

E-BOOK

From Monitoring to Observability

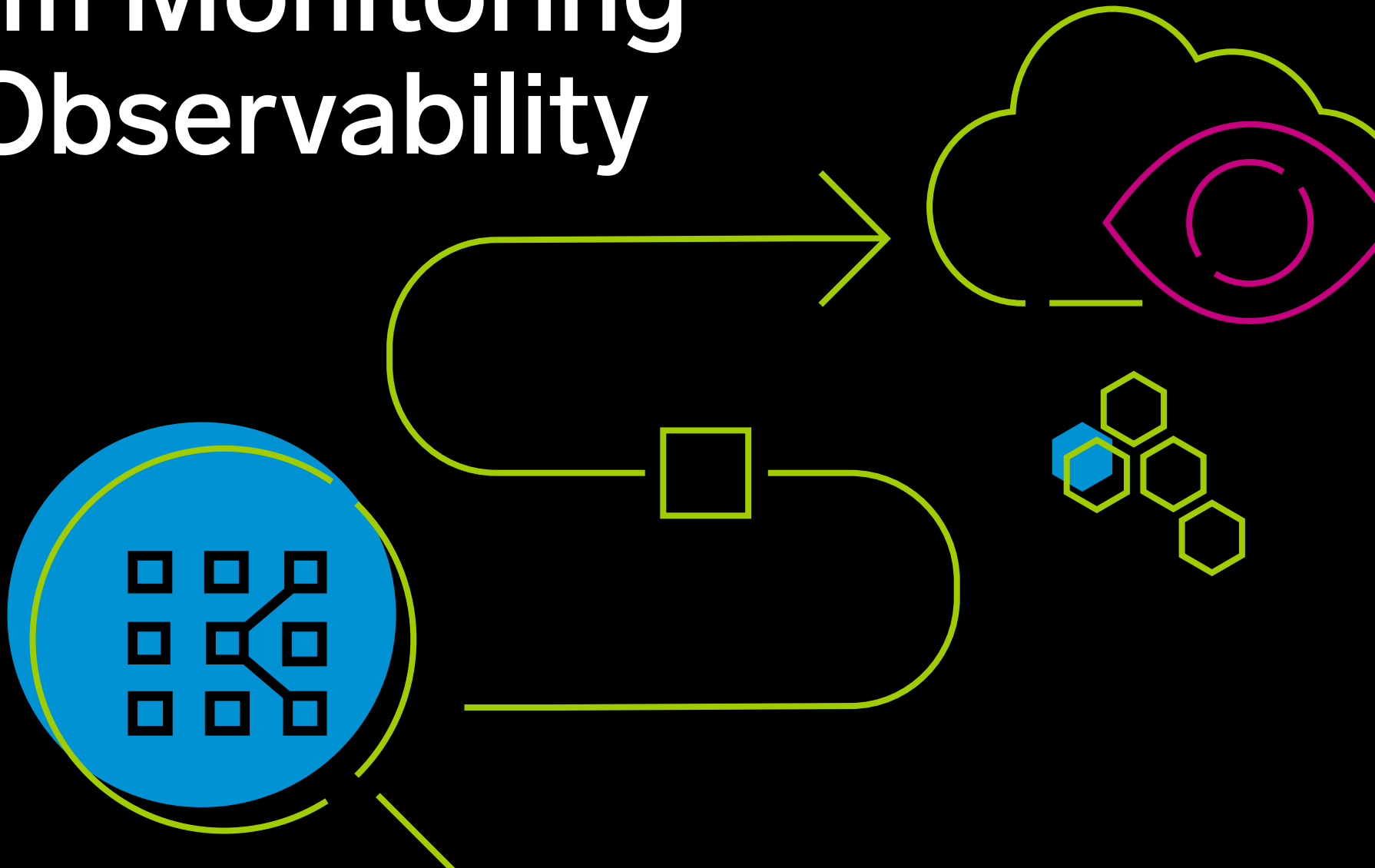
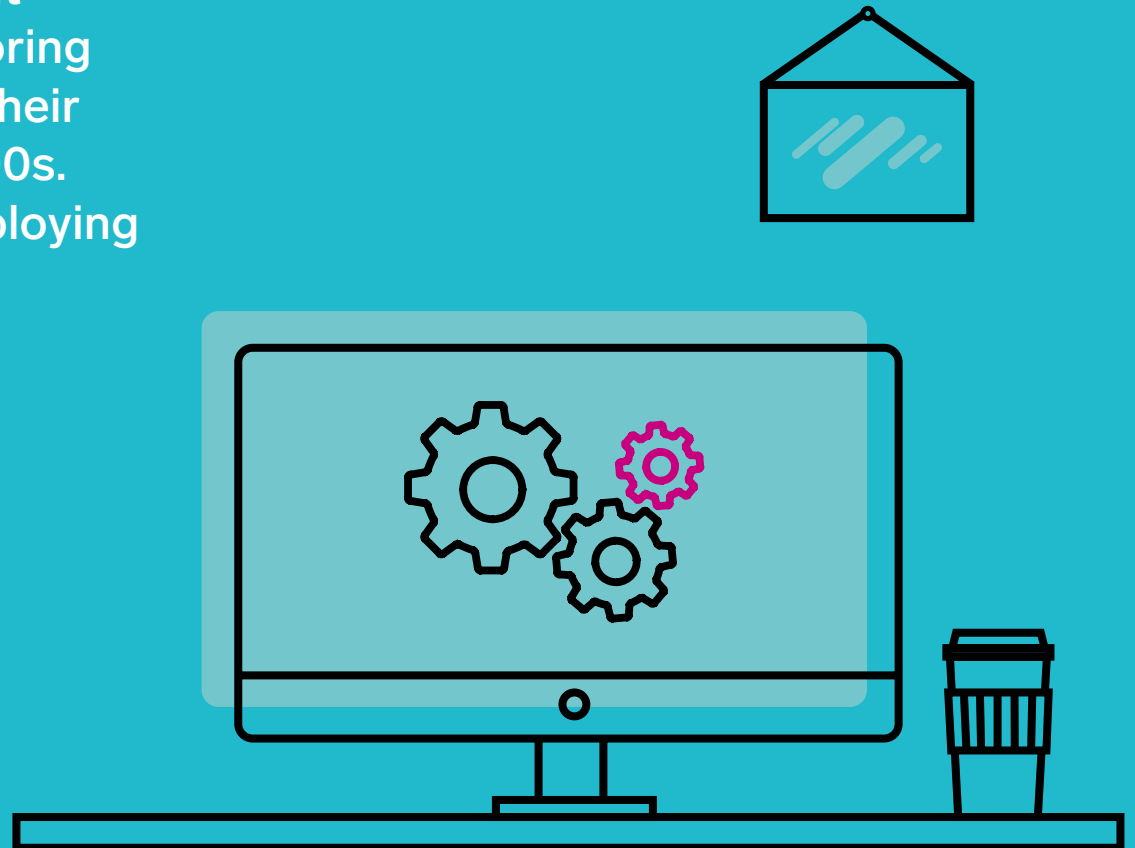


