Tracing

Document Level Classification

200

Introduction

We currently have tracing support in two forms:

- **trace span metrics**, generated by OTEL collectors and query-able from Grafana via Chronosphere datasources.
- **raw traces in Tempo**, query-able from Grafana in Tempo datasources.

Sampling

We default to 100% sampling in dev environments and 1% sampling in production environments. Trace metrics are calculated based on 100% of the data, so the counts you see there are not sampled.

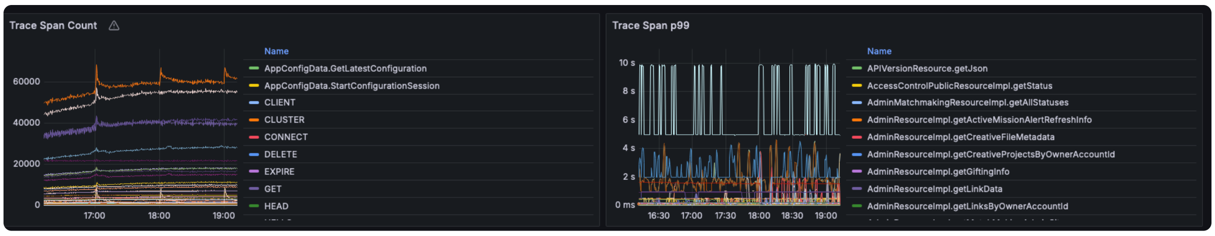
Sampling production workloads by default is necessary due to the potential for large volumes of data that some production systems generate, which can be very costly. If your production substrate cluster sends < 1000 spans/sec (we can help you find out) and you would like to sample 100% of traces, reach out to #ct-obs-support-ext and we can make an exception for your cluster.

Trace Metrics

All spans received by the OTEL collector running in your substrate collector are aggregated and turned into a set of metrics:

```
traces_span_metrics_calls
traces_span_metrics_duration_milliseconds_total
traces_span_metrics_duration_milliseconds_sum
traces_span_metrics_duration_milliseconds_bucket
```

These can be used to understand behavior within an application and determine things like client call latency. They're also a way to get metrics out of libraries that only provide instrumentation via spans. For example:

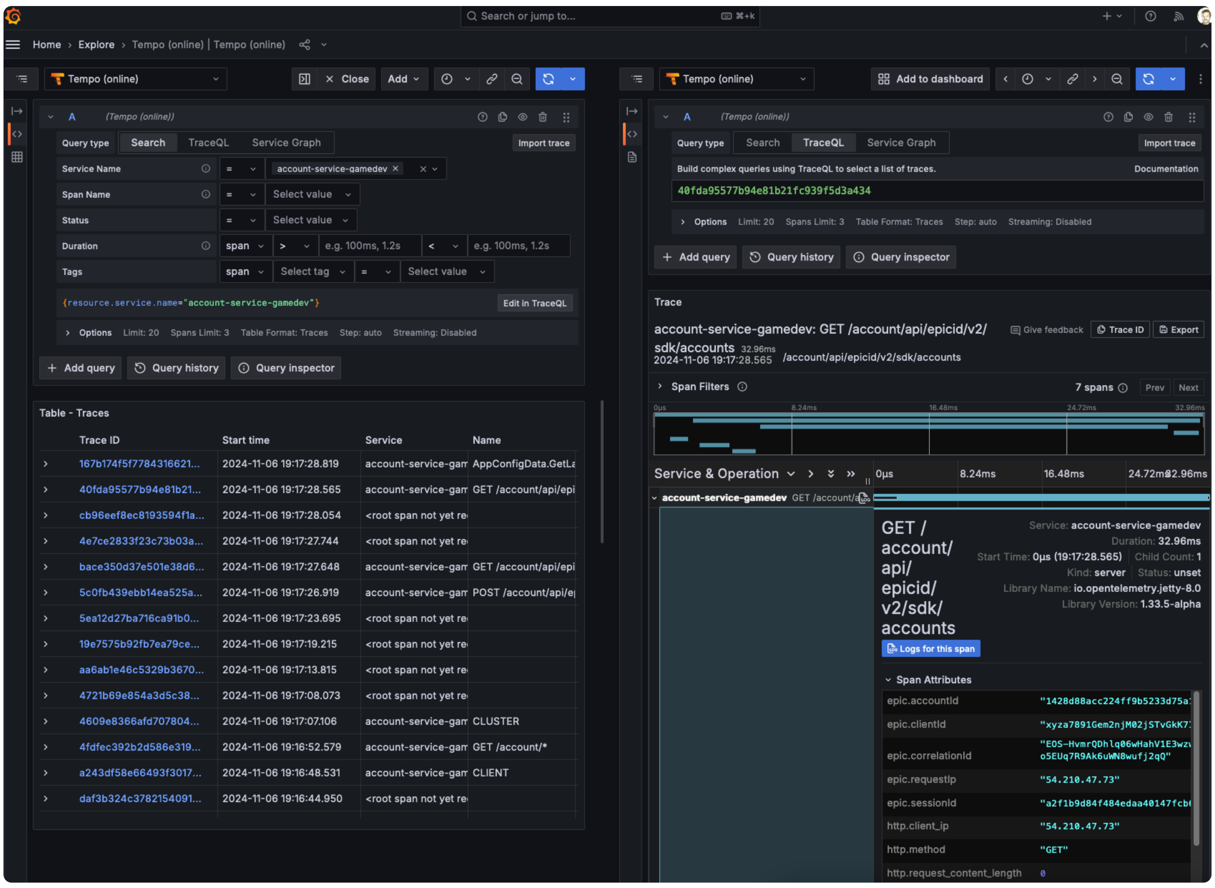


```
sum(rate(calls{service_name="$service_name"}[1m])) by (span_name, span_

histogram_quantile(0.99, sum by (le, span_name) (rate(duration_millisec
```

