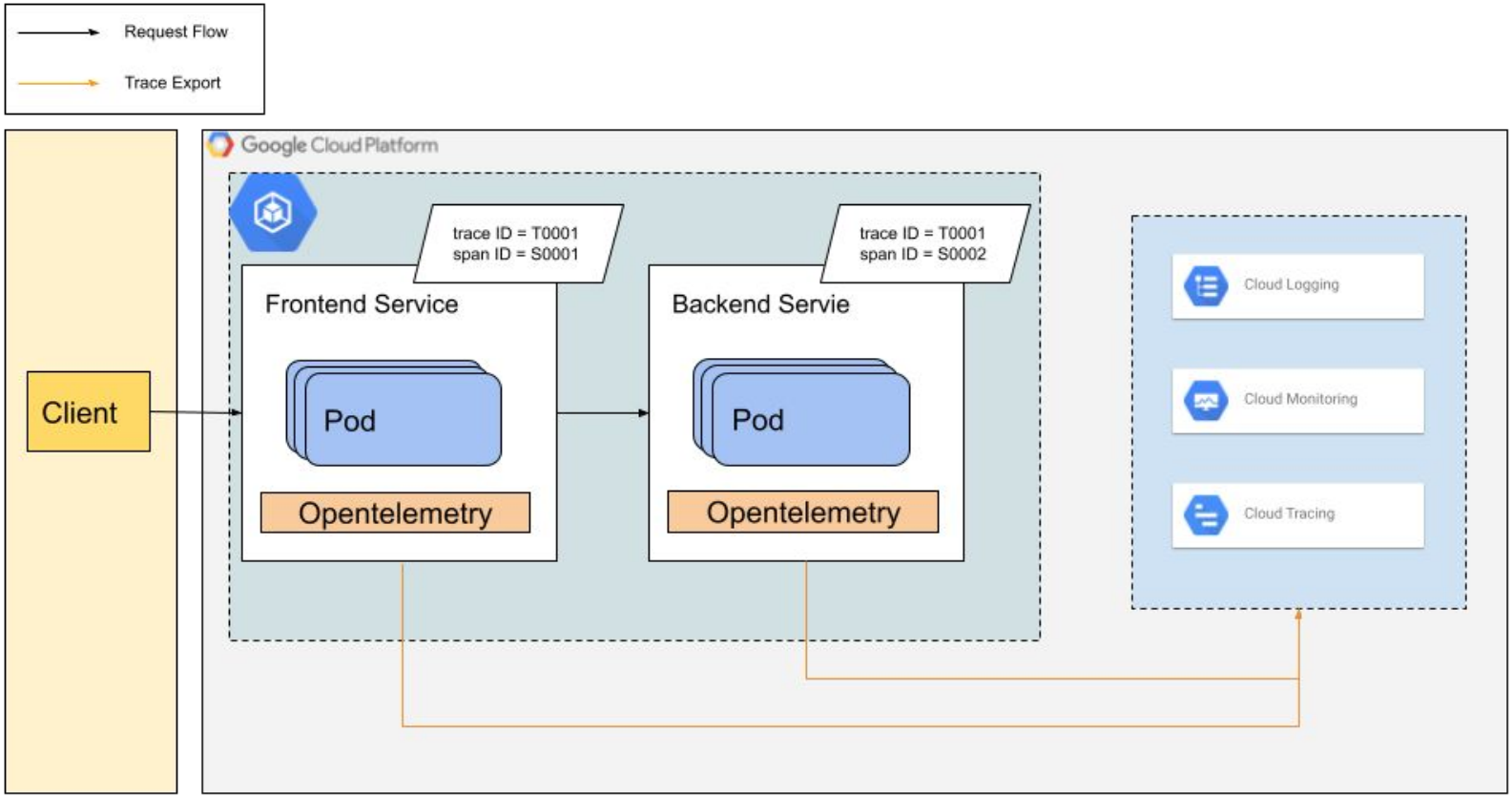
Microservice tracing on Cloud Trace with Opentelemetry

