

Tracing



- [Configuration Example](#)
- [Configuration Options](#)
- [sampleRate](#)
- [resourceAttributes](#)

Visualize the Requests Flow {:.subtitle }

The tracing system allows developers to visualize call flows in their infrastructure.

Traefik uses [OpenTelemetry](#), an open standard designed for distributed tracing.

Configuration Example

To enable the tracing:

`tracing: {}`

`[tracing]`

`--tracing=true`

`tracing:
 otlp:
 enabled: true`

Configuration Options

Field	Description	Default	Required
-------	-------------	---------	----------

<code>tracing.addInternals</code>	Enables tracing for internal resources (e.g.: <code>ping@internal</code>).	false	No
<code>tracing.serviceName</code>	Defines the service name resource attribute.	"traefik"	No
<code>tracing.resourceAttributes</code>	Defines additional resource attributes to be sent to the collector. See resourceAttributes for details.	{}	No
<code>tracing.sampleRate</code>	The proportion of requests to trace, specified between 0.0 and 1.0. Since Traefik supports parent-based sampling ratios, root spans (i.e., spans initiated by Traefik) are sampled according to this rate, while child spans inherit the sampling decision of their parent (i.e., the tracing context from incoming requests). See sampleRate for details.	1.0	No
<code>tracing.capturedRequestHeaders</code>	Defines the list of request headers to add as attributes. It applies to client and server kind spans.	[]	No
<code>tracing.capturedResponseHeaders</code>	Defines the list of response headers to add as attributes. It applies to client and server kind spans.	[]	False
<code>tracing.safeQueryParams</code>	By default, all query parameters are redacted. Defines the list of query parameters to not redact.	[]	No
<code>tracing.otlp.http</code>	This instructs the exporter to send the tracing to the OpenTelemetry Collector using HTTP. Setting the sub-options with their default values.	null/false	No
<code>tracing.otlp.http.endpoint</code>	URL of the OpenTelemetry Collector to send tracing to. Format=" <code><scheme>://<host>:<port><path></code> "	"https://localhost:4318"	Yes

<code>tracing.otlp.http.headers</code>	Additional headers sent with tracing by the exporter to the OpenTelemetry Collector.		No
<code>tracing.otlp.http.tls.ca</code>	Path to the certificate authority used for the secure connection to the OpenTelemetry Collector, it defaults to the system bundle.	""	No
<code>tracing.otlp.http.tls.cert</code>	Path to the public certificate used for the secure connection to the OpenTelemetry Collector. When using this option, setting the <code>key</code> option is required.	""	No
<code>tracing.otlp.http.tls.key</code>	Defines the path to the private key used for the TLS connection.	""	No
<code>tracing.otlp.http.tls.insecureSkipverify</code>	If <code>insecureSkipVerify</code> is <code>true</code> , the TLS connection to the OpenTelemetry Collector accepts any certificate presented by the server regardless of the hostnames it covers.	false	Yes
<code>tracing.otlp.grpc</code>	This instructs the exporter to send tracing to the OpenTelemetry Collector using gRPC.	false	No
<code>tracing.otlp.grpc.endpoint</code>	Address of the OpenTelemetry Collector to send tracing to. Format= " <code><host>:<port></code> "	"localhost:4317"	Yes
<code>tracing.otlp.grpc.headers</code>	Additional headers sent with tracing by the exporter to the OpenTelemetry Collector.	[]	No
<code>tracing.otlp.grpc.insecure</code>	Allows exporter to send tracing to the OpenTelemetry Collector without using a secured protocol.	false	Yes

<code>tracing.otlp.grpc.tls.ca</code>	Path to the certificate authority used for the secure connection to the OpenTelemetry Collector, it defaults to the system bundle.	""	No
<code>tracing.otlp.grpc.tls.cert</code>	Path to the public certificate used for the secure connection to the OpenTelemetry Collector. When using this option, setting the <code>key</code> option is required.	""	No
<code>tracing.otlp.grpc.tls.key</code>	Defines the path to the private key used for the TLS connection.	""	No
<code>tracing.otlp.grpc.tls.insecureSkipverify</code>	If <code>insecureSkipVerify</code> is <code>true</code> , the TLS connection to the OpenTelemetry Collector accepts any certificate presented by the server regardless of the hostnames it covers.	false	Yes

sampleRate

The `sampleRate` option controls trace sampling using a `ParentBased(TraceIDRatioBased)` strategy.

!!! info "Sampling Strategy Behavior"

- **Root spans** (trace originating at Traefik): Sampled according to the configured `sampleRate` using trace ID ratio-based sampling.
- **Child spans** (requests with existing trace context): Inherit the sampling decision from the parent span, regardless of the local `sampleRate`.

This ensures consistent sampling decisions across distributed traces: once a trace is sampled, all spans in that trace are sampled, providing complete end-to-end visibility.

resourceAttributes

The `resourceAttributes` option allows setting the resource attributes sent along the traces. Traefik also supports the `OTEL_RESOURCE_ATTRIBUTES` env variable to set up the resource attributes.

!!! info "Kubernetes Resource Attributes Detection"

Additionally, Traefik automatically discovers the following [Kubernetes resource attributes](<https://opentelemetry.io/docs/specs/semconv/non-normative/k8s-attributes/>) when running in a Kubernetes cluster:

- `k8s.namespace.name`
- `k8s.pod.uid`
- `k8s.pod.name`

Note that this automatic detection can fail, like if the Traefik pod is running in host network mode.

In this case, you should provide the attributes with the option or the env variable.

❏ #2

❏ 2026-10-31 14:00:36 UTC

❏ 2026-10-31 14:00:37 UTC