

Stackdriver Trace, a distributed tracing system for Google Cloud, helps you understand how long it takes your application to handle incoming requests from users or other applications, and how long it takes to complete operations like RPC calls performed when handling the requests.

Because Stackdriver Trace collects latency data from [App Engine](/appengine/docs/), [HTTP\(S\) load balancers](/compute/docs/load-balancing/http/), and applications instrumented with the [Stackdriver Trace API](/trace/api/), it can help you answer the following questions:

- How long does it take my application to handle a given request?
- Why is it taking my application so long to handle a request?
- Why do some of my requests take longer than others?
- What is the overall latency of requests to my application?
- Has latency for my application increased or decreased over time?
- What can I do to reduce application latency?
- What are my application's dependencies?

For information about debugging or profiling your application, go to [Application Performance Management](/apm/).

Stackdriver Trace runs on Linux in the following environments:

- [Compute Engine](/compute/docs/)
- [Google Kubernetes Engine \(GKE\)](/kubernetes-engine/docs/)
- [App Engine flexible environment](/appengine/docs/flexible/)
- [App Engine standard environment](/appengine/docs/standard/)
- Non-Google Cloud environments

Stackdriver Trace provides client libraries for instrumenting your application to capture trace information. For per-language setup instructions, go to [Setting up Stackdriver Trace](/trace/docs/setup/).

In the App Engine [standard environment](/appengine/docs/standard/) (/appengine/docs/standard), Java 8, Python 2, and PHP 5 applications don't need to use the Stackdriver Trace client libraries. These runtimes automatically send latency data to Stackdriver Trace for requests to application URIs. The requests include latency data for round-trip RPC calls to App Engine services. Stackdriver Trace works with all App Engine Admin APIs, with the exception of Cloud SQL.

The following table summarizes the availability of Trace client libraries and [OpenCensus](https://opencensus.io/) (https://opencensus.io/) libraries:

Language	Client library available	OpenCensus library available
C# ASP.NET Core (/trace/docs/setup/aspnetcore)	Yes	No
C# ASP.NET (/trace/docs/setup/aspnet)	Yes	No
Go (/trace/docs/setup/go)	Yes	Yes
Java (/trace/docs/setup/java)	Yes	Yes
Node.js (/trace/docs/setup/nodejs)	Yes	Yes
PHP (/trace/docs/setup/php)	Yes	Yes
Python (/trace/docs/setup/python)	Yes	Yes
Ruby (/trace/docs/setup/ruby)	Yes	No

[OpenCensus](https://opencensus.io/) (https://opencensus.io/) libraries are simpler to use because they hide some of the complexity of the corresponding Stackdriver Trace client library.

For general setup instructions, go to [Setting up Stackdriver Trace](/trace/docs/setup/) (/trace/docs/setup).

Stackdriver Trace consists of a tracing client, which collects the data and sends it to your Google Cloud project, and a console interface on Google Cloud, which lets you view and analyze the data collected by the agent.

If an OpenCensus library is [available for your programming language](#) (#language_support), you can simplify the process of creating and sending trace data by [using OpenCensus](#) (https://opencensus.io/). In addition to being simpler to use, OpenCensus implements batching which might improve performance.

If an OpenCensus library doesn't exist, instrument your code by importing the Trace SDK library and using the Stackdriver Trace API. The Stackdriver Trace API collects trace data and sends it to your Google Cloud project.

