HACKING OPENTELEMETRY

Alberto Gastaldello, Andrea Caretta - Liquid Reply IT

Conf42, June 8th 2023



AGENDA

- 1 Intro Observability tips
- 2 OpenTelemetry
- 3 Distributed Traces
- 4 Hacking OpenTelemetry
- 5 Demo

OBSERVABILITY PILLARS



OPEN TELEMETRY

ADOPTERS AND CONTRIBUTORS

Opentelemetry Enables gaining deep insights into software's performance and behavior. **CLOUD NATIVE** COMPUTING FOUNDATION Suite of tools, APIs and SDKs designed to: instrument applications generate valuable telemetry data export them to different backends

Cloud Providers

Recognize and adopt the standards issued by the CNCF

Enterprise Software

Use the Opentelemetry specification in their software to convey telemetry data

Observability Platforms

Leverage open source and support data ingestion in different formats

COMPONENTS

Auto-instrumentation agents

Packages capable of generating telemetry data without changing the application code; many libraries are supported and recognized automatically

K8s Operator

Kubernetes Operator implementation, manages Otel Collector and self-instrumentation of workloads in a distributed environment

OpenTelemetry Collector

Proxy capable of receiving telemetry data in different formats, editing and/or filtering it, and sending it in different formats to the desired backend

Language-specific SDK

They allow you to use APIs to generate telemetry data, export it to different backends propagating context information

Cross-language references

API (data types and operations), Data (semantic conventions)

DATA FLOW



DISTRIBUTED TRACES

TRACING CONCEPTS

TRACE CONTEXT **SPAN** Defines a universally recognized A span is identified as a trace unit. A "distributed trace", or simply format for propagating data and "trace", records the paths linking them together. Can be followed by a request (triggered As a building block of a distributed adopted by different platforms and by an application or an end-user) trace, it traces specific different tracking tools. propagating through multi-service operations that a request makes, architectures. reporting a drawing of what The following headers are happened during the interval in propagated: A trace consists of one or more which the operation was

performed.

spans.

- traceparent
- tracestate

TRACING CONCEPTS





A COMMUNICATING SCENARIO





FOLLOWING A TRANSACTION



TRACKING RESULTS - DYNATRACE

					؛ III	Split 📴 Add to dashboard	④ Last 1 hour ∨ ⊖	😋 Run query 🗸		
A (Tempo) Query type Search Build complex queries using	Instributed traces soft40109166789099e31645791873378								Find in remote e	nvironments
Schedubreb.sopped.stoa Options Limit: 20 + Add query S Que Node graph	182ms Response time Attributes 1 Failure nethostname: orders nettransport: ip 1 Exception netsockpeeraddr: 10	tep http:target: *****								
sock-shop:orders: POS Trace Shop:orders: POS Service & Operation sock-shop:orders on sock-shop:orders on sock-shop:orders sock-shop	Search name, URL, SQL, attribute	0 ms 20 ms	40 ms	60 ms	80 ms	too ms	120 ms	140 ms	140 ms	120 ms
sock-shopporders sock-shopporders sock-shopporders sock-shopporders sock-shopporders sock-shopporders sock-shopporders sock-shopporders	GET GET GET GET GET GET Get-indparters GET Get-indparters GET Get-indparters DOST Get-indparters Cost-indparters				-	7.				

TRACKING RESULTS - GRAFANA

Distributed traces	56f40109fe6789b9ee3f6a579f873378								
= 'POST	/orders' Trace				١	Find in remote environments			
Start time: 3	2023 March 8 12:52:58 What's new?				l				
187ms Pernons						Split 🔡 Add to dashboa	rd 🕐 Last 1 hour 👻 🔍	🔓 Run query 🗸	
1 Failure	~ A (Tempo)							0001:	
1 Exception	Query type Search TraceQL JSON File Service Graph								
	Build complex queries using TraceQL to select a list of traces.							Documentation	
56f48109f6c78959e29f63579f373378									
Search hame, o	2 OF								
sock-shop									
Sock-sho	 + Add query Query history Inspector 								
V 🔊 GET sock-s	h > Node graph								
V 🔊 GET sock									
V 🔊 Ite	en G		Irace view					* *	
✓ , e th	sock-shop:orders: POST /orders 561401091e6789b9ee316a5791873378								
,	Trace Start: 2023-03-08 12:52:58.024 Duration: 181:53ms Services: 4 Depth: 8 Total Spans: 21								
		45.38ms	90.77ms			136.15ms		181.53ms	
GET									
Sock-s	Service & Operation	0µs	45.38ms	90	0.77ms	136.15ms		181.53ms	
sock-s	h sock-shop:orders POST /orders (181.53ms)								
sock-s	h sock-shop:orders OrdersController.newOrder (174.74ms)							174.7	
POST	 sock-shop:orders GET (17.94ms) 	17.94ms							
POST	 sock-shop:carts GET /carts/(customerid:.*)/items (10.03ms) 	10.03ms							
sock-s	Sock-snop:carts ItemsController.getItems (7.35ms)	7.35ms							
V 💉 POS sock	Sock-Shop:carts CartRepository.lindByCustomerid (5.73ms)	5.73ms							
🗸 🔊 Sh	sock-shop carts find data.cart (1.22ms)	- 1.22ms							
50	cock-shop orders. (CET (9.41me)	0.44							
 , a 		8.4 IIIIs							
	sock-shop orders GET (9.47me)	9.47mm							
· ·	sock-shop orders POST (3.5mc)	0.4112		2 5mr					
× ,,,**	sock-shop orders POST (24.28ms)			24.28ms					
	sock-shop:shipping POST (shipping (12.41ms)			12 41					
	cock-shop-shipping (Contraliar aget Shipping (5.49me)				5.40mc				
	sock-shop-shipping-chipping-controller-poston(pping (3.49ms)				706 6200				
	cock-chop-guigue-macter shipsing-tack process (£2.00us)				FD 00.02.05				
	sock-shop:queue-master shipping-task process (32.0eps)				1.88mr				
	sock-shop:queue-master POST (13.63mc)				13.63ms			†	
	sock-shop:orders CustomerOrderRepository.save (10.28ms)				10.28ms				

HACKING THE COLLECTOR

ONLY LOGS AVAILABLE?



COLLECTOR MODULES & PIPELINES



COLLECTOR CONFIGURATION



FROM LOG TO TRACE



EVERYTHING CAN BE A SPAN



TELEMETRY DATA STORAGE



DEMO

CONTACTS

Alberto Gastaldello

al.gastaldello@reply.it +39 342 864 7689

Andrea Caretta

a.caretta@reply.it +39 345 970 7769