Table of contents

Monitoring and Observability: A brief history	3
Similarities between Observability and Monitoring	6
How is Observability different from Monitoring?	8
Using Monitoring and Observability together	10
Observability: Not just a buzzword	12
DevOps teams need Monitoring and Observability	14

Monitoring and Observability: A brief history

Long before anyone was talking about observability, they were talking about monitoring. Modern software monitoring tools, like Nagios and Zabbix, trace their origins to the late 1990s or early 2000s. That's when IT teams first began deploying tools that systematically monitored application environments for problems and sent alerts when something seemed amiss.

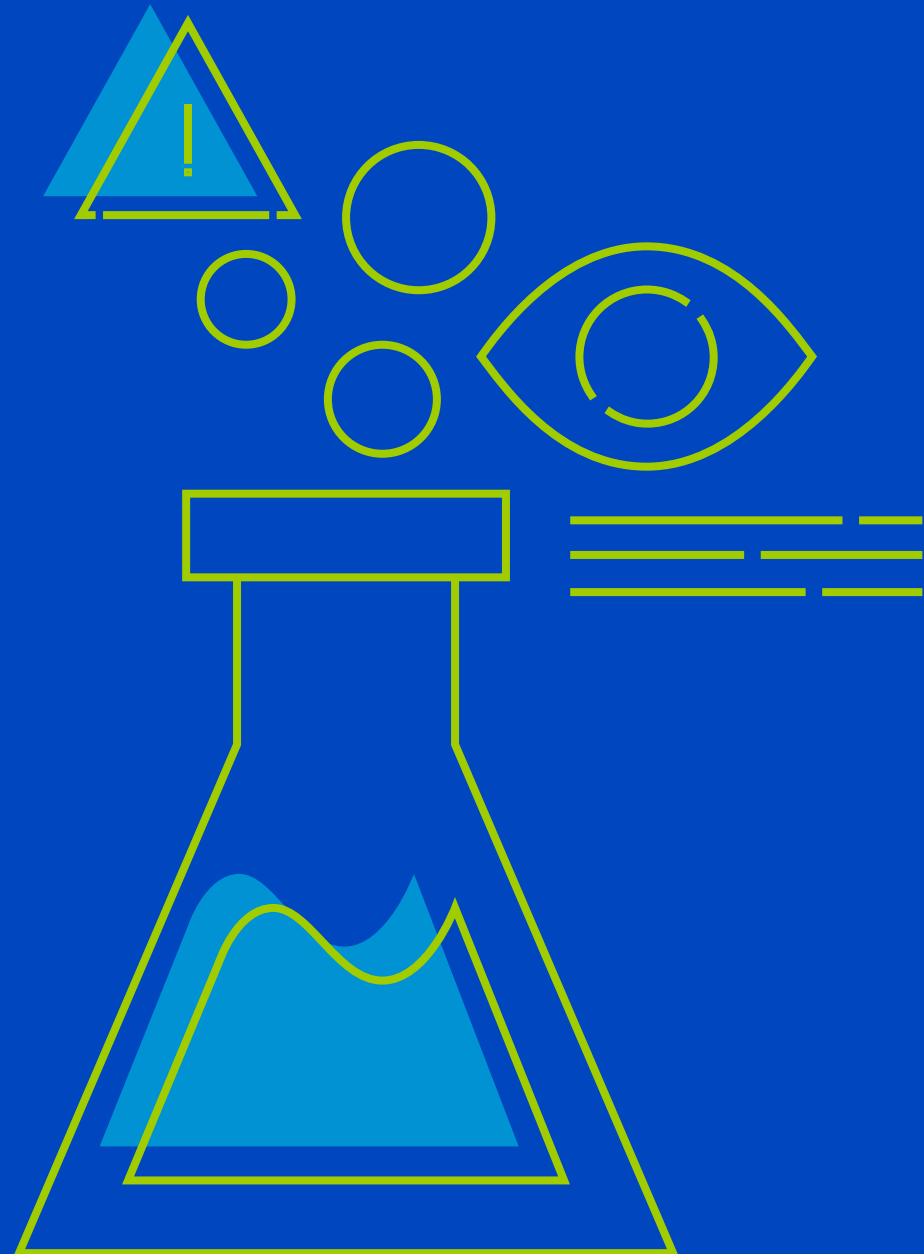


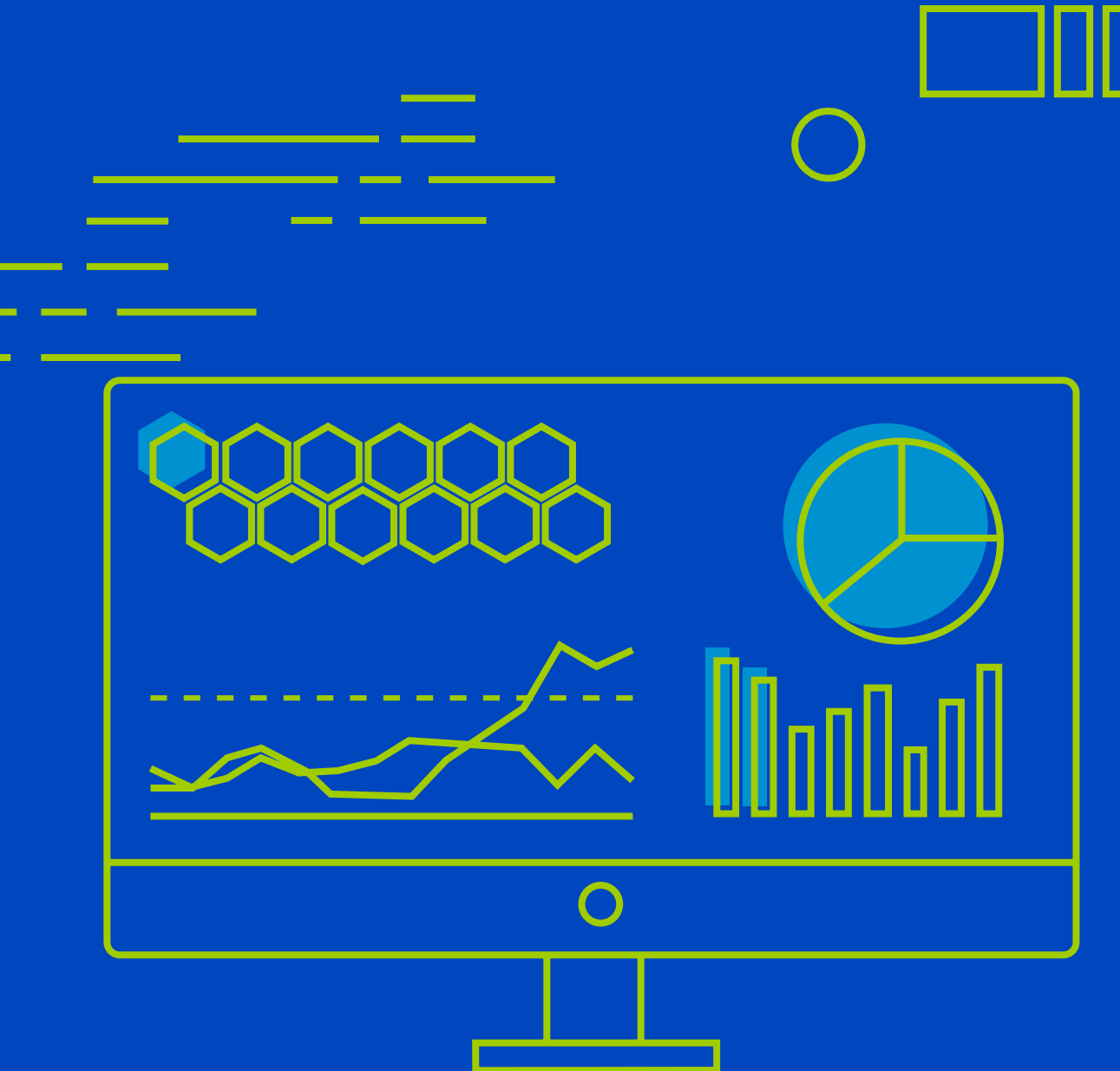
Monitoring tools have grown more sophisticated since then.

Instead of merely pinging servers to see whether they respond, or tracking CPU and memory usage to see if they are maxed out, modern monitoring tools use sophisticated analytics to identify problems within complex software environments. Nonetheless, the core foundation of monitoring remains the same: it's all about finding problems.

Observability is a much newer term, at least within the context of software.

Although you can find occasional references to the observability concept from the earlier 2010s, it wasn't until circa 2016 that observability came into vogue among Site Reliability Engineers (SREs) and DevOps teams.