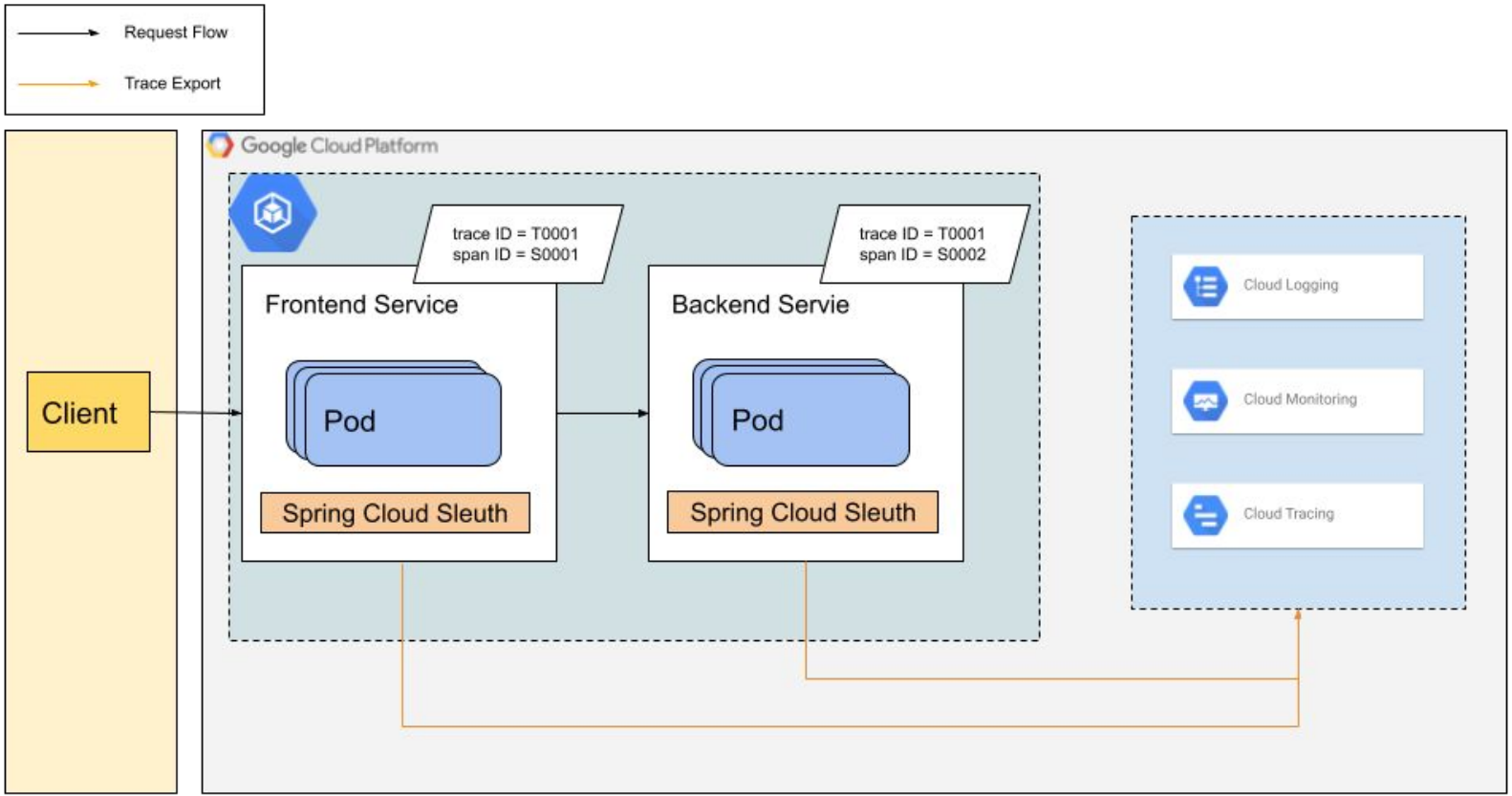
Purpose

This guide is to describe the steps of collecting and exporting trace data from Java distributed applications to Cloud Trace.

Demo Architecture (Opentelemetry)



Demo Architecture (Spring Cloud Sleuth)



Java demo application

gRPC's Java Quick start example, which uses gRPC to communicate between the client and server

Enable Cloud Trace API

- Click the following button or, in the Google Cloud console, select APIs & Services, and then select Cloud Trace API.
- On the Cloud Trace API page, if a button labeled Enable is displayed, then click it. If this button isn't displayed, then the Cloud Trace API is enabled for the selected project.

