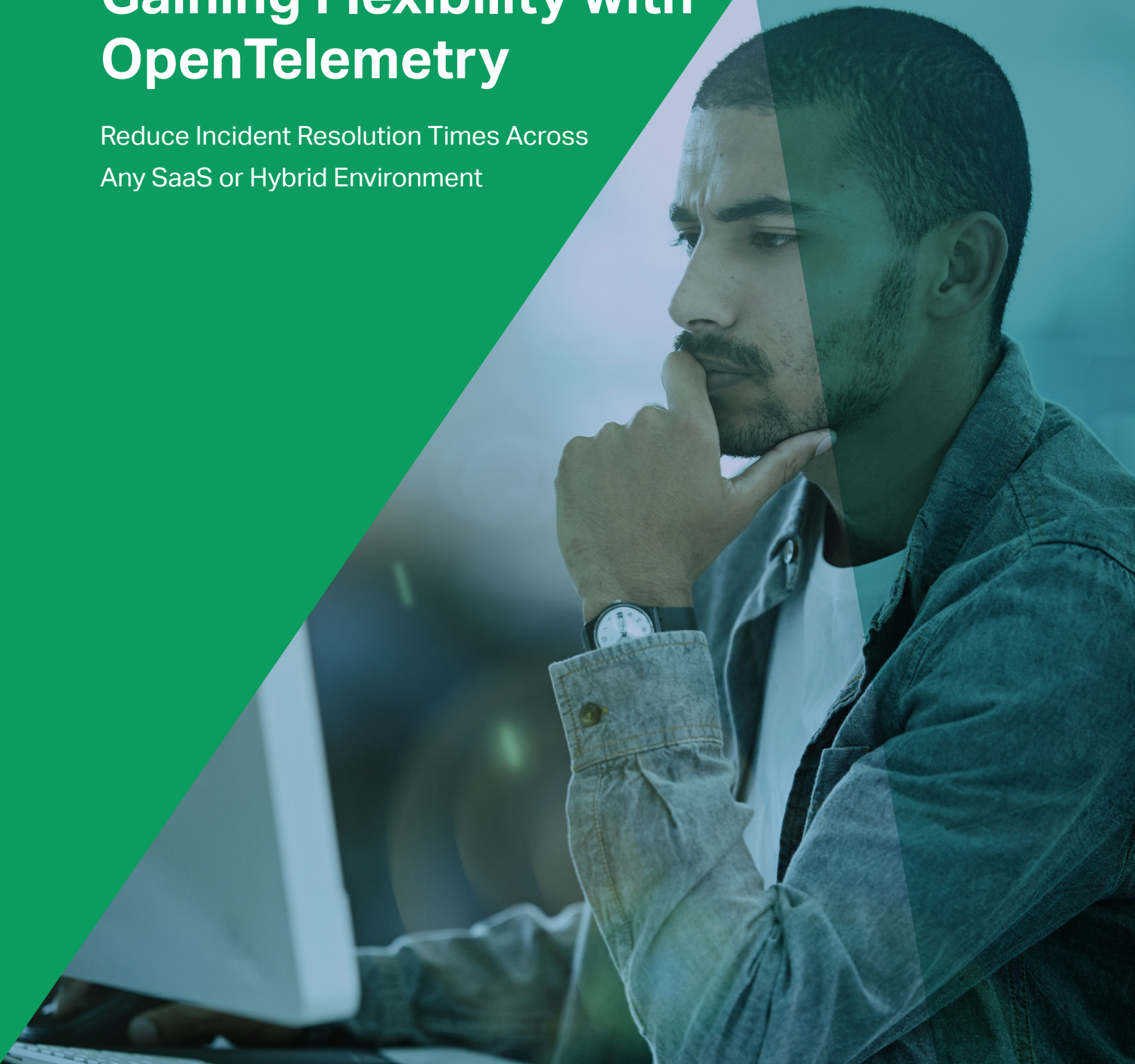




EBOOK

Gaining Flexibility with OpenTelemetry

Reduce Incident Resolution Times Across
Any SaaS or Hybrid Environment





Why You Need OpenTelemetry

The task of deploying SaaS and applications efficiently, and at scale, has led to many innovations — and many complexities. Anyone who has looked at the rat's nest of crisscrossing wires in a home electronics system can sympathize with the landscape that is today's distributed systems.

But on the other side of these complex integrated systems — from cloud technologies to virtual containers to Kubernetes — is the end user who just wants the job done. Every second counts when systems are down or not functioning as they should, yet tracking down the source of stalled environments can sometimes take hours, if not days.

When a user cannot easily access your SaaS engine, or when a payment processor or other API stalls, do you know where to look to fix the problem? This has only become more challenging as the world continues to lean on distributed systems in today's work-from-anywhere environments, highlighting a need for a solution that can mitigate these issues.

Your DevOps team needs every tool at their disposal to decrease incident resolution times. This is why you need OpenTelemetry.