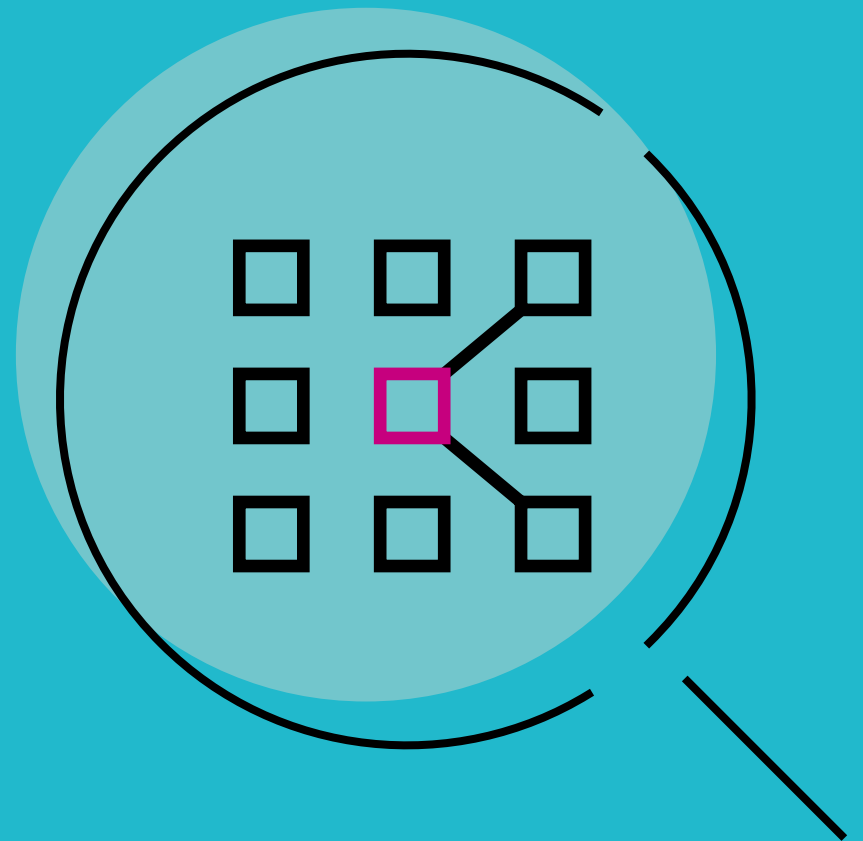


Probably because the term is so new, observability is harder to define in a clean way than monitoring. Different people and companies offer different definitions. Most definitions of observability, however, focus on the idea that observability means collecting actionable data in order to gain a true understanding of problems identified by monitoring tools.

Similarities between Observability and Monitoring

At a high level, observability and monitoring share some key characteristics.

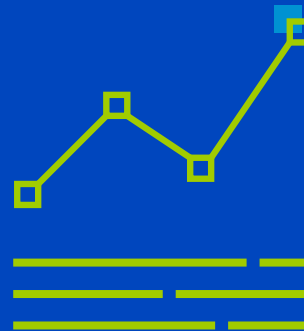
For one, they both help support the overall reliability and performance of software environments. They help find and (in the case of observability, at least) fix problems.



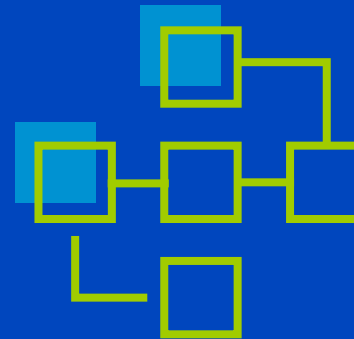
They are also similar in that they both leverage multiple data sources. Although monitoring tools may not perform the advanced data correlation and analytics of observability tools, and monitoring tools usually rely primarily on metrics, they can leverage other types of data points—including logs and traces—to collect data from a software environment. Observability solutions, too, leverage a variety of data sources to gain the depth of visibility necessary to understand complex problems.



