Goal

This is the master document for the Agent Evaluation Working Group. The goal is to discuss criteria for evaluating agents, explore options of agents interoperability and investigate each agent.

Participants

Please add your name, company, and email to this list if you would like to participate in the agent evaluation.

- Wesley Pettit, AWS, wppttt@amazon.com
- Dennis Yuan, AWS, djyuan@amazon.com
- Nikhil Dewan, AWS, nikhilde@amazon.com
- Eduardo Silva, Arm, eduardo@treasure-data.com
- Morgan McLean, Google, morganmclean@google.com
- Sergey Kanzhelev, Microsoft, sergkanz@microsoft.com
- Christian Beedgen, Sumo Logic, christian@sumologic.com
- Philip O'Toole, Google, otoolep@google.com
- Rebecca Holzschuh, New Relic, rholzschuh@newrelic.com
- Mustafa Torun, AWS, mustafat@amazon.com
- Ted Young, Lightstep, ted@lightstep.com
- Rahul Taing, Zillow, rahult@zillowgroup.com
- Zach Sherman, Timber, zach@timber.io
- Ben Johnson, Timber, ben@timber.io
- Joe Lynch, Google, joelynch@google.com
- Ling Huang, Google, lingshi@google.com
- Min Xia, AWS, xiami@amazon.com
- Yenlin Chen, AWS, yenlinc@amazon.com
- Tigran Najaryan, Splunk, tnajaryan@splunk.com
- Nick Tankersley, Splunk, ntankersley@splunk.com
- Joey Echeverria, Splunk, jecheverria@splunk.com
- Bogdan Drutu, Splunk
- John Kemnetz, Microsoft, john.kemnetz@microsoft.com

Related Resources

- Master proposal: https://docs.google.com/document/d/10ntIywKEaWfHe_hWuRO_P2JbHN2Y8vYpTAHwmaIQqGY/edit#
- OpenTelemetry Log SIG

- https://docs.google.com/document/d/1cX5fWXyWqVVzYHSFUymYUfWxUK5hT97gc23w595LmdM/edit#heading=h.lro8kgalys10
- https://gitter.im/open-telemetry/logs

Why we're doing this

The OpenTelemetry Collector captures metrics and traces from applications today and we're generally happy with it. However, as the OpenTelemetry Logging project kicks off, we need to capture logs from the sources that generate them today. This requires working with an existing agent.

Many agents exist today, and while there's existing interest / dependencies / investments from several OpenTelemetry contributors in FluentD / FluentBit (Amazon, Google, and a few others already use FluentD / FluentBit and contribute to it, and FluentD / FluentBit are already part of the CNCF), we want to make sure that we evaluate other options as well. As such, we're doing a quick survey of other logging agents.

Many of the long-term logging capabilities being developed in the Logging SIG (processing logs in a native format, receiving and exporting them, etc.) will be developed in the OpenTelemetry Collector.

Evaluation Criteria

- **Existing Feature Set**: How many key features does the agent already have?
- **Missing Feature Set**: What features does the agent need, and how hard would it be to add them?
- **Existing community adoption**
  - Which major vendors use it or are moving to use it?
  - Are there known high-scale deployments?
  - How many active contributors are there from how many different organizations? What is the activity distribution?
  - Is there a governance process or is it owned by one company?
  - Is it part of an open source body like the Apache Foundation, Linux Foundation / CNCF, etc.?
  - Will it be easy to contribute to and make changes to support OpenTelemetry?
- **Performance**
  - Does it scale across cores?
  - How does the performance change between different buffering strategies (in-memory vs. disk)
  - How much CPU is required to process 100 logs/s, 1,000 logs/s, and 10,000 logs/s

    - How much memory is required to process 100 logs/s, 1,000 logs/s, and 10,000 logs/s
    - Any glaring reliability issues?
    - Are benchmarks integrated into the development process to prevent regressions?
        - Is it easy to add benchmarks to the project?
- **Security**
    - Is the source code audited?
        - Are the dependencies used to build the agent audited? Is there a known system for tracking security issues in an automated fashion?
        - Can the security patches on the dependencies be applied quickly?
    - Are there known security risks with the agent itself (memory management & safety, overflow, etc.)
    - Can the agent transmit logs securely?
    - What are the stories for authentication, encryption, signing of agent binaries
- **Portability / compatibility**
    - Which environments and operating systems can the agent run on?
    - What runtimes are required?
    - Can the agent run on embedded systems?
    - Does the agent require any dependencies?
- **Project Management**
    - Is there a project roadmap and is it public?
    - Is there a changelog?
    - How are development priorities set and agreed upon?
    - Are issues properly maintained, triaged, and prioritized?
    - Is there a release schedule?
    - How are breaking changes scheduled and handled?
- **Development environment**
    - Are we happy with the agent's code?
    - Is the programming language suitable for continued development? More specifically,
        - Is it easy to write unit tests, integration tests, e2e tests?
        - Whether it allows us to manage dependencies in a controlled (and preferably automated) manner?
        - Allow developers to leverage existing tools to improve the stability & performance, such as perf, gdb/lldb, valgrind, LLVM sanitizers
- **Bundlability**
    - Can the agent be run as a library?
    - Can it be included in other agents?
- **Configuration and extensibility**

Desired Outcomes

Short term solution might be different from the long-term solution. We want to make this decision prior to the OT Collector GA.

