launch attacks — like sophisticated phishing schemes and target software supply chains — as a way to compromise assets.

Faced with challenges like these, how can businesses keep their IT resources secure? Part of the answer is integrating security into the application lifecycle so that software is tested and validated for security initially — rather than only after production. Another is to ensure efficient coordination between stakeholders when a breach is detected.

Both practices hinge largely on collaboration. Without technology that allows multiple groups (developers, IT engineers, and security analysts) to work together, it's impossible to bake security into all aspects of the software delivery lifecycle and ensure that teams can mitigate security issues rapidly.

Chapter Summary: Collaboration as the Future of IT

Technology that drives collaboration has always been important. But it has assumed unprecedented urgency in the current IT environment. Today, shifts in the way people work, increasing customer expectations, growing complexity within application environments, and pervasive security challenges require multiple stakeholders to coordinate workflows and share information more efficiently than ever before.

ENHANCING COMMUNICATION ACROSS TEAMS THROUGH LOGGING

Now that we know why collaboration is crucial for modern businesses, we can dive into achieving it.

As discussed in the introduction, centralized logging plays a vital role in facilitating collaboration across teams — primarily technical teams. This chapter explains why.

Logging and Team Success: An Overview

Centralized logs that are easily shareable facilitate communication between teams by allowing engineers to investigate and analyze logs for your applications, which can aid in drawing conclusions based on the actual time-series application log data. As a result, engineers can easily trace the time-series logs back to a likely root cause when there is an issue.

Observability and logging bring engineers together to remediate common problems and more significant issues as they occur across the environmental stack.

For example, logs can help you quickly identify problems with your database or high latency issues with your web application. They can also reveal potential issues reflected in your general Kubernetes health metrics, which can help you better understand the overall health landscape of your Kubernetes clusters.

Whatever the issue may be, logs can help you locate, analyze, and fix many of the problems you face with your tech stack, and they can bring engineers from different teams together to troubleshoot and resolve issues.

Individuals Become "Teams" When They Work Together

When an incident occurs, different teams put their heads together to remediate the issue in a group incident discussion often called a "war room." In this situation, everyone stops whatever they're doing to join a call so that they can achieve a common understanding of:

- What happened?
- Why did it happen?
- How do we resolve it?
- How can we ensure that it doesn't happen again?

Understanding the "What" and the "Why"

To fully understand an issue, you need to know why it occurred in the first place. You should ask yourself what happened and find out what led up to the event that caused your entire application to come crashing down. Analyzing your organization's logs can help you see when things were going well and identify what happened before things started to go south. Logs can help you determine the "what" and the "why" when you're triaging an incident in your war room.

By understanding the "what" and the "why" of an issue, you can make an effective plan to remediate it thoroughly and ensure that it doesn't happen again. That's why having complete, structured logging in place is crucial to your company's success. Without it, you'd be going in circles trying to figure out the "what" and the "why" for everyday problems, not to mention the much more significant issues that could cost you your business.

Easy Wins – Everyday Problems that Logging Can Resolve

Let's say that you're an engineer going about your typical work week. You're working on some sprint tickets and going about your day. Then, you notice some anomalies in your application. You might ask yourself some questions, such as:

- Why is the application so slow today? Could it be running low on compute resources?
- I got a 500 HTTP status code error in the development environment today. I wonder what happened to trigger it?
- This database is not processing my query as fast as it usually does. I wonder if my query is slowing down because other users are running queries against the database?

You can answer all of these questions by simply setting up some basic logging using LogDNA. With centralized logging, you can easily share and diagnose application problems. You can share the log to ensure everyone has the same context when discussing solutions. With LogDNA, you won't have to ask yourself why issues happen anymore, and you'll be able to analyze and triage everyday problems as well as more significant incidents. A setup through LogDNA will ensure that your teams spend less time in the war room and enjoy more relaxed collaboration when remediating issues and building features for your application. As the cliche goes, "An ounce of prevention is better than a pound of a cure."

Let Logs Tell the Story

Setting up your logging in a way that is structured and easy to analyze is also critical. If your logs are hard to understand and difficult to query, no one will use them. Structured logs typically come in the JSON format, which is a standard across all operating systems. When your logs are in the JSON format, you can readily identify the following:

- The time and date for the logged event
- The logging level triggered (debug, info, warning, error, critical, etc.)
- Information about what happened
- The Event ID (the unique identifier for the logged event).