..... Logs



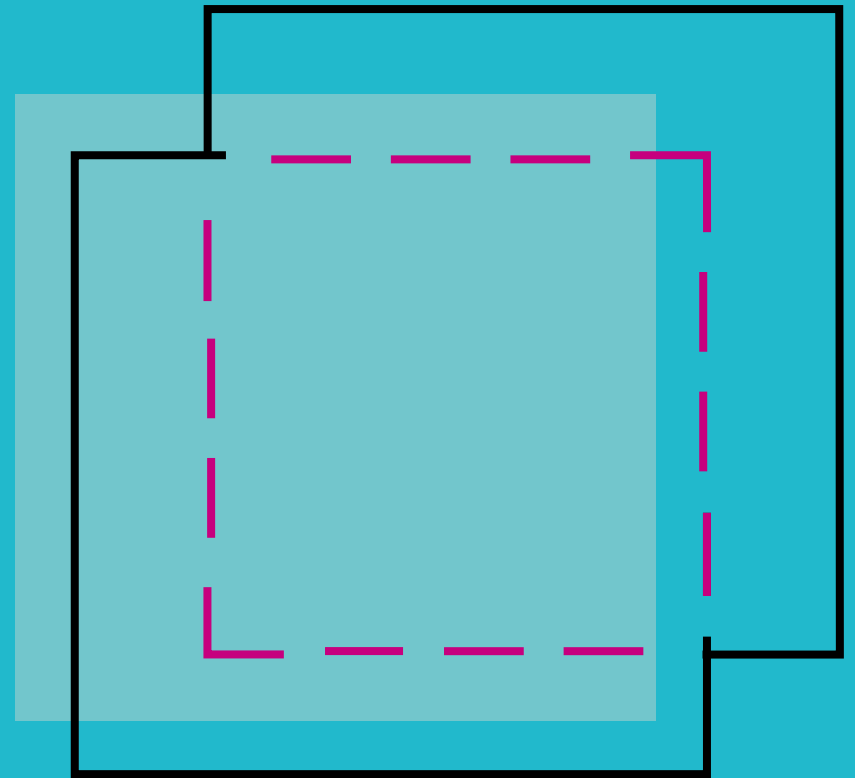
..... Metrics



..... Traces

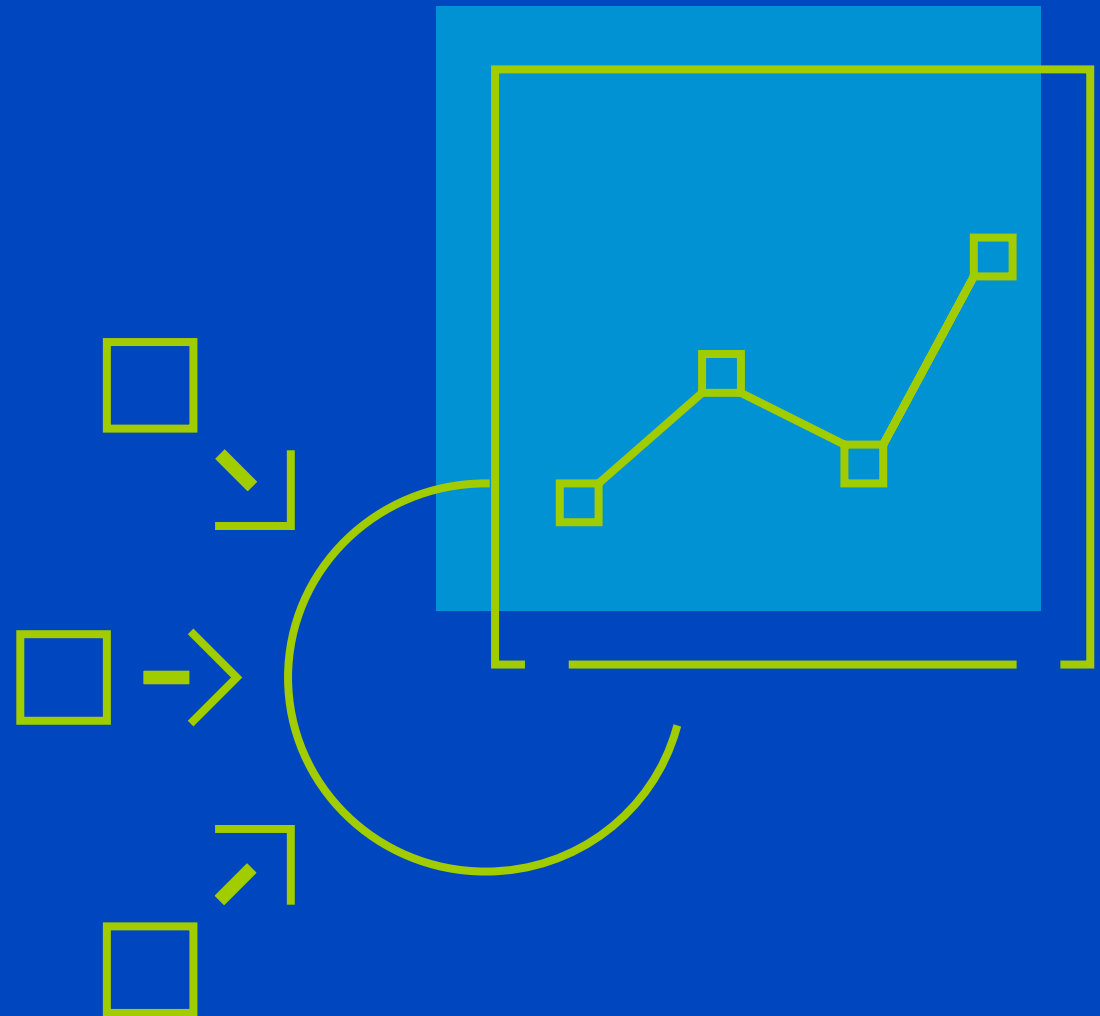
How is Observability different from Monitoring?

The difference between monitoring and observability boils down to the end goal. Whereas monitoring focuses on finding problems, observability focuses on understanding and resolving them.



To put this into context, generating an alert when a node fails in your Kubernetes cluster would be an example of monitoring. That's simple enough to do: you'd run an agent on the node itself (or possibly within Kubernetes) that would monitor whether the node is responding, and trigger an alert if it's not.