The Case for Observability

Traditionally, developers have responded to the need for troubleshooting by instrumenting code to spit out telemetry data for analysis. This generates logs, metrics, and traces that act as breadcrumbs on the trail.

In theory, this approach ensures DevOps can keep a finger on the pulse of distributed environments, allowing for clearer communication between developers and IT. The assumption is that when the software breaks down on the user end, DevOps can work with telemetry data to diagnose and troubleshoot the problem.

In practice, however, it's not as straightforward.

From the complexities of the distributed systems landscape to the fact that different components of a microservice architecture might talk in different languages, it's inevitable that crucial information might remain siloed and inaccessible. This is further compounded by the rate of telemetry data; developers and IT teams are often overloaded with data that makes it nearly impossible to identify which information to prioritize.

Taking this one step further, if a new API needs to be integrated into your ecosystem, DevOps has to figure out new integration pathways.

Introducing: Observability Teams

Teams that build and manage complex modern distributed applications built on microservices or serverless architectures and languages, like node.js, php, and go, for example, frequently use open source instrumentation. In fact, according to Gartner, 70% of new cloud-native application monitoring will use open-source instrumentation, rather than vendor-specific agents, for improved interoperability by 2025.

To further illustrate this transition, a new team is emerging within cloud-native service companies that leverages both open source and off the shelf monitoring tools: Observability Teams.

Observability Teams are responsible for defining monitoring standards, delivering monitoring data, measuring the reliability of monitoring solutions, and maintaining tooling and storage of monitoring data. Basically, the Observability Team ensures the digital experience is as seamless as possible.

This illustrates that DevOps and Site Reliability Engineers (SREs) continue to bear the brunt of the challenges driven by distributed systems. In an effort to manage these challenges, these teams have added backend solutions to translate telemetry data through a single, commonly understood vocabulary. But this isn't a long term solution.

The pressure is endless, and relentless. OpenTelemetry offers a path to solve these challenges.



Why OpenTelemetry is DevOps' Friend

OpenTelemetry is a standardized vendor-agnostic way of collecting telemetry data and works with whatever distributed systems are in your ecosystem. Equally important, it does not just work with your systems today; it will continue to work with them tomorrow.

Why OpenTelemetry is Different

When it comes to observability, there are various ways to collect telemetry data across your distributed environment. These include:

- **Distributed Tracing**
Distributed tracing, also known as distributed request tracing, is a method used by developers and IT teams to profile and monitor applications in a distributed environment. These environments usually consist of software built using microservices or serverless architecture. Distributed tracing helps pinpoint where failures occur and what causes poor performance. This is often the best way for DevOps, operations, software, and site reliability engineers (SREs) to quickly get answers.
- **OpenTracing**
Originally grown out of Jaeger and Zipkin, OpenTracing offers a vendor-neutral API for adding tracing instrumentation to applications and delivering that data into distributed tracing systems. It's worth noting, however, that although OpenTracing has been popular with developers, it requires scalable storage and search capabilities, making it insufficient for scaling to enterprise class deployments
- **OpenTelemetry**
OpenTelemetry is an open source observability framework created by the Cloud Native Computing Foundation (CNCF) that brings together a collection of specifications, tools, APIs, and SDKs to provide a more efficient way to troubleshoot, debug, and manage applications.



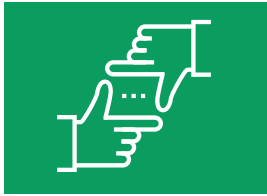
This is made possible by bringing together OpenTracing and OpenCensus (a set of libraries for various languages that enable application metrics and distributed traces to be collected and transferred in real-time). The merger between OpenTracing and OpenCensus created a new, unified standard (aka OpenTelemetry) that allows for a complete telemetry system, one that enables more effective monitoring and observability capabilities across today's distributed systems.

Worth noting is that perhaps the greatest differentiator of OpenTelemetry is that it is completely vendor-agnostic. This means that DevOps teams can easily embed observability with Jaeger, Zipkin, or any other standards-based approach, another perk to the convergence of OpenTracing and OpenCensus.

Setting up OpenTelemetry to do its job well requires a few components for telemetry extraction, gathering, relaying, and analysis. These include APIs that can collect telemetry data from different language frameworks, exporters that can route data to the right centralized platform, and software development kits (SDKs) that can connect the APIs to the exporters.

Observability is more streamlined with OpenTelemetry, enabling DevOps teams to spend less time figuring out how to collect data and instead focus on interpreting data for insights into systems behavior. The key takeaway: With OpenTelemetry, DevOps gains more flexibility, and team members can trace problem nodes faster and reduce incident response times across any SaaS or hybrid environment.

While this framework might sound complicated, there are solutions that can help offset the heavy lifting for your team. Remember that OpenTelemetry is a lifestyle — once installed, the process flow can work forever and keep pace with an evolving cloud-native architecture. If IT requires fully featured management capabilities to support enterprise-class deployments, it might be time for a single solution for all your telemetry analytics needs.

