

I attended the the webinar “AWS and Sumo Logic: Observability With OpenTelemetry”

 dev.classmethod.jp/articles/i-attended-the-the-webinar-aws-and-sumo-logic-observability-with-opentelemetry

HemanthKumar R

October 5, 2023

Today's Speakers



sumo logic

Melissa Sussmann
Lead Technical Advocate



aws

Michael Hausenblas
Product Owner

目次

Introduction

Hemanth from the Alliance Department here. In this blog i wanted to give an summary of a recent webinar that caught my attention: "AWS and Sumo Logic: Observability With OpenTelemetry"

Speakers for the Session

Today's Speakers



sumo logic

Melissa Sussmann
Lead Technical Advocate



aws

Michael Hausenblas
Product Owner

Overview of Sumo Logic & AWS

Sumo logic a leading log Analytics Platform catering for both security and Observability use cases. Serving AWS customer was a tremendous success. Its success story is entwined with the enhancement of application security and dependability by various businesses. Let's examine one of these success stories:

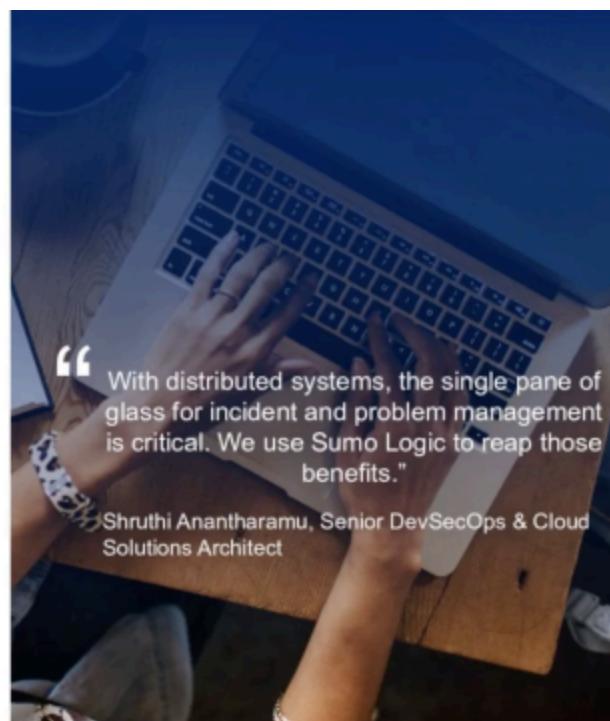


TOKIO MARINE
HCC

Sumo Logic supports Tokio Marine's adoption of EKS

- Gained visibility into their full AWS stack as they moved from VMs to EKS
- Reclaimed time that would have been lost digging through logs for pertinent information when triaging alerts with centralized log data from EKS containers and APIs, custom dashboards and incisive queries
- Reduced cross-region latency from seven seconds to sub-second

sumo logic



Importance of OpenTelemetry (Otel)

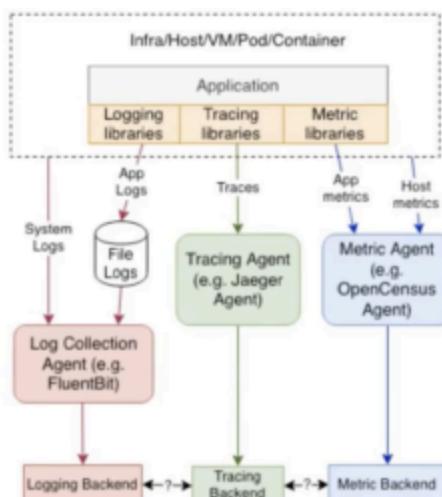
OpenTelemetry, more often known as Otel, has become a key player in the observability field. It was created in 2019 as a result of the union of Google's opentracing and opencensus projects. But why is Otel important? Let's quickly review telemetry pipeline data collecting to better comprehend this. Otel is more than simply a data filter; it also creates richer metadata. It supports multiple backends, including information about Kubernetes containers and other crucial logging and trace data. Otel, often known as opentelemetry, is essentially an open standard for data gathering. It is a vendor-independent diagnostic tool that can gather application telemetry data such as traces, metrics, logs, and other information and send it to the backend of your choice.