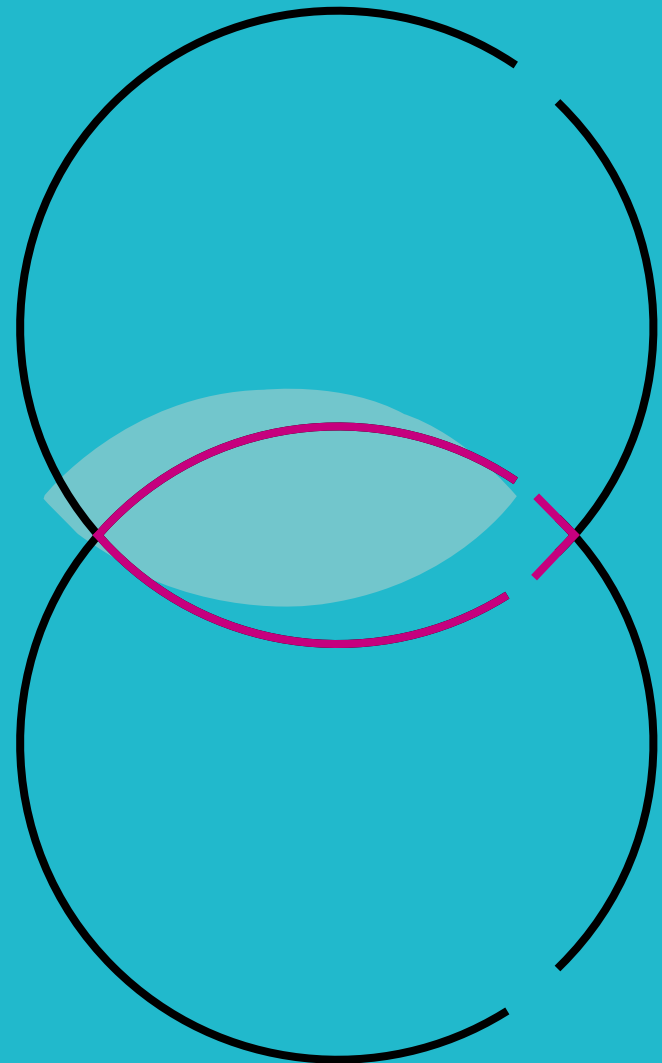
In contrast, observability of the node failure would entail collecting data from multiple sources—the node, the hypervisor hosting it (if it's a virtual machine), the various Kubernetes services that were interacting with the node, and the pods hosted on the node—and then correlating those data sources to determine why the node failed. Maybe it was hosting a pod that suffered a major crash and took the node with it. Maybe it had a kernel panic at the OS level. Maybe it was shut down by an IaaS provider because you didn't pay your bill. Whatever the root cause of the issue, monitoring alone wouldn't expose it, but observability would.