Install the OpenTelemetry packages

- Maven:

```
<dependency>
  <groupId>com.google.cloud.opentelemetry</groupId>
  <artifactId>exporter-trace</artifactId>
  <version>0.15.0</version>
</dependency>
```

- Gradle:

```
dependencies {
  implementation platform("io.opentelemetry:opentelemetry-bom:1.22.0")
  implementation('io.opentelemetry:opentelemetry-api')
 }
```

Configure the export of spans to Cloud Trace

- To export the collected Trace data, Create a TraceExporter object:

```
TraceExporter traceExporter = TraceExporter.createWithConfiguration(
  TraceConfiguration.builder().setProjectId("MY_PROJECT").build());
```

- After the exporter is configured, set the TracerProvider:

```
OpenTelemetrySdk.builder()
        .setTracerProvider(
            SdkTracerProvider.builder()
                .addSpanProcessor(SimpleSpanProcessor.builder(traceExporter).build())
                .build())
        .buildAndRegisterGlobal();
```

Create spans

- Create a span
- Create nested spans
- Set span attributes
- Create spans with events
- Create spans with links

Reference:
https://opentelemetry.io/docs/instrumentation/java/manual/#acquiring-a-tracer

Configure sampling

- Don't use the AlwaysOnSampler sampler in a production environment. This sampler can generate a large volume of trace data and increase the cost.

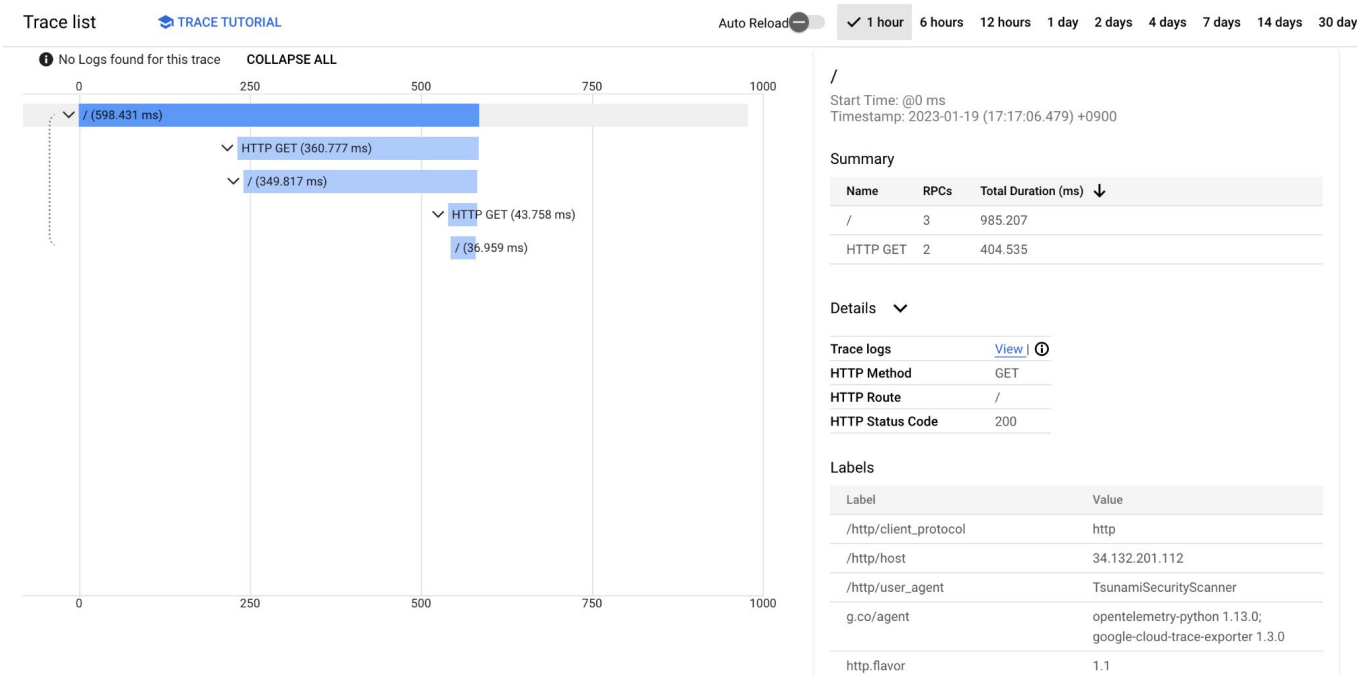
Reference: https://opentelemetry.io/docs/instrumentation/java/manual/#sampler

Deploy Java application to GKE

- Prepare Dockerfile
- Create Pod
- Create Deployment and Services

Viewing the traces from Cloud Trace

In Cloud Trace, The traces should be displaying like the example below:



GCP Environment for Demo

- Create a GCP demo project
- GCP project User Account for demo
- GCP project Service Account for demo
- Enable Cloud Trace API
- Others...