1. What you'll learn: Overview

This CodeLabs will take you through the basics of the new distributed tracing feature. Following these steps will take you through:

* Launching and configuring a PubSub+ Event Broker Software
* Launching the OpenTelemetry Collector configured to use Solace modules
* Launching Jaeger which offers a user interface to view traced events
* Publishing and receiving messages to/from your broker to generate broker trace events
* Use auto-instrumented JMS application that will generate end to end linked traces (publisher – broker – receiver traces)

Upon successful completion of this Code Labs, we encourage you to experiment with distributed tracing and the environment provided to see how it fits with your use case(s). This can include other message sources, Open Telemetry exporters, and telemetry analysis tools. Please note that as a Demo feature using a standard broker edition release there are some restrictions.

Limitations and caveats

For this release, trace events will be generated for published messages (guaranteed and promoted direct) upon broker receipt and when the message is enqueued by the broker. This release does support context propagation to bind telemetry for the same message from multiple sources. ~~The areas or feature interactions to avoid include but are not limited to:~~

* ~~Direct Messaging (tracing not supported)~~
* ~~HA (High Availability) / Redundancy~~
* ~~XA Transactions~~
* ~~Replication~~
* ~~DMR (Dynamic Message Routing)~~
* ~~Appliances are not supported at this time~~

This codelabs project is provided for demonstration purposes only. The sample applications included herein (solace-publisher and solace-queue-receiver), the configuration, and the setup scripts are not intended for general use, nor do they contain necessary certificates, or configuration for a secure session connection. As such they should only be used in a local environment for feature demonstration purposes only.

Please contact your SE for support.