Chapter Summary

Logs can ensure better communication across all of your teams, whether they're in the war room or trying to answer daily questions about observability. Using structured logging, you can ensure that everybody looking at your logs can analyze them quickly and gain a deeper understanding of your application.

OTHER DEVOPS TECHNOLOGIES THAT DRIVE COLLABORATION

Logs are only one of the tools that DevOps teams can use to streamline collaboration. By drawing on the three dimensions of DevOps culture laid out in The DevOps Handbook, this chapter points to other technologies and techniques that help technical teams achieve even greater levels of collaboration and productivity.

Background

According to The DevOps Handbook (Kim, Humble, Willis, and Debois. IT Revolution Press, 2016), three factors enable developing a DevOps culture: flow, feedback, and continual learning and experimentation. By being selective with tooling, companies can minimize confusion and inefficiency while improving the three most important factors for DevOps.

Flow

Flow is the ability to get changes into production quickly and then improve their quality and reliability. Tools such as Jira and Trello contribute to your organization's flow by making work more visible and managing the size and duration of tasks. In addition, boards, issues, and cards can help you break larger projects into smaller tasks.

This visibility will improve workflows in several ways. For example, the ability to see where a unit of work is in the overall process in an easy-to-read format will allow stakeholders to see any bottlenecks. This visibility also helps employees prioritize their work in the context of their company's overall goals. It's common for teams to use tools like kanban or scrum boards to display their work.

Decomposition, which minimizes tasks by breaking them down into smaller units, is a common way to work

more efficiently. Decomposing clarifies daily goals, enables quicker feedback, and makes work seem more approachable and less overwhelming. It's crucial to have a proper tool that can help your teams (and your company) manage the size of projects and tasks by breaking them down into minor units.

Feedback

Feedback is simply information about the results of your processes and projects. For DevOps, this centers on feedback from teams downstream in the process and feedback from production applications. Collaboration tools like Slack and Mattermost have instant messaging functionality, enabling feedback from downstream and across the organization. Meanwhile, tools like LogDNA provide application feedback via logs that developers can use to create safer, more efficient, and resilient software.

Since one of the central tenets of DevOps is to fix problems as far upstream in the creation process as possible, getting feedback from everyone that your work impacts can make a dramatic difference. It's helpful to contact those who work upstream (and even downstream) from your team. Also, as more and more employees are working remotely with fewer and fewer face-to-face interactions in the office, having a tool that enables easy remote collaboration will help them incorporate fixes at the earliest possible point.

In addition, it's critical to implement effective Application Performance Monitoring (APM). Proper logging will help developers write more efficient code, troubleshoot issues in production, create automation to recover from incidents, and more. However, the more complex the environment, the more critical it is to leverage a tool that centralizes your logs in a single location and gives teams the ability to sort through them. It's almost unavoidable when more teams are using the environments, including developers focusing on new features. The right tool will minimize the amount of time that they spend sorting through logs for insights.

Continual Learning and Experimentation

Finally, companies must ensure that they also focus on continual learning and experimentation. Doing so facilitates learning and individual professional development, which will, in turn, contribute to both team knowledge and company knowledge. Your company should provide controlled test environments to experiment correctly. Specifically, you should create the environment via automation to change only the variable you are testing. Companies typically use some form of source control, automated builds, containerization, infrastructure-as-code, or even a configuration management tool to enable this level of control.

For example, you might begin by writing and committing code to a source control repository like GitHub or GitLab. That would trigger a build in an application such as Jenkins or CircleCI to create an image for a container runtime (containerd or CRI-O, for example). Then, you would run that image on infrastructure created by a tool like Terraform or Pulumi. If necessary, you'd use Ansible or another configuration automation tool like Puppet for configuration.

Chapter Summary

It's essential to keep in mind that DevOps is not a team — it's a culture. With that in mind, it's typical for what we would traditionally call the operations team to be the group that drives DevOps. To create a DevOps culture, your company's processes and tools must align well. By staying focused on the principles of flow, feedback, and continual learning and experimentation, you can carefully select the best tools for enabling DevOps in your company.

