OpenTelemetry Configuration Service

Author: jmontana@google.com
Date: May 15, 2020
Status: Draft
Go Link: go/otel-dynamic-config

Overview

This proposal adds support for a configuration service to OpenTelemetry. A configuration service is used to remotely configure metric export schedules and trace sampling rates for processes instrumented with OpenTelemetry. This will allow users to make decisions about tradeoffs between instrumentation fidelity and resource usage without having to redeploy or restart running jobs. Much like OpenTelemetry's vendor-agnostic metric and trace exporting, OpenTelemetry will also include a mechanism for using the collector as an optional bridge for translating arbitrary configuration backends into the OpenTelemetry configuration service protocol understood by the agent, so that any vendor can implement their own configuration service to be used by the OpenTelemetry agent.

For example, during normal use, a user could use sparse networking, CPU, and memory resources required to send and store telemetry data by specifying sampling of 0.1% of traces, collecting critical metrics such as SLI-related latency distributions and container CPU stats only every five minutes, and not collecting non-critical metrics. Later, while investigating a production issue, the same user could easily increase information available for debugging by reconfiguring some of their processes to sample 2% of traces, collect critical metrics every minute, and begin collecting metrics that were not deemed useful before. Because this change is centralized and does not require redeploying with new configurations, there is lower friction and risk in updating the configurations.

Design

Changes required to support a configuration service are broken down into three parts: (1) specifying a configuration service protocol, (2) updating the agent/SDK to use the new protocol to drive collection frequencies, and (3) updating the collector to allow it to act as a shim configuration service for the agent. This design follows a similar model to metric and trace exporting, which also support shimming from the protocol used between agent and collector to an arbitrary protocol between the collector and the monitoring backend.

OpenTelemetry Configuration Service Protocol

A new configuration service protocol will be added to the OpenTelemetry specification. A configuration service implementing the protocol takes in a Resource consisting of key-value attributes, and returns a list of metric collection schedules, a list of trace sampling configurations, and a recommended wait time to cache results before querying the configuration service again.

Each metric collection schedule will consist of the following:

- Inclusion patterns consisting of lightweight rules for identifying metrics to which the schedule applies (e.g., "exactly matches cpu_usage", "all metrics", or "metrics that start with cpu_").
- Exclusions patterns consisting of lightweight rules for identifying metrics to which the schedule does not apply, even if they match an inclusion pattern.
- The period at which applicable metrics should be collected/exported, e.g., 1 hour, 30 minutes, 10 minutes, 5 minutes, 1 minute, 30 seconds. Longer periods must be divisible by all shorter periods.
- Possibly some opaque metadata (e.g., "key") that is useful to a specific monitoring backend, and which would be passed into an exporter during metric writes.

Trace sampling configurations will consist of the following:

- Inclusion patterns consisting of lightweight rules for identifying which traces apply based on sampling parameters.
- The strategy and extent of sampling that should be done - most likely a sampling probability.

OpenTelemetry Agent/SDK

The OpenTelemetry agent may be configured to periodically query a configuration service endpoint for updates to configurations. For backends that use an alternative configuration protocol, the collector may be configured as a configuration service, acting as a shim to the alternative backend service.

For metrics, the OpenTelemetry agent will support periodically querying a configuration service to determine how frequently metrics should be collected and exported, as an alternative to the current method of exporting all metrics at a fixed period.

For traces, a new Sampler implementation similar to ProbabilitySampler will support reading from a configuration service to determine which trace sampling configuration best applies, using the probability returned by the configuration service instead of a fixed probability.

OpenTelemetry Collector

The collector will support a new interface for a ConfigurationService that can be used by an agent, allowing a custom implementation of the configuration service protocol described above, to act as an optional bridge between an agent and an arbitrary configuration service. This interface can be implemented as a shim to support accessing remote configurations from arbitrary backends. The collector is configured to expose an endpoint for requests to the ConfigurationService, and returns results on that endpoint.

The collector will support both implementing a standalone ConfigurationService, and combining ConfigurationService and Exporter implementations for monitoring and tracing backends that have integrated remote configurations.

Related Resources

- Specification issue #372: Remote sampling specification for proto layout
- Collector issue #273: Watch config file for changes.
- WIP config reloading pull request by Shopify.