Jaeger - Getting Started

1. All in one Docker image

This image, designed for quick local testing, launches the Jaeger UI, collector, query, and agent, with an in memory storage component.

The simplest way to start the all in one docker image is to use the pre-built image published to DockerHub (a single command line).

```
docker run -d -e COLLECTOR_ZIPKIN_HTTP_PORT=9411 -p5775:5775/udp -p6831:6831/udp
-p6832:6832/udp \
  -p5778:5778 -p16686:16686 -p14268:14268 -p9411:9411 jaegertracing/all-in-one:latest
```

You can then navigate to `http://localhost:16686` to access the Jaeger UI.

The container exposes the following ports:

Port	Protocol	Component	Function
5775	UDP	agent	accept zipkin.thrift over compact thrift protocol
6831	UDP	agent	accept jaeger.thrift over compact thrift protocol
6832	UDP	agent	accept jaeger.thrift over binary thrift protocol
5778	HTTP	agent	serve configs
16686	HTTP	query	serve frontend
9411	HTTP	collector	Zipkin compatible endpoint

2. Kubernetes and OpenShift

Kubernetes and OpenShift templates can be found in the Jaegertracing organization on Github.

3. Sample Application

3.1. HotROD (Rides on Demand)

This is a demo application that consists of several microservices and illustrates the use of the OpenTracing API. A tutorial / walkthough is available in the blog post: Take OpenTracing for a HotROD ride.

It can be run standalone, but requires Jaeger backend to view the traces.

3.1.1. Running

```
mkdir -p $GOPATH/src/github.com/jaegertracing
cd $GOPATH/src/github.com/jaegertracing
git clone git@github.com:jaegertracing/jaeger.git jaeger
cd jaeger
make install
cd examples/hotrod
go run ./main.go all
```

Then navigate to `http://localhost:8080`.

3.1.2. Features

- Discover architecture of the whole system via data-driven dependency diagram.
- View request timeline and errors; understand how the app works.
- Find sources of latency and lack of concurrency.
- Highly contextualized logging.
- Use baggage propagation to:
- Diagnose inter-request contention (queueing).
- Attribute time spent in a service.
- Use open source libraries with OpenTracing integration to get vendor-neutral instrumentation for free.

3.1.3. Prerequisites

- You need Go 1.9 or higher installed on your machine.
- Requires a [running Jaeger backend](#all-in-one-docker-image) to view the traces.

4. Client Libraries

Look here.

5. Running Individual Jaeger Components

Individual Jaeger backend components can be run from source. They all have their `main.go` in the `cmd` folder. For example, to run the `jaeger-agent`:

```
mkdir -p $GOPATH/src/github.com/jaegertracing
cd $GOPATH/src/github.com/jaegertracing
git clone git@github.com:jaegertracing/jaeger.git jaeger
cd jaeger
make install
go run ./cmd/agent/main.go
```

6. Migrating from Zipkin

Collector service exposes Zipkin compatible REST API `/api/v1/spans` and `/api/v2/spans` for both JSON and thrift encoding. By default it’s disabled. It can be enabled with `--collector.zipkin.http-port=9411`.

Zipkin Thrift IDL file can be found in jaegertracing/jaeger-idl. It’s compatible with openzipkin/zipkin-api