- Deciding / merging (long-term) into a single collector for all telemetry types
- Deciding to use specific collectors for specific purposes (for example, using the OT collector for metrics and traces, FluentBit for logs)

Agents Under Consideration

Fluent Bit

- Doc: Fluent Bit Agent Evaluation
- Inside of the CNCF, Eduardo is part of this workstream
- Highly performant
- Appears to be where many customers and vendors (Google, Amazon, etc.) are already migrating
- Written in C

OpenTelemetry Collector + FluentBit

Info: https://opentelemetry.io/docs/collector/about/

Vector (https://vector.dev/)

- Heavily associated with timber.io, appears to be a rewrite of FluentBit in Rust
- Amazon conducted performance tests; in-memory performance was fast but overflowing to disk induced a massive degradation
- Logs and metrics data models, with plans for tracing.
- Highly performant.
- Memory-safe and correct.

FluentD

- Lots of functionality, more than even FluentBit
- Issues with throughput and resource-efficiency

CloudWatch Agent

- Consists of custom logs functionality built on top of Telegraf
- Has a subset of the functionality offered by FluentD and FluentBit

- Similar to the performance of FluentBit + Go (Dennis and Wesley will benchmark)- significantly higher (5x) memory consumption
- Written in Go, will soon be open source

DataDog Agent

- Custom (open source), written in Go
- Mark to follow up

Elastic Agent(s) / Filebeat

- Custom (open source)
- Mark to follow up

Blue Medora

- Currently closed source
- Written in Go
- Will be presented on 4/22

High-Level Goals

These are goals for the unified agent that we want to achieve long term:

- Build a unified Agent and Collector that can collect all 3 observability signals: logs, traces and metrics. We may reuse existing log collection agents to speed up time-to-market, assuming existing solutions fit our needs.
- Agent to be a single package to deploy and manage, one configuration place for all signals.
- Support modern and legacy applications: file log collection and direct-to-agent collection approaches.
- Must fit OpenTelemetry's goal of having a vendor-agnostic collection by supporting many backend protocols.
- Be compliant with OpenTelemetry semantic conventions for all signals and sources, ensuring precise correlation on the backend.
- Aligned with OpenTelemetry Collector vision.
- Is friendly with other open standards that wish to extend or be extended by our agent.
- This agent will be adopted and maintained by the developers and organizations who participate here.

Agent Use Cases for Logs

(copied from master "OpenLogging" doc)

- Be pluggable and extensible to enable anyone to develop additional plug-ins for input, transform and output
    - Should have output plugins for all major industry log analytics solutions
- Support heartbeat to identify whether log collection for an endpoint is active and current
- Enable collection of security logs, audit logs
- Low CPU and memory usage for log processing and transport
- High throughput
    - Able to collect, process and send up to 20k messages/events a second
- Support a wide array of log inputs:
    - Log files (covers Kubernetes)
        - File rotation and checkpointing
    - Syslog
    - Systemd/journald/logd/wineventlogs
    - Docker (reading log files suffices, but a proper docker log driver is nice)
- Support log transformation/filtering features
    - Filter in/out logs based on log level/pattern matching
    - Parse common log formats (apache, nginx, etc)
    - Parse multiline log messages
    - Inject metadata into logs
    - Mask data at the source for security reasons
- Support a wide-array of log destinations (please add)
    - Google Stackdriver
    - Azure Monitor
    - Kafka
    - ElasticSearch
    - Splunk
        - S2S
        - HTTP 1/2
    - AWS
        - CloudWatch Logs
        - S3
        - Kinesis/Firehose
    - Datadog
    - SumoLogic
- Support configurable local log retention, buffering and offline collection policy (fail open/close)
- Export logs to multiple backends at the same time (pluggability)
- Supports pluggable authentication to backend systems (OAuth, IAM, certs etc)
- First class support for deployment in container and Kubernetes environments

- Support for auto enrichment of logs based on endpoint topology (k8S) and instance metadata (Cloud).
- Support for serverless log collection.
- Support for instrumenting IOT device logs.
- Support log collection from remote systems such as IOT, network devices
- Offer input plug-ins for common security collectors (auditD, file activity, network activity, win security log)
- Offer input plug-ins for common EDR solutions (Crowdstrike, Microsoft defender, AWS inspector etc)
- Offer standard output plug-ins for common streaming platforms (Kafka, PubSub, Kinesis)
- Offer digitally signed and trusted agent, library and plug-in distribution
- Support live, non-disruptive upgrade and configuration of the logging agent
    - No restart configuration updates would be nice
- Offer a CI/CD and automated functional and performance tests for released agents, libraries to ensure compliance with standards
- Ability to be embedded as a library
- Offer a security bug bounty program
- Security requirements
    - Tamper-proof log delivery (digitally signing / hashing logs to prove that they haven't been changed)
    - High delivery reliability (can't lose logs)
- Broad OS support
    - Linux/Unix (32/64 bit, powerpc, ARM)
    - Windows
    - MacOS
    - Solaris
    - AIX
    - FreeBSD