

This page describes how to use a [Zipkin](http://zipkin.io/) (http://zipkin.io/) server to receive traces from Zipkin clients and forward those traces to Stackdriver Trace for analysis.

You might want to use a Zipkin server if your application is already instrumented with Zipkin and either you don't want to run your own trace backend or you want access to Stackdriver Trace's advanced analysis tools.

This page describes several ways to set up your Zipkin server:

- [Using a container image](#) (#container-image-option)
- [Running outside of Google Cloud](#) (#non-GCP-option)
- [Modifying an existing Zipkin server](#) (#existing-zipkin-server)

This functionality is maintained by the Zipkin project itself and isn't officially supported by Google.

A container image of the Stackdriver Trace Zipkin Collector is available on GitHub. This [repository](https://github.com/openzipkin/docker-zipkin-gcp) (https://github.com/openzipkin/docker-zipkin-gcp) contains the Docker build definition and layers Google Cloud support on the base Zipkin Docker image, in addition to detailed setup steps.

You can run this on your container host of choice, including Google Kubernetes Engine.

To run the image:

If you are running this container within the Google Cloud, such as on a Compute Engine instance or Google Kubernetes Engine cluster, the environment's default credentials are automatically captured and traces are automatically sent to Stackdriver Trace.

For the full setup process, go to the [GitHub repository for the Zipkin Docker image](https://github.com/openzipkin/docker-zipkin-gcp) (<https://github.com/openzipkin/docker-zipkin-gcp>).

As described on this page, you must also [configure your Zipkin tracers](#) ([#how_to_configure_zipkin_tracers](#)).

If you would like to build and run the collector outside of Google Cloud, such as on a physical server running on-premises, complete the following steps:

1. [Sign in](https://accounts.google.com/Login) (<https://accounts.google.com/Login>) to your Google Account.

If you don't already have one, [sign up for a new account](https://accounts.google.com/SignUp) (<https://accounts.google.com/SignUp>).

2. In the Cloud Console, on the project selector page, select or create a Cloud project.

★ **Note:** If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

[Go to the project selector page](https://console.cloud.google.com/projectselector2/home/dashboard) (<https://console.cloud.google.com/projectselector2/home/dashboard>)

A billing account isn't required as the collector won't be running on Google Cloud.

To allow your service to authenticate to the Stackdriver Trace API:

1. [Create a service account](/iam/docs/creating-managing-service-accounts) (</iam/docs/creating-managing-service-accounts>).

- a. Ensure that the new service account has been [granted a project editor role](#) (/iam/docs/granting-roles-to-service-accounts#granting_access_to_a_service_account_for_a_resource)

so that it can write data to the Trace API.

- b. Select **Furnish a new private key** and choose JSON.
- c. Save the JSON credentials file to a directory on the machine that will be running the collector service.

Configure your network configuration to allow TCP traffic on port 9411 to pass to the machine running the Zipkin collector.

If you'll be submitting traces from applications hosted outside of the firewall, note that Zipkin tracer-to-collector traffic is not encrypted or authenticated. Connections between the Stackdriver Trace Zipkin collector and the Stackdriver Trace API are encrypted and authenticated, as are connections sourced from the Stackdriver Trace instrumentation libraries.

See [Using a container image](#) (#container-image-option), above.

Follow the instructions in the common [Configure Zipkin tracers](#) (#how_to_configure_zipkin_tracers) section on this page.

The Zipkin project maintains instructions on how to use Stackdriver Trace as a storage destination for an existing Zipkin server. These are available on the [GitHub repository for the Zipkin Docker image](#) (<https://github.com/openzipkin/docker-zipkin-gcp>).