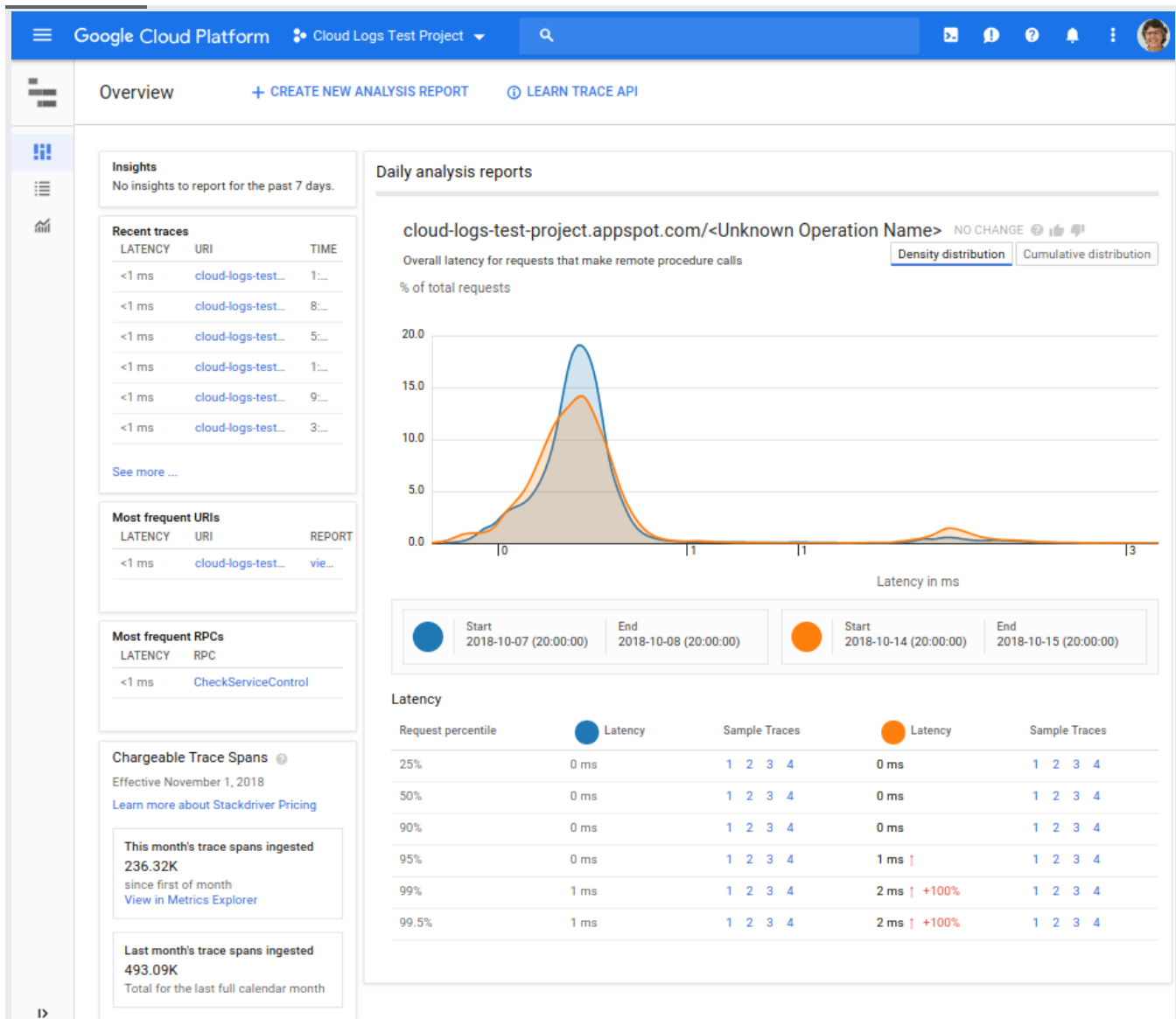
After the agent has collected trace data, you can view and analyze that data in near real-time in the Stackdriver Trace interface. The interface contains three windows: **Overview**, **Trace list**, and **Analysis reports**.

To get started using Stackdriver Trace, in the Cloud Console, **Trace**:

[Go to Trace](https://console.cloud.google.com/traces) (https://console.cloud.google.com/traces)

The Trace console won't display data unless you have access to a Google Cloud project that is receiving Trace data.

The **Overview** window provides summary information about your application:



The **Insights** pane displays a list of performance insights for your application, if applicable. This pane highlights common problems in applications, such as consecutive calls to a function, that if batched, might be more efficient.

The **Recent traces** pane displays the most recent traces. For each, the latency, URI, and time are displayed. You can use this summary to understand the current activity in your application.