Observability 1.0 vs Observability 2.0

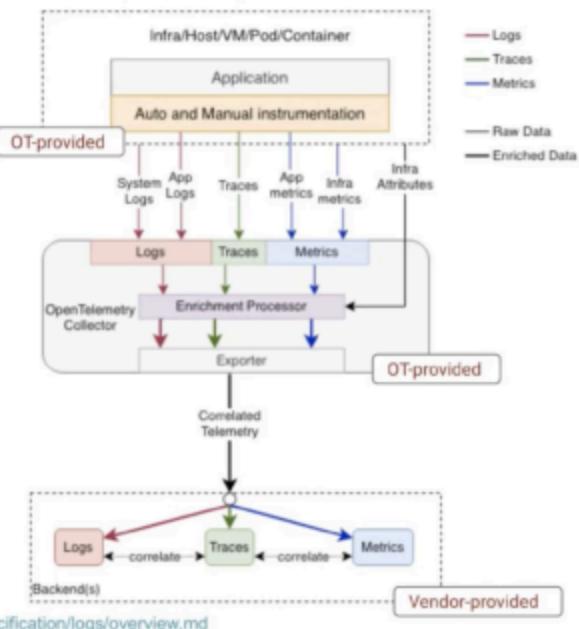
A big change occurred with the transition from Observability 1.0 to Observability 2.0. The latter uses a uniform collecting mechanism and does away with the necessity for three different backends. Otel does away with the need for brand-new libraries and signals. As a result, building libraries takes less time, and managing various signals requires less effort. Everything is more organized when there is just one agent/collector. Otel additionally urges providers to adopt the OpenTelemetry Protocol (OTLP) for a more uniform method.

Observability 1.0 vs 2.0 (aka "the promise")

Separate Collection



OpenTelemetry Collection



<https://github.com/open-telemetry/opentelemetry-specification/blob/master/specification/logs/overview.md>

Embracing the OpenTelemetry Standard: Why Now?

Why use the OpenTelemetry standard? Why now?



Foundation of Elite Performing Teams in DevSecOps

- Highly converged telemetry, eliminate data silos across IT
- Simplifies analysis across increasingly deep-system architectures

Observability Driven Development - Observable Systems Design

- Build a culture of continuous improvement in Observability
- Build a culture of production feature prototyping & testing

Monthly Commits to OpenTelemetry

- OTel is the second largest CNCF project (behind Kubernetes)
- Over 23K code commits / month and growing
- [Up from 7K commits/month in 2020](#)

OpenTelemetry Signals

Logs: Primarily meant for human consumption, provide information about what is occurring and our current position. Metrics: Numerical signals that combine data from several different services to represent indicators. Traces: These signals utilize the context dispersal principle to function across a range of services. The waterfall model is commonly used to depict them and shows the services along the request flow. Profiles: A relatively recent subset of telemetry signals designed for certain applications.

Telemetry signals



Sumo Logic Contribution to Otel and how you can Contribute to Otel

Upstream Contributions

Sumo Logic OTEL Contributions

Some auto-instrumentation SDKs Sumo Logic supports and has contributed to include: [Go](#), [Istio](#), [Kubernetes Tracing](#), [JavaScript](#), [Java](#), [.NET](#), [Python](#), [Ruby on Rails](#)

Notably, we also recently added PHP auto-instrumentation upstream and explain in detail with code examples and installation steps how to try it out in this [opentelemetry.io blog post](#) by Przemek Delewski.



How to Contribute to OTEL

- There are frequently issues labeled with "help wanted" or "good first issue"
- E.g. here's the [list for opentelemetry-java](#):
- Don't be shy. Leave a note in the issue, asking for assignment. Keep your PR's small if possible
- Each repo also has a [CONTRIBUTING.md](#) doc, which describes the specific details. Please read it before contributing :)
- Contributor License Agreement

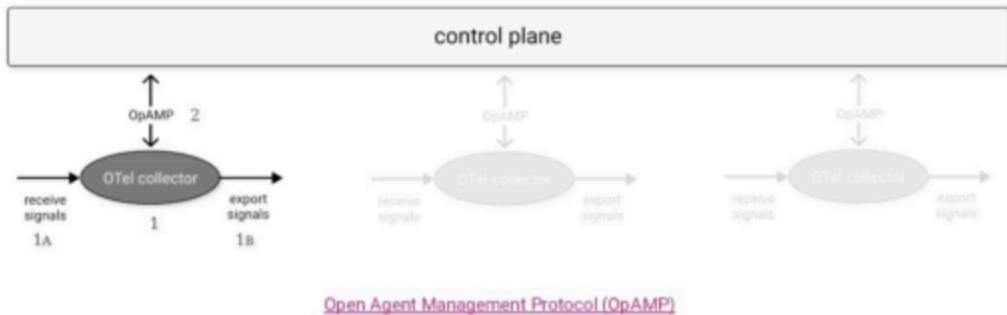
○ Create a separate module for JMH benchmarks	Feature Request	Help wanted
○ Annotations support for non-auto instrumented code	Feature Request	Help wanted
○ ImmutableBaggage.isValid allows non-zero-length strings that contain nothing but spaces and/or tabs	Blocked open	good first issue
○ Add jmx exporter	Feature Request	Help wanted
○ Investigate if we can/want to make fields and return values @Nonnull by default.	Feature Request	Help wanted
○ Use a single named logger instance for all invalid API usages	API	Feature Request
○ Prometheus Push-based MetricExporter	Feature Request	Help wanted
○ Add the ability to create more than one ZPageServer per VM	Help wanted	release after ga
○ Add OSGI support for API and SDK	Help wanted	release after ga