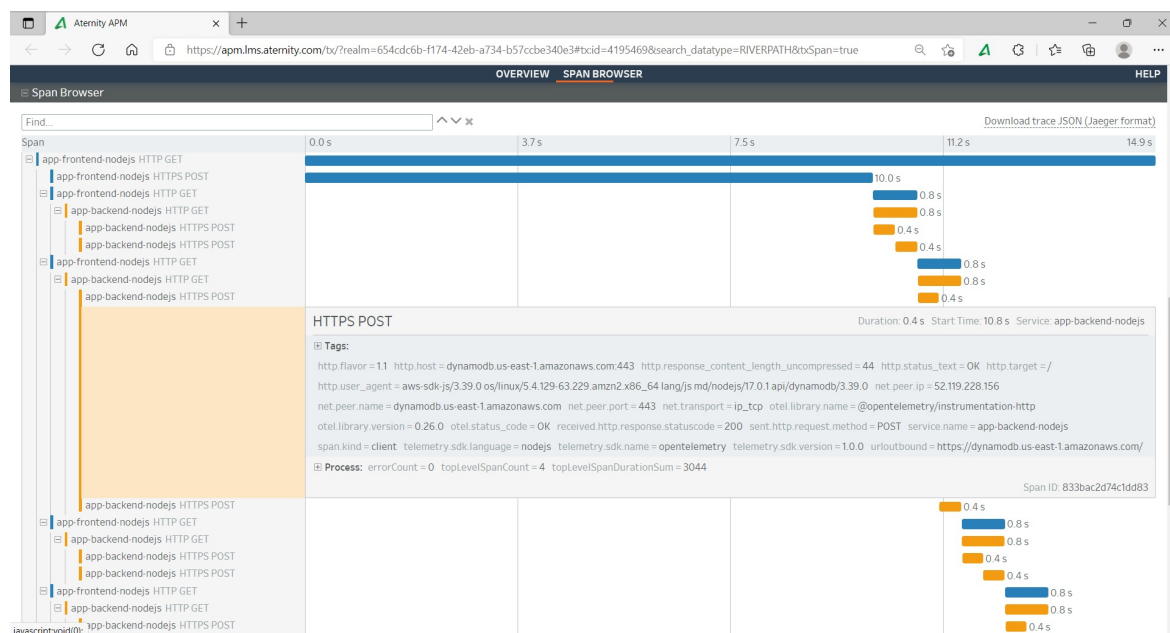


Expanding Observability with Aternity OpenTelemetry Analytics

All too often, DevOps and SREs have to patch together ad hoc solutions for gathering and analyzing telemetry data across complex distributed, cloud-native microservices. And though OpenTelemetry solves the challenges of disparate data across various systems, it doesn't always provide the right level of insight. Basically, it gives the story from the machine point of view without offering insight into the actual end user digital experience.

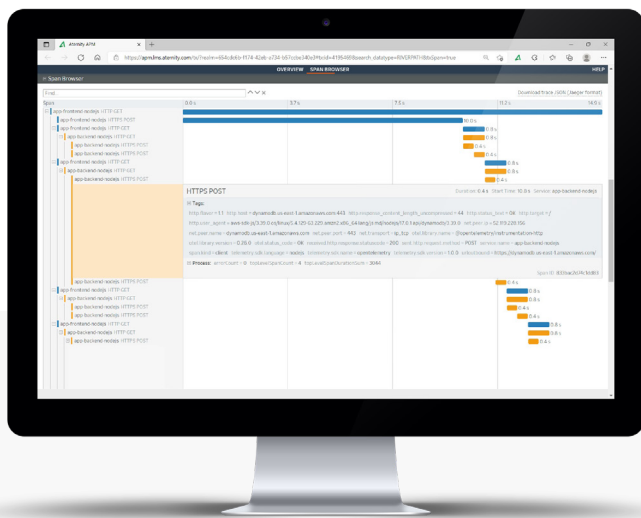
[Aternity OpenTelemetry Analytics](#) collects all the telemetry data across your multi-cloud and distributed systems environment enabling DevOps and IT teams to trace every user transaction to more effectively and efficiently optimize performance for multi-language apps.

The Aternity OpenTelemetry Analytics service captures and stores every transaction in the context of a user's business activity. Unlike other solutions that only sample telemetry data at predetermined intervals, Aternity does not sample. This observability at scale empowers teams to zero in on the source of problems faster and decrease incident resolution times.



Additionally, Aternity OpenTelemetry Analytics provides end-to-end distributed tracing and analytics across complex cloud and microservices environments in the context of user business activity. It is not just about giving you a holistic view of your distributed network; it's about connecting the dots between the network view and the customer view. It helps build context so your IT or DevOps teams can trace the user journey alongside the cloud architecture status and map the two more accurately. As a result, incident resolution can get easier for all future incidences of similar user behavior.

DevOps teams, in turn, can rest assured that their distributed cloud-native environments can be deployed and managed at scale. It's a win-win.



Contact us to learn more about how Aternity OpenTelemetry Analytics can help your organization deliver a seamless digital experience across your distributed environment.