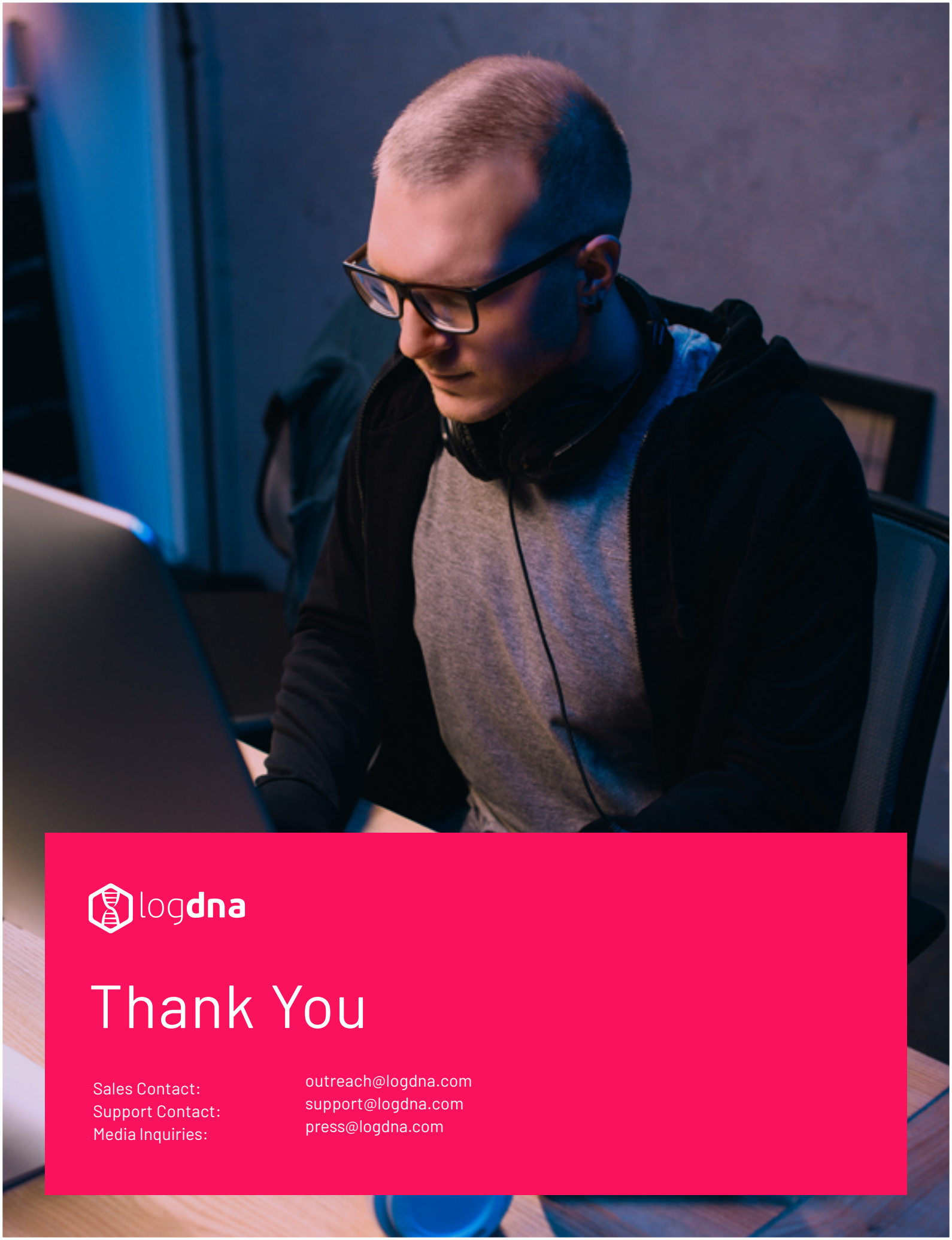
CONCLUSION

Making collaboration a cultural priority is one step toward building a collaborative organization — a feat that businesses must achieve to remain competitive in the present era of distributed workforces, growing customer expectations and IT resources that are constantly increasing in scale and complexity.

But alongside culture, tools are essential for enabling collaborative organizations. In particular, logging, which allows all stakeholders to establish a common source of visibility into what is happening within complex systems, makes it much easier for teams that are otherwise isolated from each other to communicate, define shared goals and collaborate around them. Likewise, different DevOps solutions that help technical teams define procedures, measure outcomes, and experiment continuously round out the tooling that teams need to operate collaboratively.

About LogDNA

LogDNA 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 is mission critical.



logdna

Thank You

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