



LAB 23: QUARKUS MONITOR TRACE

Autor: José Díaz

Github Repo: <https://github.com/喬日斯/quarkus-bcp-2025.git>

Abre el proyecto **20-monitor-trace-start**

1. Abrelo en tu editor favorito.

El proyecto Quarkus calculator contiene 3 microservicios:

solver

Evalúa una expresión dada. Devuelve el valor si la expresión es un número decimal, o delega las expresiones de suma y multiplicación a los servicios correspondientes.

adder

Obtiene dos ecuaciones y devuelve la suma de sus resultados. Depende del microservicio **solver** para resolver ambos lados de la suma.

multiplier

Obtiene dos ecuaciones y devuelve el producto de sus resultados. Depende del microservicio **solver** para resolver ambos lados de la multiplicación.

Los tres microservicios exponen su propia API REST. Una sola llamada del cliente para resolver una ecuación podría resultar en múltiples llamadas entre estos microservicios.

2. Agrega la extensión **opentelemetry** a los tres servicios para habilitar el tracing.

- a. Ejecuta el **./add-tracing.sh** (linux o macosx) o **./add-tracing.ps1** (windows).

3. Inicia la instancia local de **jaeger** usando podman o docker.

- a. Inspecciona el script **jaeger.sh**, el cual inicia Jaeger en un contenedor.
- b. En una nueva terminal inicia el contenedor usando el script **jaeger.sh** (linux o macosx) o **jaeger.ps1**(windows).

```
[student@workstation ~]$ sh ~/D0378/monitor-trace/jaeger.sh
Starting the all-in-one Jaeger container
...output omitted...
"Starting jaeger-collector HTTP server","http host-port":"14268"}
...output omitted...
"Query server started","port":16686,"addr":"16686"}
"Health Check state change","status":"ready"
...output omitted...
```

4. Configura el servicio adder enviando información de tracing a Jaeger.

- a. Edita el **src/main/resources/application.properties** y agrega las siguientes propiedades.

```
# Enable Tracing (OpenTelemetry)
quarkus.otel.service.name=adder
quarkus.otel.traces.sampler=traceidratio
quarkus.otel.traces.sampler.arg=1
quarkus.log.console.format=%d{HH:mm:ss} %-5p
traceId=%X{traceId}, spanId=%X{spanId},
parentId=%X{parentId}, sampled=%X{sampled}
[%c{2.}] (%t) %s%e%n
quarkus.otel.exporter.otlp.traces.endpoint=http://
localhost:4317
quarkus.otel.exporter.otlp.traces.protocol=grpc
```

El **quarkus.otel.exporter.otlp.traces.endpoint** configura el URL del Jaeger collector, el cual recolecta la tracing data de los microservicios.

- b. Verifica que la misma configuración existe para los proyectos **multiplier** y **solver**.

5. Inicia los 3 microservicios.

Revisa y ejecuta el script **start.sh** (linux o macosx) o **start.ps1** (windows) en una nueva terminal.

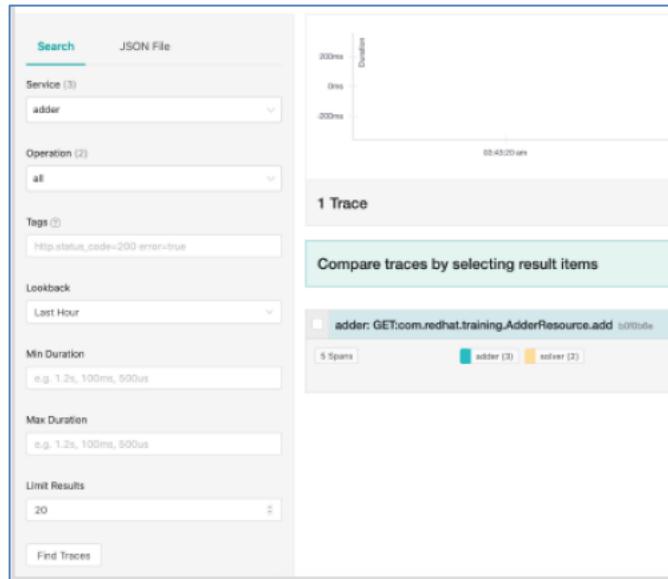
```
[student@workstation ~]$ sh ~/D0378/monitor-trace/start.sh
Starting the 'solver' project
...output omitted...
Starting the 'adder' project
...output omitted...
Starting the 'multiplier' project
...output omitted...
Press enter to Terminate
...output omitted...
```

6. Captura trazas invocando los REST endpoints expuestos por los 3 microservicios. Usa la consola web de Jaeger para visualizar las trazas y tiempo de ejecución.
 - a. En tu maquina navega a la consola web de Jaeger <http://localhost:16686> en un navegador. Aún no deberías ver ninguna traza porque no has invocado ningún endpoint en la aplicación.
 - b. Abre una nueva terminal e invoca al endpoint para el microservicio adder.

