

[Management Tools](https://cloud.google.com/products/management/) (https://cloud.google.com/products/management/)

[Stackdriver: Observability Suite](https://cloud.google.com/stackdriver/) (https://cloud.google.com/stackdriver/)

[Stackdriver Trace](https://cloud.google.com/trace/docs/) (https://cloud.google.com/trace/docs/)

[Documentation](https://cloud.google.com/trace/docs/) (https://cloud.google.com/trace/docs/) [Guides](#)

Creating and viewing analysis reports

Analysis reports in Stackdriver Trace show you an overall view of the latency for all requests, or for a subset of requests, to your application.

Note: Trace must have at least 100 stored traces that match an analysis reports' filter parameters. When there are fewer than 100 traces that match, the creation of the report fails and a warning message is displayed. You select the filter parameters for custom analysis reports.

Daily reports

Trace automatically creates a daily report that compares the previous day's performance with the performance from the same day of the previous week for the top three endpoints. The daily report is displayed on the Trace **Overview** window and is listed in the **Analysis report** window.

For details on viewing the daily analysis report, go to [Viewing a report](#) (#view_an_analysis_report).

Custom analysis reports

The daily analysis report is created automatically and doesn't let you choose the content of the report. However, you can create a custom analysis report and select which traces are included in the report.

Custom reports include:

- A histogram of latency data.
- A table that list the latency by percentile along with links to sample traces.
- (Optional) Latency data from a different time range that you use for comparison.
- A bottleneck pane that lists the RPC calls that are significant contributors to latency.

You can create a new report by entering your report parameters manually, or you can duplicate an existing report and edit the parameters. Both of these approaches are described in the following sections.

Creating a new report

To create a new analysis report:

1. In the Google Cloud Console, go to **Trace**:

[GO TO TRACE \(HTTPS://CONSOLE.CLOUD.GOOGLE.COM/TRACES\)](https://console.cloud.google.com/traces)

2. On the **Analysis reports** tab, click **New Report**.

3. Enter or select the following parameters in the form that appears:

- **Request filter**: The full URI, the URI prefix, or trace filter terms (<https://cloud.google.com/trace/docs/trace-filters>).

For example, when a project root span is `Recv`, to include all requests, enter the root span name, `Recv`, or a prefix of the root span name. To restrict the report to traces with a URI prefixed by `Recv/cart`, enter that prefix into the filter.

You can also filter traces by span attributes. For example, when you add `/http/status_code:5`, the report includes only those traces that have a span with the attribute key `/http/status_code` and attribute values prefixed by `5`.

- **HTTP method**: HTTP method for the request. Default is **All**.
- **HTTP Status**: HTTP status returned by the request. Default is **All**.
- **Report Name**: Human-readable name for the report. When no name is specified for this optional field, Trace creates a name based on the specified request filter and the time at which the report was created.
- **Module**: App Engine module (Java (<https://cloud.google.com/appengine/docs/java/modules/>), Python (<https://cloud.google.com/appengine/docs/python/modules/>), Go (<https://cloud.google.com/appengine/docs/go/modules/>) or PHP (<https://cloud.google.com/appengine/docs/php/modules/>)) that handled the request. Default is **All**.

- **Version:** Version of the App Engine application that handled the request. Default is **All**.
 - **Time Range:** Time range during which the requests occurred. You can select **Last Hour**, **Last 24 Hours**, **Last Week** or **Custom Range**. When you select **Custom**, you must specify the start and end times for the range. Default is **Custom**.
4. (Optional) To compare two sets of latency data, select **Compare to baseline** and enter the time range for the comparison data.
 5. Click **Submit**.

Duplicating a report

In Trace, you can duplicate the filter parameters of an existing report:

1. In the Google Cloud Console, go to **Trace**:

[GO TO TRACE \(HTTPS://CONSOLE.CLOUD.GOOGLE.COM/TRACES\)](https://console.cloud.google.com/traces)

2. On the **Analysis reports** tab, click the name of the report whose structure you want to duplicate.
3. Click **Create similar report**.
4. In the **New report request** window, edit one or more report parameters, including the name of the report.
5. Click **Submit**.

Viewing a report

To view an analysis report:

1. In the Google Cloud Console, go to **Trace**:

[GO TO TRACE \(HTTPS://CONSOLE.CLOUD.GOOGLE.COM/TRACES\)](https://console.cloud.google.com/traces)

2. To view the list of system-generated daily reports and custom reports available for viewing, click the **Analysis Reports** tab.
3. To view a report, click the report name.