The **Most frequent URIs** and **Most frequent RPCs** list the most frequent URIs (and RPCs) from the previous day, along with the average latency. If you click a link in either of these tables, you open the **Trace list** window which is preconfigured to display *matching requests*. Using the **Trace list** window, you can view latency as a function of time, and investigate details of any individual trace.

The **Chargeable Trace Spans** pane displays the number of spans ingested in the current calendar month, and the total for the previous month. You can use this information to monitor your costs for

using Stackdriver Trace. For more information on costs, go to the [Pricing \(#ingest_and_scan\)](#) section on this page.

The **Daily analysis reports** pane displays latency data for the previous day and compares it to the latency data from 7 days prior. Using this data, you can view if your latency behavior is changing over time. You can't change the format of the reports on this page. However, you can create custom analysis reports.

To learn more about the Stackdriver Trace **Overview** window, go to [Viewing Traces in the Cloud Console \(/trace/docs/trace-overview\)](#).

The **Trace list** window lets you examine an individual traces in detail. By default, this view displays a plot of the latency for all traces and lists, in a table, the most recent traces.

If you select a specific trace to examine, two new panes are opened: a **Timeline** pane that displays all spans in the trace and an information pane that displays details about the selected span. By default, the root span is selected and the details include summary information about the request. To find traces for investigation, you use filters.

For more information on filtering and viewing trace details, go to [Finding and viewing traces \(/trace/docs/finding-traces\)](#) and [Viewing trace details \(/trace/docs/viewing-details\)](#).

The **Analysis reports** window lets you to create custom reports. These reports can include all traces, or by specifying filters, only include a subset of traces. You can also create reports that compare two different sets of latency data. For more information, go to [Creating analysis reports \(/trace/docs/analysis-reports\)](#).

Latency data is packaged into a *trace*, which is composed of spans. The spans are *ingested* by the Stackdriver Trace backend which then makes the data available for real-time viewing and reporting in the Cloud Console. When you view trace data, the stored spans are *scanned* as part of the retrieval process.

Stackdriver Trace pricing is determined by the number of Trace spans ingested, and scanned, on a monthly basis. For more information, go to [Trace pricing \(/stackdriver/pricing#trace-costs\)](#).

You can create an alert to notify you if your ingested span count exceeds a predefined threshold. For more information, go to [Creating alerting policies \(/trace/docs/trace-alerting\)](/trace/docs/trace-alerting).

Stackdriver Trace enforces a rate quota on your of API requests and an ingestion quota on the spans you sent to Trace. For more information, see [Quotas and limits \(/trace/docs/quotas\)](/trace/docs/quotas).

To view your current usage:

1. In the Google Cloud Console, select **APIs & Services**, and then select **Stackdriver Trace API**, or use the following button:

[Go to Trace API \(https://console.cloud.google.com/apis/api/cloudtrace.googleapis.com/overview\)](https://console.cloud.google.com/apis/api/cloudtrace.googleapis.com/overview)

2. Click **Quotas**.

You can create an alert to notify you if you have exceeded your quotas. For more information, go to [Creating alerting policies \(/trace/docs/trace-alerting\)](/trace/docs/trace-alerting).

Product or feature is in a pre-release state and might change or have limited support. For more information, see the [product launch stages \(/products/#product-launch-stages\)](/products/#product-launch-stages).

Stackdriver Trace is a VPC Service Controls supported service. For more information, see [VPC Service Controls documentation \(/vpc-service-controls/docs/\)](/vpc-service-controls/docs/).

- To learn more about the Stackdriver Trace console, go to [Viewing traces in the Cloud Console \(/trace/docs/trace-overview\)](/trace/docs/trace-overview).
- To get started using Stackdriver Trace, in the Cloud Console, go to **Trace**, or use the following button.

[Go to Trace \(https://console.cloud.google.com/traces\)](https://console.cloud.google.com/traces)

- Try the [Quickstart \(/trace/docs/quickstart\)](/trace/docs/quickstart).