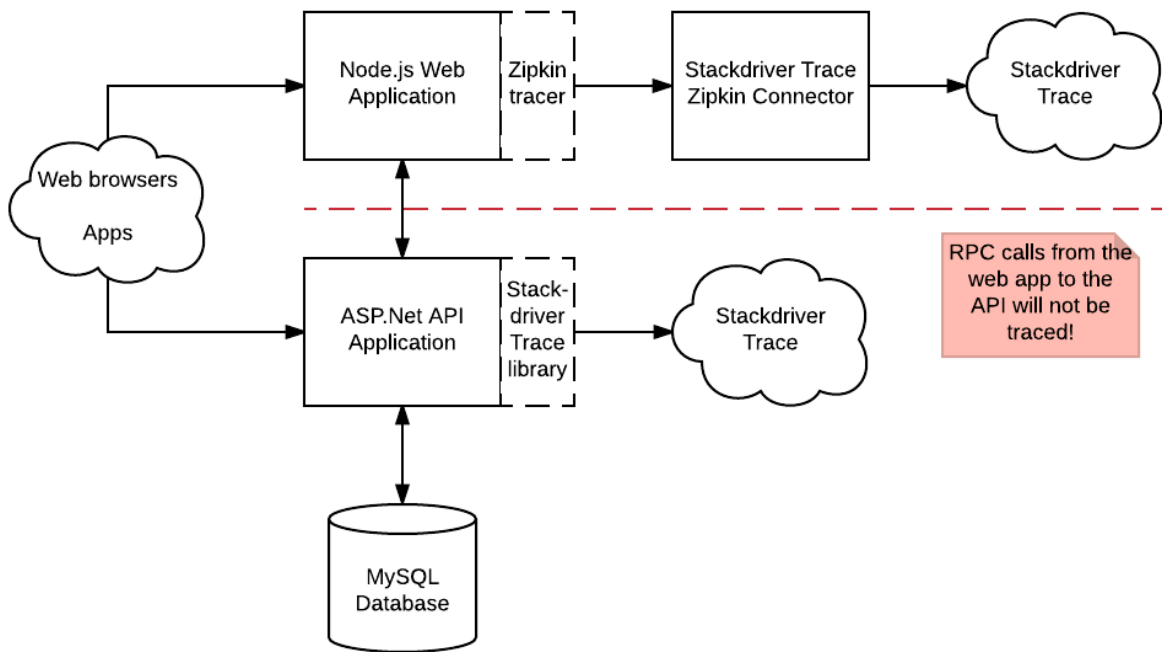
No matter how you host the Stackdriver Trace Zipkin Collector, you'll need to configure your Zipkin tracers to send data to it.

The collector can be referenced by its internal IP address, external IP address (if it will be receiving traces from applications hosted outside of Google Cloud), or hostname. Each Zipkin tracer is configured differently - for example, to point a Brave tracer at a collector with the IP address 1.2.3.4, the following lines must be added to your Java codebase:

Q: What are the limitations?

This release has two known limitations:

1. Zipkin tracers must support the correct Zipkin time and duration semantics. For more information, go to [Instrumenting a library](https://zipkin.io/pages/instrumenting.html). (<https://zipkin.io/pages/instrumenting.html>) and scroll down to the section on **Timestamps and duration**.
2. Zipkin tracers and the Stackdriver Trace instrumentation libraries can't append spans to the same traces, meaning that traces that are captured in one library won't contain spans for services instrumented in the other type of library. For example:



Requests made to the Node.js web application will be traced with the Zipkin library and sent to Stackdriver Trace. However, these traces will not contain spans generated within the API application or for the RPC calls that it makes to the Database. This is because Zipkin and Stackdriver Trace use different formats for propagating trace context between services.

For this reason we recommend that projects wanting to use Stackdriver Trace either exclusively use Zipkin-compatible tracers along with the Zipkin Collector, or use instrumentation libraries that work natively with Stackdriver Trace (like the official [Node.js](https://github.com/GoogleCloudPlatform/cloud-trace-nodejs) (<https://github.com/GoogleCloudPlatform/cloud-trace-nodejs>), [Java](https://github.com/GoogleCloudPlatform/cloud-trace-java) (<https://github.com/GoogleCloudPlatform/cloud-trace-java>), or [Go](https://godoc.org/cloud.google.com/go/trace) (<https://godoc.org/cloud.google.com/go/trace>) libraries).

Q: Will this work as a full Zipkin server?

No, this feature only writes data to Stackdriver Trace.