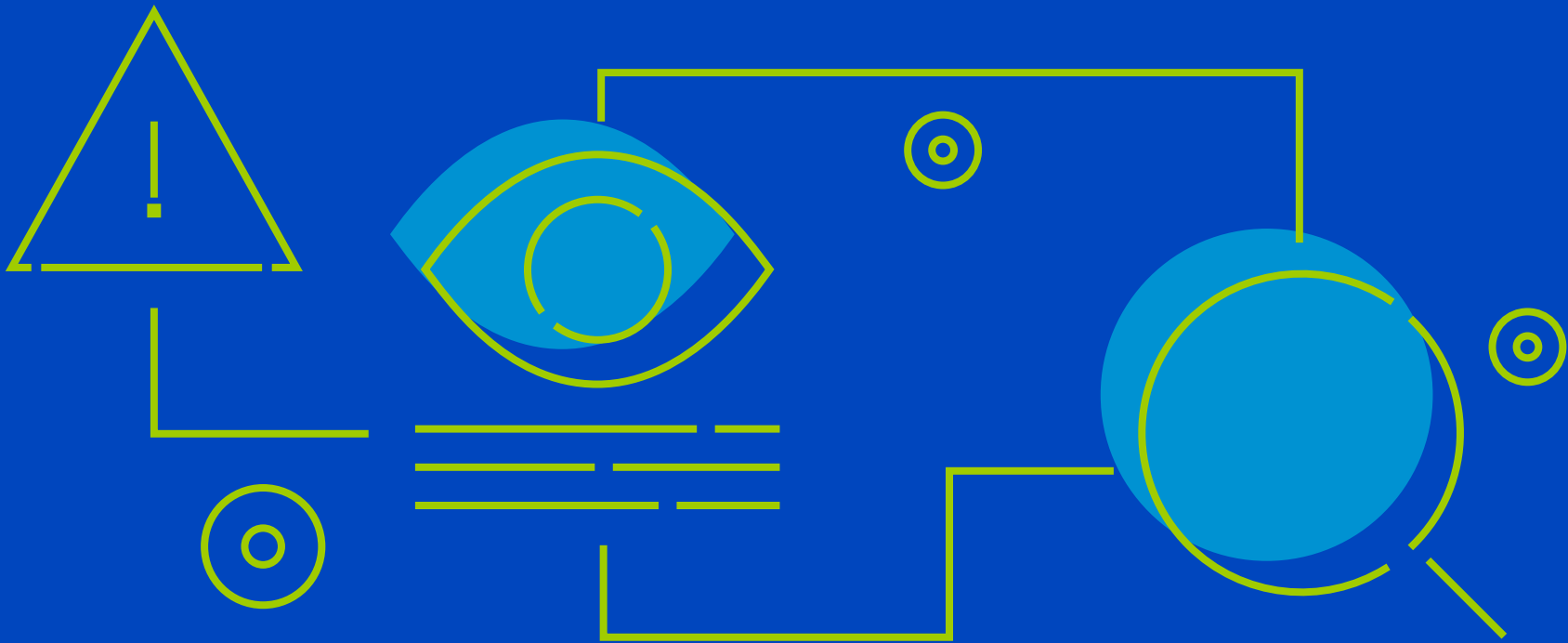


Using Monitoring and Observability together

In general, DevOps teams seem to agree that observability and monitoring are distinct types of operations that address different problems

Still, monitoring and observability are inseparably linked. Monitoring tools can tell you when something goes wrong, and observability tools can help you investigate the issue after you detect a problem.





Pairing monitoring and observability is beneficial because not all problems identified by monitoring tools require sophisticated investigation. Maybe your monitoring tools send an alert telling you that a server went offline, but it was part of a planned shutdown, for instance. In that case, you need not collect and interpret multiple types of data to understand what happened. You

can just log the alert and move on. But when serious problems arise and you need to troubleshoot them quickly, observability data is crucial. Although you could technically collect the same type of data manually that observability tools deliver automatically, the data collection would take time and delay incident resolution. Observability tools ensure that you always have the data you need

on hand to interpret a complex problem. Many solutions also offer recommendations or automated analyses that can help teams sift through complex observability information and identify root-cause problems more efficiently.