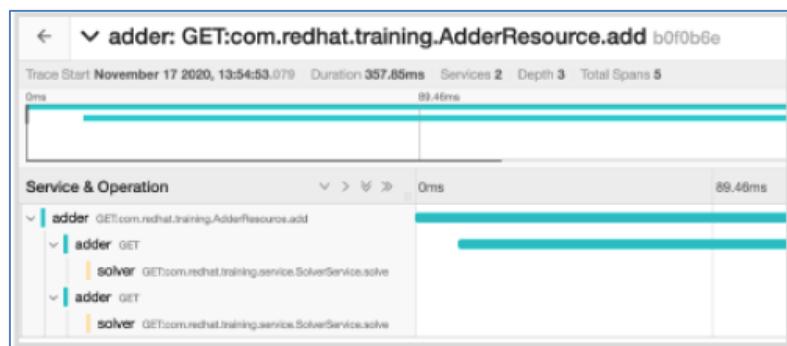
`curl http://localhost:8081/adder/5/3`

8.0

- c. Refresca la consola web de Jaeger. Selecciona el servicio **adder** desde el campo **Service** en el panel **Search** a la izquierda. Clic en **Find Traces** para ver las trazas.



- d. Clic en la traza **adder:GET:com.bcp.training.AdderResource** para ver los detalles de la traza.

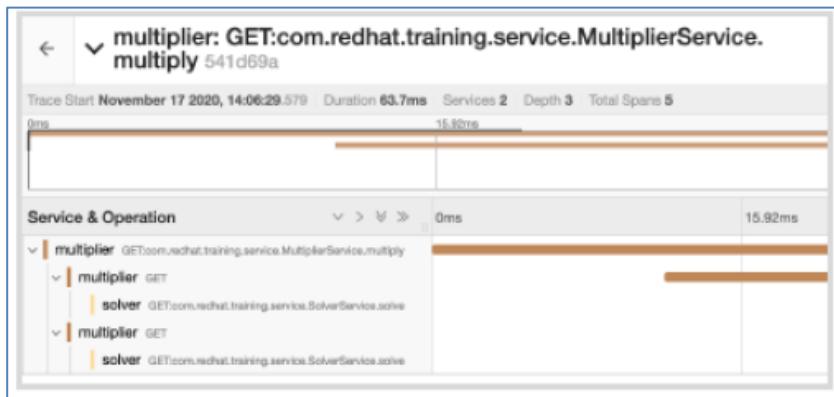


- e. Switch a la terminal donde tu ejecutaste el comando curl e invoca el endpoint para el microservicio **multiplier**.

`curl http://localhost:8082/multiplier/5/4`

20.0

- f. Encuentra la traza para esta invocación seleccionando el servicio **multiplier** desde el campo **Service** en la pagina home de Jaeger y luego clic en **Find Traces**. Clic en **multiplier:GET:com.bcp.training.MultiplierService.multiply** trace para ver el detalle de la traza. La salida debería ser algo similar a esto.

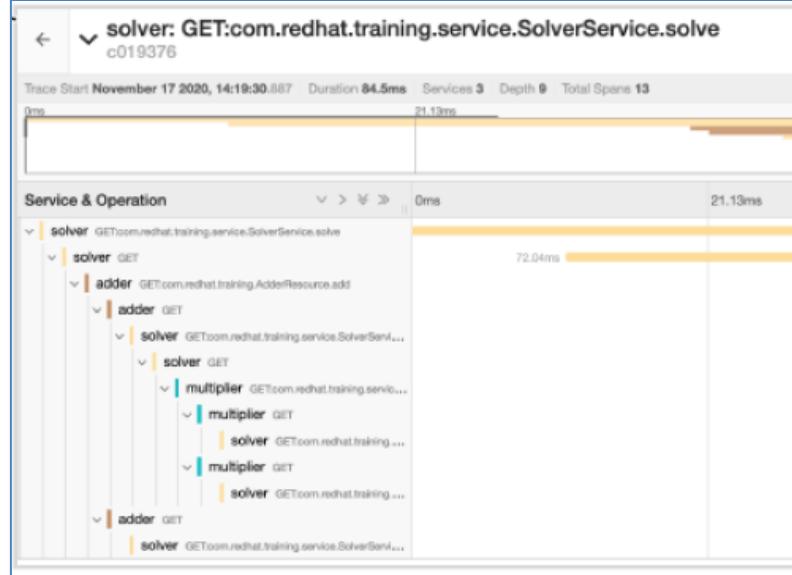


- g. Invoca el endpoint para el microservicio **solver**. Este servicio puede tomar ecuaciones compuestas con adición y multiplicación como entrada.

`curl “http://localhost:8080/solver/5*4+3”`

23.0

- h. Encuentra la traza para esta invocación seleccionando el servicio **solver** desde el campo **Service** en la pagina home de Jaeger y luego clic en **Find Traces**. Clic en **solver:GET:com.bcp.training.SolverService.solve** trace para ver el detalle de la traza. La salida debería ser algo similar a esto.



7. Limpiar todo. Para los microservicios y el contenedor de Jaeger.
 - a. Para los 3 microservicios, presiona ENTER en la ventana terminal donde tu ejecutaste el script start.sh
 - b. Detene el contenedor jaeger, presiona Ctrl+C donde tu ejecutaste el script jaeger.sh.

El laboratorio ha terminado.

Enjoy!

José