7. Adding context to published messages (JMS automatic instrumentation)

Clean-up from previous sections

If there are messages on your queue from previous sections, let's take a moment to delete them.

solbroker(admin)# message-spool message-vpn default

solbroker(admin/message-spool)# delete-messages queue q

This will delete all spooled messages in q

Do you want to continue (y/n)? y

Using solace-publisher application with context propagation and auto-instrumentation

This command will launch the solace-publisher application and publish a message as well as push additional context information to the collector. Be sure to update this argument's IP to point to your collector: -Dotel.exporter.otlp.endpoint=http://localhost:4317

Be sure to replace <absolute\_path\_to\_the\_jar\_file> with an absolute path to the jarfile ‘opentelemetry-javaagent-all.jar’ on your machine. There are 2 places in the command where this needs to be done.

[pl89@dev3-166 tracing-codelab]$

java -javaagent:<absolute\_path\_to\_the\_jar\_file>/opentelemetry-javaagent-all.jar -Dotel.javaagent.extensions=<absolute\_path\_to\_the\_jar\_file>/solace-opentelemetry-jms-integration.jar -Dotel.propagators=solace\_jms\_tracecontext -Dotel.exporter.otlp.endpoint=http://localhost:4317 -Dotel.traces.exporter=otlp -Dotel.metrics.exporter=none -Dotel.instrumentation.jms.enabled=true -Dotel.resource.attributes=“service.name=SolaceJMSPublisher” -Dsolace.host=localhost:55557 -Dsolace.vpn=default -Dsolace.user=default -Dsolace.password=default -Dsolace.topic=solace/tracing -jar solace-publisher.jar

The following command will launch solace-queue-receiver application to consume the message that was just published as well as provide additional context information directly to the collector about this message being consumed.

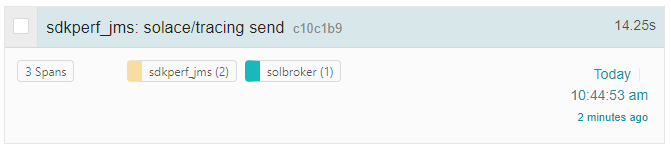
Be sure also here to replace <absolute\_path\_to\_the\_jar\_file> with an absolute path to the jarfile ‘opentelemetry-javaagent-all.jar’ on your machine. There are 2 places in the command where this needs to be done.

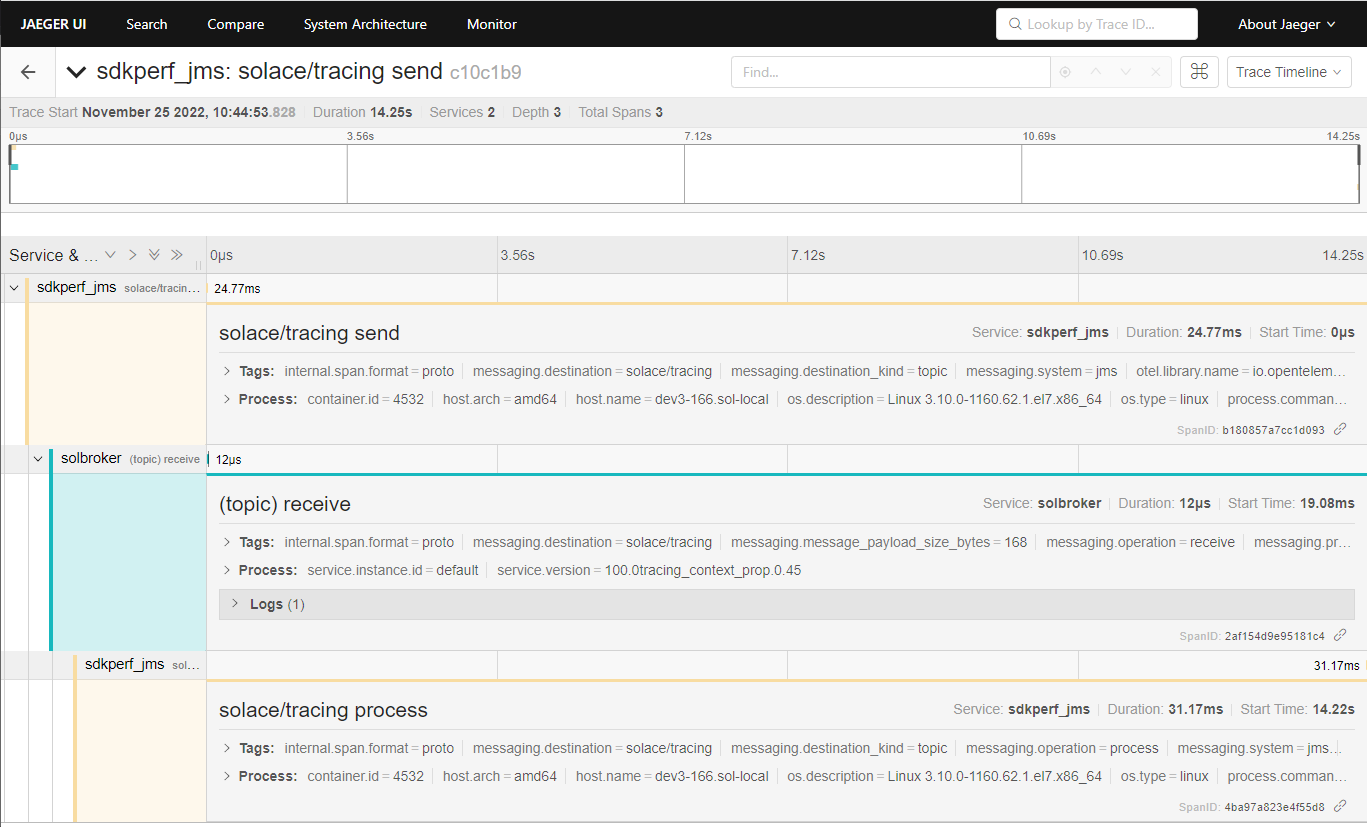
[pl89@dev3-166 tracing-codelab]$

java -javaagent:<absolute\_path\_to\_the\_jar\_file>/opentelemetry-javaagent-all.jar -Dotel.javaagent.extensions=<absolute\_path\_to\_the\_jar\_file>/solace-opentelemetry-jms-integration.jar -Dotel.propagators=solace\_jms\_tracecontext -Dotel.traces.exporter=otlp -Dotel.metrics.exporter=none -Dotel.instrumentation.jms.enabled=true -Dotel.resource.attributes="service.name=SolaceJMSQueueSubscriber" -Dsolace.host=localhost:55557 -Dsolace.vpn=default -Dsolace.user=default -Dsolace.password=default -Dsolace.queue=q -Dsolace.topic=solace/tracing -jar solace-queue-receiver.jar

When you are done testing and wish to end the solace-queue-receiver applicaiton, simply send ctrl+c from the keyboard..

Verify trace generated in Jaeger

A new trace should have been generated, notice how it has 3 spans. 

Opening up the newly generated trace will allow you to easily follow the sequence of events. 

The first span was generated by the publisher when the message was published. The second span was generated by the PubSub+ Broker when the message was received. The third span was generated by the consumer when the message was received.