View by density distribution

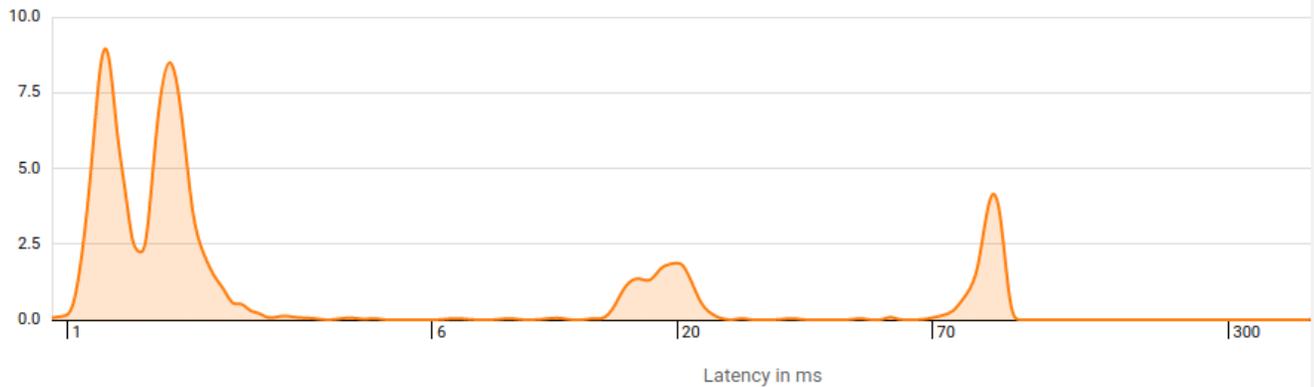
By default, the **Density distribution** button is enabled. With this setting, the x-axis is the trace latency on a logarithmic scale while the y-axis is the percent of total requests with that latency:

Recv.hipstershop.ProductCatalogService.ListProducts

Density distribution Cumulative distribution

Overall latency for requests

% of total requests



When you create a comparison report, Trace overlays the data from the two data sets so that you can compare them and analyze any changes:

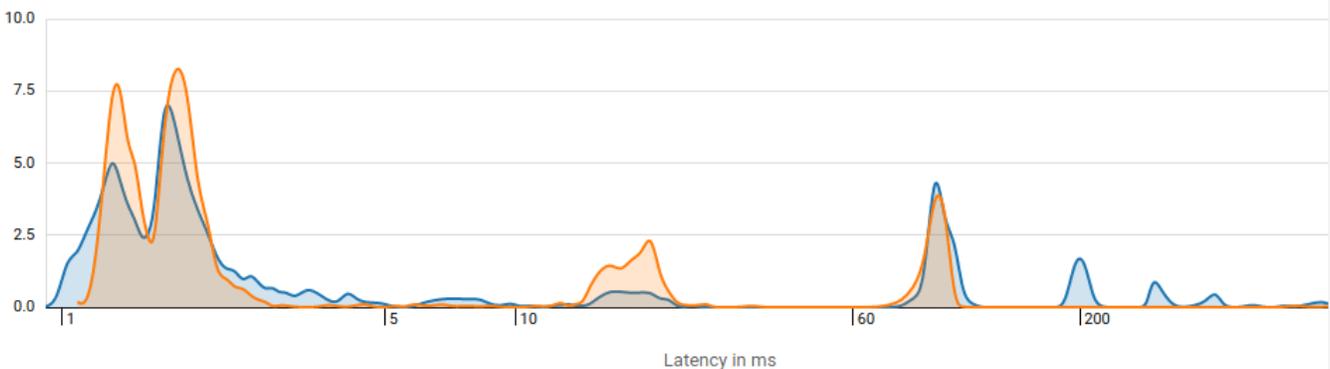
Recv.hipstershop.ProductCatalogService.ListProducts

NO CHANGE

Density distribution Cumulative distribution

Overall latency for requests

% of total requests



In the comparison report, the legend on the report identifies the time range for the orange and blue graphs. These have been omitted in the screenshot.

