UNDERSTANDING OPENTRACING, OPENCENSUS, AND OPENMETRICS

When looking to improve performance monitoring and metric data collection of your cloud-based software system applications, the open source projects known as OpenTracing, OpenCensus, and OpenMetrics can assist you in making the process easier. Each used to unify instrumentation and observability through standardized APIs and formats, in the following article we will look at what these projects or tools are, how they differ, and how to use them.

Three Pillars of Observability in Software

A process of software maintenance that came along with the influx of cloud-based systems, observability is a move to solve the issue of it becoming harder and harder to monitor software as it is spread across multiple servers and microservices. Essentially a subset of monitoring that can be broken down into the action of observing and gathering three types of data in order to ensure the instrumentation, performance, and health of a system - observability includes:

1. Metrics - quantifiable pieces of data.
2. Traces - records of requests and interactions made of spans.
3. Logs/Events - specific occurrences within a span.

With these three pillars, DevOps and IT professionals can see how a system is behaving currently as well as work to foresee any potential issues in the future. To further assist with observability and

instrumentation, the cloud community came up with open source projects in hopes to standardize the process of collecting these pillars. Each project dedicated to making observability easier, it is essential to note that each project has a different focus and group of supporters.

Brief Differences

The two best-known projects, OpenTracing and OpenCensus, have made names for themselves as they both aim to create a more effective environment for monitoring distributed tracing. They seek to answer the questions regarding where the distributed event happens, where it failed, how spikes behave in your system, what service is working, what services are no longer used, and so on. When each project is used correctly, you can observe, monitor, and de-bug your system with ease.

Utilizing many of the same tracing APIs, both vendor-neutral, and both allowing you to monitor several back-ends like Zipkin and Datadog, the differences between the two projects arise when we look at who created them and how they are run.

OpenCensus is a Google Open Source community project where OpenTracing, as well as OpenMetrics, are Cloud Native Computing Foundation projects. OpenCensus, as defined by Datadog, "is a collection of language-specific libraries for instrumenting an application, collecting stats (metrics), and exporting data to a supported backend." OpenTracing is "a standardized API for tracing and provides a specification that developers can use to instrument their own services or libraries for distributed tracing. OpenTracing also provides a way for developers to collect metrics, though it's not an out-of-the-box implementation."

With that, it is clear to see the differences. The OpenTracing API makes it simple to change a preferred storage backend but relies on you to implement your own tracers that are compatible with the specifications of the project. While OpenCensus is built to include multiple backend exports automatically, it only supports the collection of various data types based on the backend and language.

Newer on the scene, OpenMetrics aims to standardize the format for metric data. It is ideally used in conjunction with OpenTracing and in the future is expected to replace Prometheus exposition format with CNCF.

To better understand each project, let's look at each individually.

OpenTracing Application

An examination of this project's general API reveals that it works to give developers and IT specialists the ability to add instrumentations that do not lock them to any language, vendor, or product. Further, this application is ideal because it gives the opportunity for teams to use and adapt tools for tracing as they evolve, meaning that you can try out a number of tools without the risk of becoming attached to code in a specific language or library.

Working to make switching tracers without changing code easy, OpenTracing does this by forming a common language around what a trace is. As the official project website states, "Traces in OpenTracing are defined implicitly by their Spans. In particular, a Trace can be thought of as a directed acyclic graph (DAG) of Spans, where the edges between Spans are called References." The relationships of these references and spans expand from there to include each behavior as a "method" in a typical programming language. With OpenTracing, now your team can apply a single

API at both service and application level, as mentioned above, and manage observability with multiple collecting and storing techniques as long as the API is compatible with your specified language.

OpenCensus Application

Ideal for both metric and trace gathering, OpenCensus is a larger scope project that intends to provide structure to observability and instrumentation through a collection of supported languages and frameworks. As stated on the project's website, "The core functionality of OpenCensus is the ability to collect traces and metrics from your app, display them locally, and send them to any analysis tool (also called a 'backend'). However, OpenCensus provides more than just data insight." In fact, this project is expanding the ability for developers and IT professionals to adopt tracing tools with automatic gathering based on a language as well as leverage metrics. In the end, it simplifies the manipulation and export of gathered data in order to better debug, understand patterns, prep for potential problems, and more. And, you can send data to more than one back-end at the same time.

OpenTelemetry

As the two projects above tackle the same issue with various overlapping mechanisms, it is only natural that in time they would combine. And, that is exactly what OpenTelemetry is. New to the world of cloud software observability, at this time we will just touch briefly on what the project's website defines itself as:

"OpenTelemetry makes robust, portable telemetry a built-in feature of cloud-native software. Providing a single set of APIs, libraries, agents, and collector services to capture distributed traces and metrics from your application." In short, the leaders of the two above mentioned projects are committed to co-producing and unifying the initiative to simplifying distributed tracing and metrics collection.

OpenMetrics Application

An initiative to transmit metrics at scale, this project utilizes text representation as well as protocol buffers. Influenced by the Prometheus exposition and still in the beta stage, in a recent article from Medium, the project is described as, "enabling all the systems to ingest and emit data in a certain wire format and agree on what that wire format should be. To talk to each other about themselves over HTTP. It does not intend to prescribe what you must do on the other end. It aims to introduce the concept of n-dimensional spaces via labels into the world." Finding a way to solve the problems we all have with a vendor-neutral guideline.

Simplifying Instrumentation and Observability

With growing community supporters and frameworks, it is easy to see that each of these projects is greatly influencing the ability of companies worldwide to easily monitor their systems. In the future, as the projects continue to merge and work towards one commonly accepted thought process, things will only become more simplified with minimal configuration.