2. What you need: Prerequisites

Docker

This CodeLabs relies on the use of Docker If you do not already have Docker installed, you will first need to do that. Docker Desktop can be installed for ease of use. At least 4 GiB and 2 cores should be made available for Docker. If more physical resources are available, providing more may improve your experience (e.g. 8 GiB and 4 cores).

Java

This CodeLabs relies on the features found in modern Java JRE version (Open JDK or Oracle JDK when appropriate license is available by user). For this demo you must have Java 16 or higher.

To validate that Java is correctly installed on your system type following commands in your console:

[pl89@dev ~]$ java -version

If Java is correctly installed on our machine (not exactly like below…) this will be printed indicating a vendor and the version of the Java installed on your machine.

[pl89@dev ~]$ java -version

openjdk version "16" 2021-03-16

OpenJDK Runtime Environment (build 16+36-2231)

OpenJDK 64-Bit Server VM (build 16+36-2231, mixed mode, sharing)

Downloading the tracing-codelab package

The tracing-codelab package contains/require the following items:

* docker-compose.yaml (Docker images will be downloaded by Docker environment upon first launch from a Docker hub):
  + Docker image of the Solace PubSub+ Event Broker
  + Docker image of the OpenTelemetry Collector Contrib (<https://github.com/open-telemetry/opentelemetry-collector-contrib>) packaged with a Solace receiver modules
  + Docker Image for the Jaeger all in one
* otel-collector-config.yaml
* solace\_config\_keys.env
* .env (file with environment variables used in a docker compose files)
* solace-publisher.jar (command line Solace jms application for publishing of messages)
* solace-queue-receiver.jar (command line Solace jms application for receiving of messages from a JMS Queue)
* opentelemetry-javaagent-all.jar (OpenTelemetry Java Instrumentation API)
* solace-opentelemetry-jms-integration.jar (Solace PubSub+ OpenTelemetry Integration API for JMS) [LINK]

You will be able to download the following package from the Solace product download site: XXADD\_ LINK\_HEREXXX

When extracting from this archive, it is mandatory that there are no SPACES in the full path to the working directory.

[pl89@dev ~]$ tar -xf tracing-codelab.tar.gz

[pl89@dev ~]$ cd tracing-codelab

Creating and launching the containers

The following command will download and launch all containers necessary for the codelab (Internet access will be required to download images from Docker hub).

[pl89@dev tracing-ea]$ docker compose up -d

[+] Running 6/6

⠿ jaeger-all-in-one Pulled 3.2s

⠿ 8663204ce13b Pull complete 0.5s

⠿ b86734d97f6d Pull complete 0.6s

⠿ 0d263244379d Pull complete 0.6s

⠿ e57f20e53339 Pull complete 2.5s

⠿ f4969b810177 Pull complete 2.5s

[+] Running 4/4

⠿ Network tracing-ea\_default Created 0.0s

⠿ Container tracing-ea-jaeger-all-in-one-1 Started 0.6s

⠿ Container tracing-ea-solbroker-1 Started 0.7s

⠿ Container tracing-ea-otel-collector-1 Started 1.6s

NOTE: Be aware the command demonstrated above is `docker compose` and not docker-compose. Make sure to use a recent version. Both docker compose and docker-compose may be available on your system and could differ in version.

Few notes to the code lab configuration

The *.env* file contains several environment variables that are used within the *docker-compose.yaml* file and may need to be changed by user depends on the runtime environment:

* Solace Pub Sub Plus broker port PUBSUB\_PORT\_NUMBER=55557
* Open Telemetry contribution repository collector docker image tag and version **otel/opentelemetry-collector-contrib:0.67.0**
* Solace PubSub+ broker docker image tag and version **XXX**