Tempo

Tempo can be used to view raw traces and see service topology. You can access Tempo via Grafana datasources in the Explore UI. For more information on how to query trace data, see the [TraceQL documentation.]



Query type: Search | TraceQL | Service Graph

Service Name = account-service-gamedev
Span Name = Select value
Status = Select value
Duration span > e.g. 100ms, 1.2s < e.g. 100ms, 1.2s
Tags span Select tag = Select value

```
{resource.service.name="account-service-gamedev"}
```

Options Limit: 20 Spans Limit: 3 Table Format: Traces Step: auto Streaming: Disabled

Table - Traces

Trace ID	Start time	Service	Name
167b174f5f7784316621...	2024-11-06 19:17:28.819	account-service-gam	AppConfigData.GetLa
40fda95577b94e81b21...	2024-11-06 19:17:28.565	account-service-gam	GET /account/api/epi
cb96eef8ec8193594f1a...	2024-11-06 19:17:28.054	<root span not yet re	
4e7ce2833f23c73b03a...	2024-11-06 19:17:27.744	<root span not yet re	
bace350d37e501e38d6...	2024-11-06 19:17:27.648	account-service-gam	GET /account/api/epi
5c0fb439ebb14ea525a...	2024-11-06 19:17:26.919	account-service-gam	POST /account/api/e
5ea12d27ba716ca91b0...	2024-11-06 19:17:23.695	<root span not yet re	
19e7575b92fb7ea79ce...	2024-11-06 19:17:19.215	<root span not yet re	
aa6ab1e46c5329b3670...	2024-11-06 19:17:13.815	<root span not yet re	
4721b69e854a3d5c38...	2024-11-06 19:17:08.073	<root span not yet re	
4609e8366afd707804...	2024-11-06 19:17:07.106	account-service-gam	CLUSTER
4fdfec392b2d586e319...	2024-11-06 19:16:52.579	account-service-gam	GET /account/*
a243df58e66493f3017...	2024-11-06 19:16:48.531	account-service-gam	CLIENT
daf3b324c3782154091...	2024-11-06 19:16:44.950	<root span not yet re	

Query type: Search | TraceQL | Service Graph

Build complex queries using TraceQL to select a list of traces.

```
40fda95577b94e81b21fc939f5d3a434
```

Options Limit: 20 Spans Limit: 3 Table Format: Traces Step: auto Streaming: Disabled

Trace

account-service-gamedev: GET /account/api/epicid/v2/sdk/accounts 32.96ms
2024-11-06 19:17:28.565 /account/api/epicid/v2/sdk/accounts