Collector Best Practices

Directly use the Otel to communicate data to the backend and directly instrument application code, which can be ECS, a Lambda function, etc. One negative is that you must touch the code to make any necessary modifications. Use Otel Collector's binary, which runs in a container environment, as that is the most versatile method. Any modifications you want to make now just require a runtime configuration change to the collector's configuration, not the code. Send OLTP from the application to the load balancer, which spreads out to several open telemetry collectors, for larger deployments or massive data to process. A structure approach to management would be better for them if they had numerous collectors, and they didn't all have to be open telemetry collectors like legacy or open source agents. OpAMP, which is used to handle a slew of these collectors, is a component. There is a boot strapping device on each auto collector. OpAMP is typically used for configuration patterns, health performance, and sleet management.

Collector Best Practices

Fleet management opentelemetry.io/docs/collector/management/



Trends currently seen

RUM (Real-Time User Monitoring): The focus here is on monitoring end-users in real-time. This involves tracking core web vitals and addressing issues related to page load times.

Continuous Profiling: Profiling conducted at intervals as short as every 10 seconds, or even continuously, is gaining traction. This relatively new concept is slated to become even more prominent in 2024.

Semantic Conventions (semconv): These conventions define allowable keys and values, enhancing consistency and standardization in telemetry data.

Common Trends?

- Real User Monitoring (RUM)
https://docs.google.com/document/d/16Vsdh-DM72AfMq_Flt9yT9ExEWF4A_vRbQ3jRNBe09w/
- Continuous Profiling (CP)
<https://github.com/open-telemetry/oteps/blob/main/text/profiles/0212-profiling-vision.md>
- Semantic Conventions (semconv)
 - <https://opentelemetry.io/docs/specs/semconv/>
 - <https://www.youtube.com/channel/UCHZDBZTIfdy94xMjMKz-MA/videos>
 - <https://docs.google.com/document/d/10xG7DNKWRhxNmFGt3yYd3980a9uwS8IMI2LvQL3VNK8/>
 - Breaking changes to be expected
 - ESC merge into OTEL semconv

Conclusion

Observability, powered by OpenTelemetry, is at the forefront of how we understand and manage complex systems in times to come. As trends continue to evolve, embracing this technology puts organizations for a data-rich and informed future.



EVENTS



[【1/29（木）】クラスメソッドの会社説明会を開催します](#)

開催前



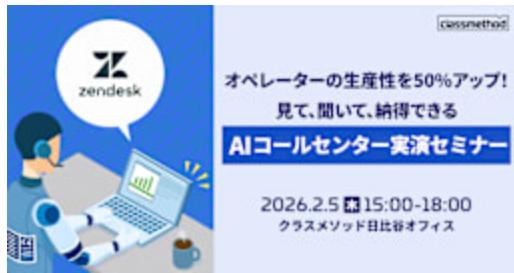
[【1/28（水）】クラスメソッドの新卒向け会社説明会を開催します](#)

開催前



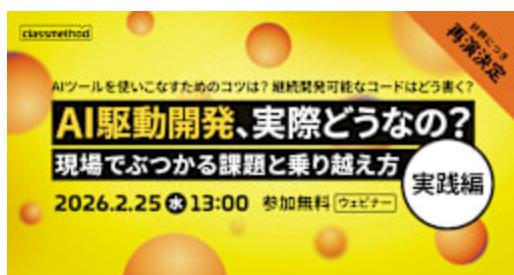
[【CMグループ/エンド直案件特集】ITフリーランス向け「CMパートナーズ」説明会 by クラスメソッド](#)

開催前



【2/5 (木) 東京】オペレーターの生産性を50%アップ！見て、聞いて、納得できるAIコールセンター実演セミナー

開催前



【2/25 (水) 】AI駆動開発、実際どうなの？【実践編】～現場でぶつかる課題と乗り越え方～

開催前



【1/29 (木) 】今日から始めるAWSセキュリティ対策 3ステップでわかる実践ガイド

開催前

[セミナー一覧](#) [会社説明会一覧](#) [勉強会一覧](#)