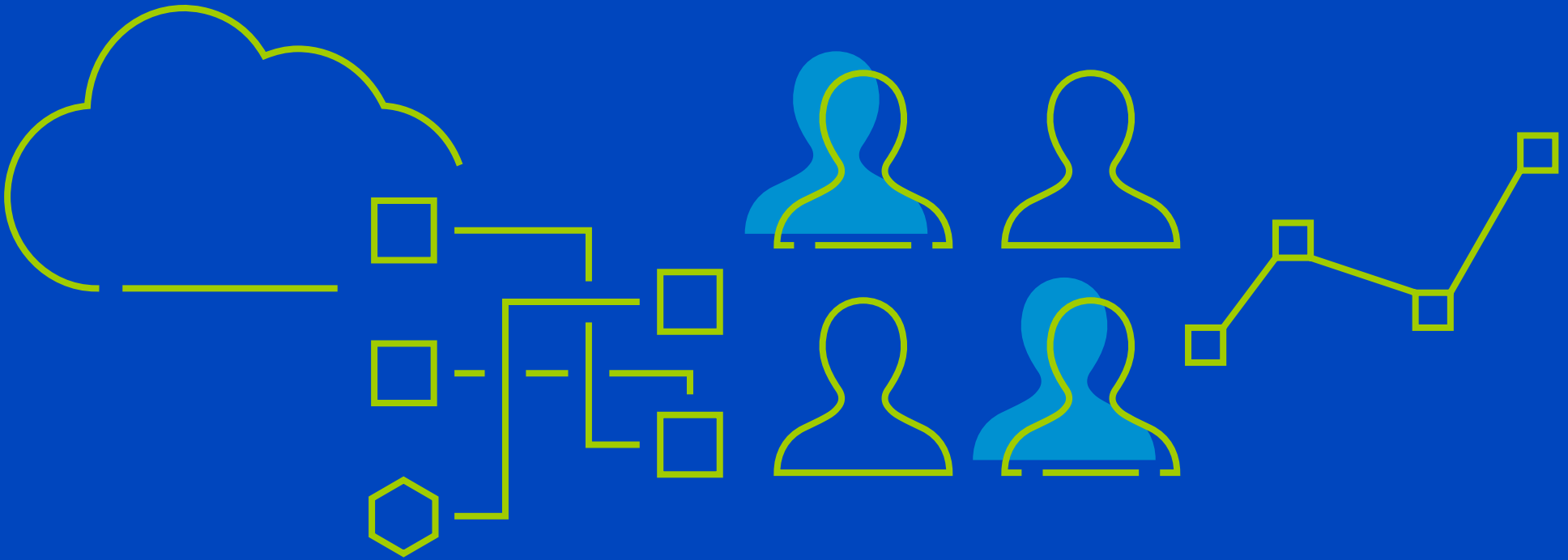
Note, too, that a system must be observable – meaning that it must generate data that can be collected and interpreted – in order

to be monitored. A closed system that offers no facilities for data collection (such as a proprietary IoT device that generates no metrics) can't be monitored for failure because it lacks observability. And, of course, it can't be observed, either, because there is no data to use as the foundation for observability.

Observability: Not just a buzzword

You could argue that observability is a buzzword in the sense that teams were performing observability before the term existed. The idea of analyzing diverse data sources in order to understand complex application performance and reliability issues is hardly new.





On the other hand, the buzz surrounding observability does seem to reflect some important changes that have taken place in the DevOps ecosystem in recent years. For one, the emergence of endlessly complex cloud-native environments has made the ability to collect and correlate complex data sets absolutely critical.

When all of your applications ran as monoliths on virtual machines, simply monitoring them and then investigating issues manually

may have been enough to keep everything running. In the cloud-native world of microservices, containers, serverless functions, and scale-out infrastructure, however, you need deeper visibility than you can achieve through monitoring alone.

The observability trend perhaps reflects increased awareness that DevOps practices should be tied to business goals. Monitoring helps technical teams identify technical problems, like a server that is

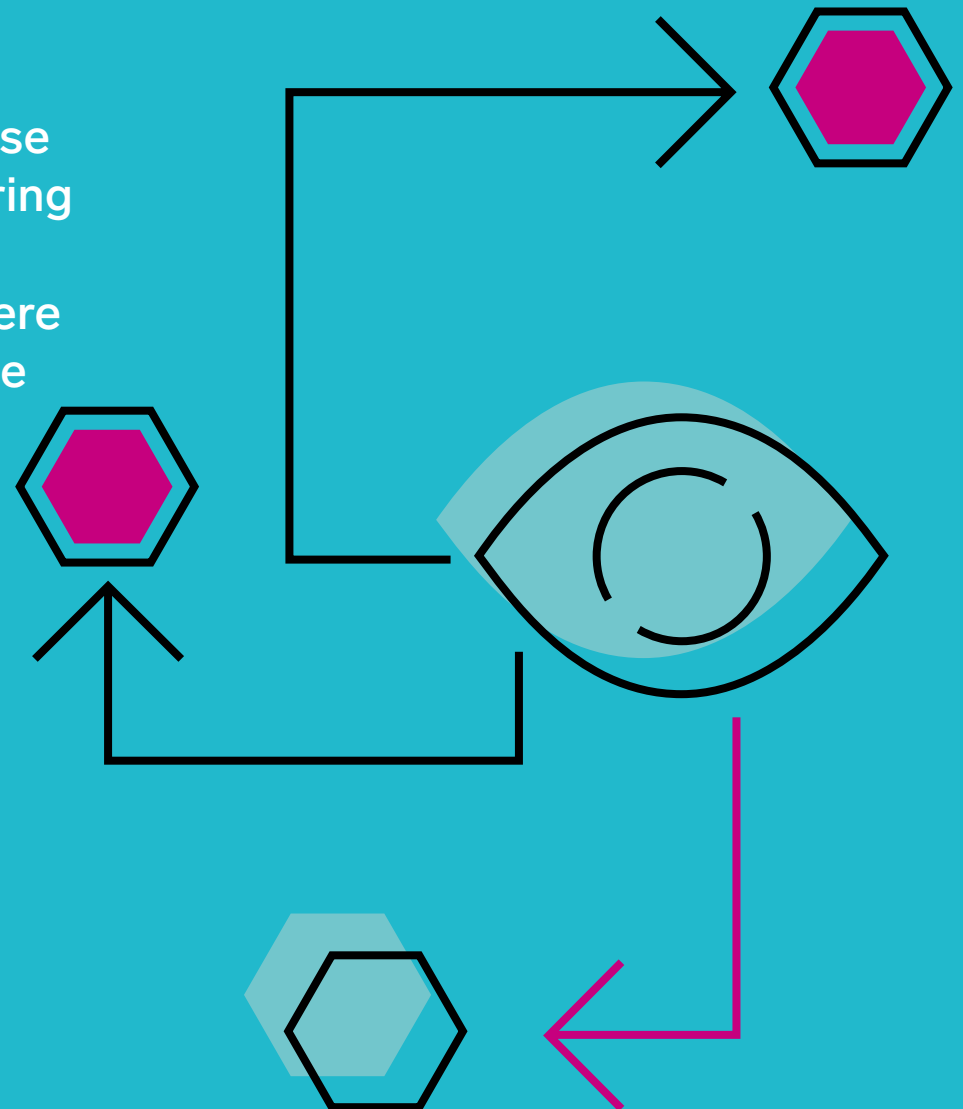
down or an application that is not responding. Observability provides finer-tuned insights into application behavior and performance. It can help teams hone in on a microservice that is causing problems for a particular subset of users, for example, or a resource allocation that is insufficient for handling application demand at certain times of day.