Span Filters 7 spans

0µs 8.24ms 16.48ms 24.72ms 32.96ms

Service & Operation 0µs 8.24ms 16.48ms 24.72ms 32.96ms

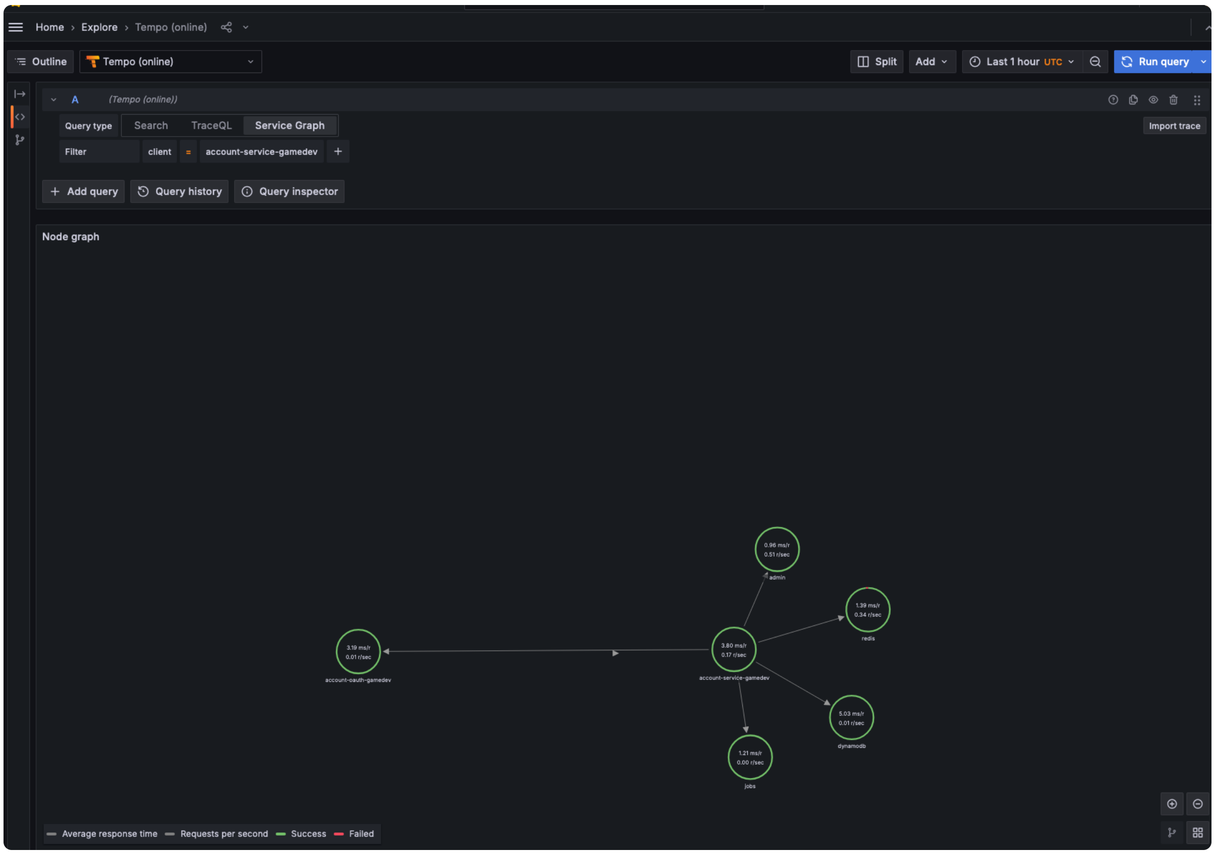
account-service-gamedev GET /account/a

GET /account/api/epicid/v2/sdk/accounts

Service: account-service-gamedev
Duration: 32.96ms
Start Time: 0µs (19:17:28.565) | Child Count: 1
Kind: server | Status: unset
Library Name: io.opentelemetry.jetty-8.0
Library Version: 1.33.5-alpha

Span Attributes

epic.accountId	"1428d88acc224ff9b5233d75a
epic.clientId	"xyza7891Gem2njM02jSTvGkK7
epic.correlationId	"EOS-HvmrQDhlq06wHahV1E3wz o5EUq7R9Ak6uWN8wufj2qQ"
epic.requestIp	"54.210.47.73"
epic.sessionId	"a2f1b9d84f484edaa40147fcb
http.client_ip	"54.210.47.73"
http.method	"GET"
http.request_content_length	0



Query type: Search | TraceQL | Service Graph

Filter client = account-service-gamedev

Sending Traces

In order for your traces to route properly in the environment, we require the following attributes and header to be set. Without these, your data will be ignored and not received. It's important to note we sample traces using probabilistic sampling processor. If your environment is `prod || live` , your traces will be sampled at 1%. Any other environment definition will be sampled at 100% meaning all data is received.

```
# Attributes
service.name: "foo"
service.instance.id: "bar"
"epic.environment": "dev / prod / live" # see sampling percentages in a

# Headers
"X-Epic-Tenant-ID": "foobar" # this must match your Tempo Tenant, eg. "
```

Note: all trace collectors use **http/protobuf**

Substrate

If you are using OTEL auto-instrumentation, you should already see a significant amount of trace data for your service. If you can augment this with custom trace spans using an OTEL SDK or other library as long as it exports data in OTLP format. Use the collector address below to send trace data. If using environment variables to configure your SDK, you can set this using OTEL_EXPORTER_OTLP_TRACES_ENDPOINT.

```
http://trace-collector.observability.svc.cluster.local:4318/v1/traces
```

Internal + External Collectors

We have internal + external collectors to receive traces for services running outside of substrate. If you can reach the service network, use the **Internal Trace Collector**. If you're outside of Substrate and cannot reach the service network, use the **External Trace Collector**. Credentials are stored in Vault

```
# Internal Collector
https://internal-traces-collector.observability.internal.epicgames.net/

# External Collector
https://external-traces-collector.fbfb.live.use1a.on.epicgames.com/v1/t
```

Do's & Don'ts

- **DO** use span attributes to store high cardinality data (like URL, correlation ID, query parameters, etc)
- **DON'T** put high cardinality data in Span names. This breaks both Tempo indexing and Trace Metrics. See the OTEL Trace API specification for more details on creating good span names
- **DON'T** send prod data to dev processor - if you believe you're missing valuable data due to sampling reach out to the observability team

Page Information:
Page ID: 81068444
Space: Cloud Developer Platform
Downloaded: 2025-07-12 04:08:58