View by cumulative distribution

To view the total number of requests whose latency is less than or equal to the latency value on the x-axis, click **Cumulative distribution**:

Recv.hipstershop.ProductCatalogService.ListProducts

NO CHANGE

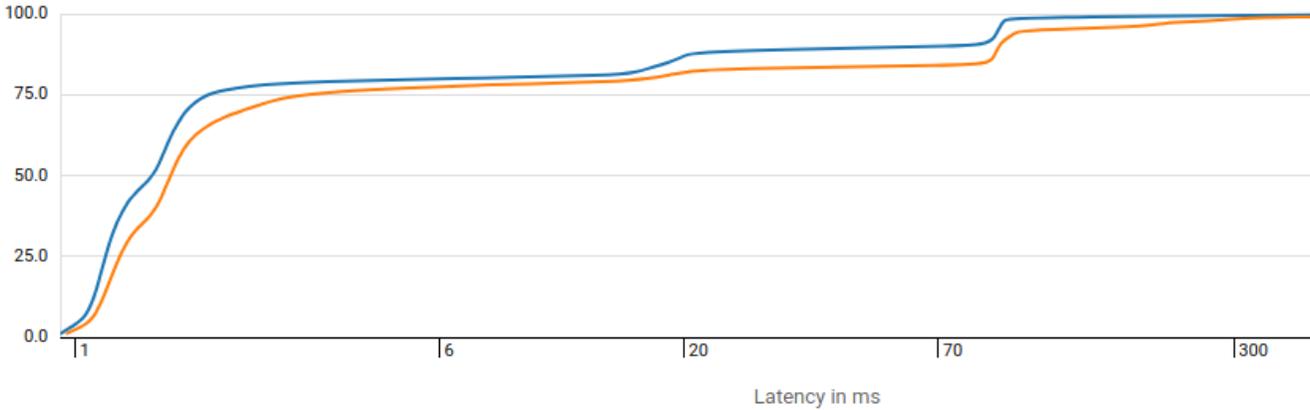


Cumulative latency for requests that make remote procedure calls

Density distribution

Cumulative distribution

% of total requests



Latency pane

The **Latency** pane beneath the graph view shows latency percentiles, and it includes links to traces for representative sample requests. When you click a trace number, a **Trace details** window opens that displays the trace **Timeline** and information about the root span. For more detail on viewing the **Timeline**, go to [trace details view](https://cloud.google.com/trace/docs/viewing-details) (<https://cloud.google.com/trace/docs/viewing-details>).

The following image shows a latency pane for an analysis report. In this case, there is only one data set which is indicated by the letter **A**:

Latency			
Request percentile	A Latency	Sample Traces	
25%	22 ms	1	2 3 4
50%	26 ms	1	2 3 4
90%	36 ms	1	2 3 4
95%	54 ms	1	2 3 4
98%	115 ms	1	2 3 4
99%	121 ms	1	2 3 4

When you create a comparison report, the latency data pane contains information for both data sets. The legend for the pane identifies the entries that belong to each of the data sets.

Bottlenecks pane

A **Bottleneck** indicates that the latency value for a span in a specific trace falls outside the expected range of latency values. Stackdriver Trace determines the expected range of latency values by using statistical techniques that rely on computing and comparing empirical latency distributions for subsets of traces.

The **Bottlenecks** pane, when present, lists RPC calls that are significant contributors to latency in the requests. This pane displays the RPC calls by percentile with representative sample requests. You can click a number under **Sample traces** to view the details of a specific trace.

The following image shows a **Bottlenecks** pane for an analysis report. In this case, there is only one data set which is indicated by the letter A:

Bottlenecks					
Request percentile	 Bottleneck	Sample Traces			
50%	18 ms on /datastore_v3.RunQuery	1	2	3	4
90%	25 ms on /datastore_v3.RunQuery	1	2	3	4
99%	25 ms on /datastore_v3.RunQuery	1	2	3	4

In a comparison report, the bottlenecks highlights the RPC calls that caused latency distribution changes between the two sets. The legend on this pane identifies the entries that belong to each of the data sets.

What's next

- [Setting up Stackdriver Trace](https://cloud.google.com/trace/docs/setup) (https://cloud.google.com/trace/docs/setup)
- [Viewing traces across projects](https://cloud.google.com/trace/docs/cross-project-traces) (https://cloud.google.com/trace/docs/cross-project-traces)
- [Creating alerting policies](https://cloud.google.com/trace/docs/trace-alerting) (https://cloud.google.com/trace/docs/trace-alerting)

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (https://developers.google.com/terms/site-policies). Java is a registered trademark of Oracle and/or its affiliates.