In these ways, observability helps align technical issues with business issues. It makes it clear which

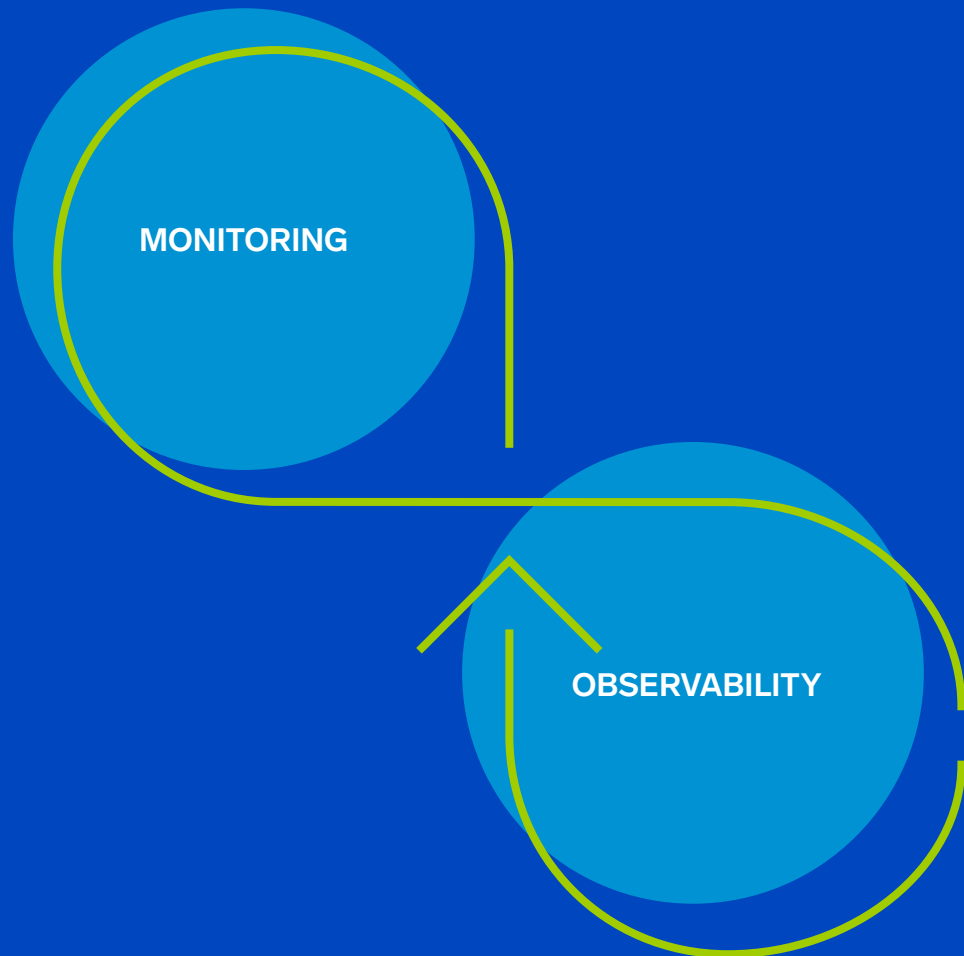
business priorities (like serving a certain user group or handling a business event that leads to a spike in traffic) are impacted by technical shortcomings. In contrast, monitoring provides only high-level indications that something is wrong, without offering much opportunity for measuring the specific business impact of the problem.

DevOps teams need Monitoring and Observability

You could go on and on in debating the precise similarities and differences between monitoring and observability, or arguing about whether observability is actually a useful term or a mere buzzword. But that's not going to improve the outcome of your DevOps processes.



Instead, perhaps the best way to think about monitoring and observability is to approach them both as must-haves for modern DevOps teams. Yes, there may be some nuance surrounding how you implement them or exactly what each term means. But viewed from the perspective of DevOps operations as a whole, you need both monitoring and observability to find and fix application issues and, in turn, ensure an optimal end-user experience.



About Sumo Logic

Sumo Logic Inc., (NSDQ: SUMO) is the pioneer in continuous intelligence, a new category of software, which enables organizations of all sizes to address the data challenges and opportunities presented by digital transformation, modern applications, and cloud computing. The Sumo Logic Continuous Intelligence Platform™ automates the collection, ingestion, and analysis of application, infrastructure, security, and IoT data to derive actionable insights within seconds. More than 2,100 customers around the world rely on Sumo Logic to build, run, and secure their modern applications and cloud infrastructures. Only Sumo Logic delivers its platform as a true, multi-tenant SaaS architecture, across multiple use-cases, enabling businesses to thrive in the Intelligence Economy.



To learn how Sumo Logic's Application Observability solution
can accelerate your DevOps processes, visit:

<https://www.sumologic.com/solutions/application-monitoring/>



sumo logic

Toll-Free: 1.855.LOG.SUMO | Int'l: 1.650.810.8700

305 Main Street, Redwood City, CA 94603

© Copyright 2021 Sumo Logic, Inc. Sumo Logic is a trademark or registered trademark of Sumo Logic in the United States and in foreign countries. All other company and product names may be trademarks or registered trademarks of their respective owners.

Updated 10/2021