

How Distributed Tracing Works

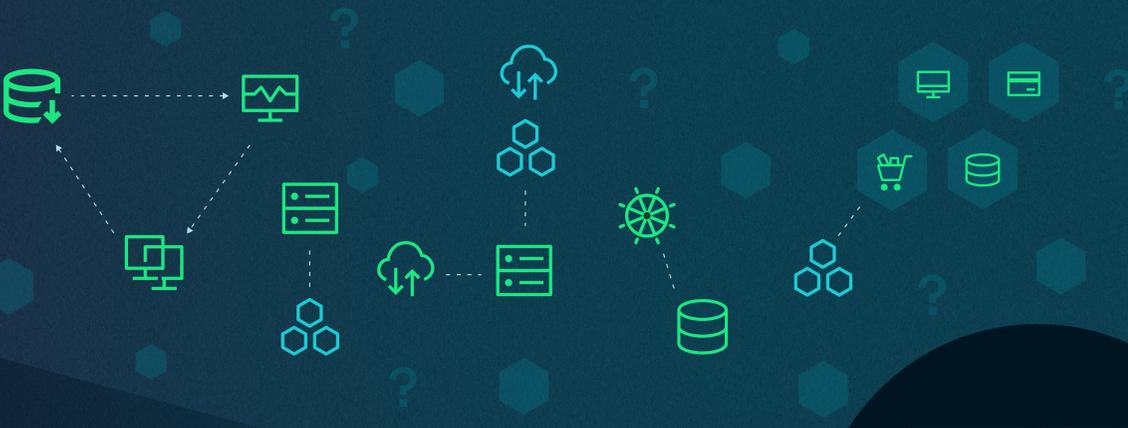
Understanding modern digital environments

New technologies introduce greater complexity for monitoring software and systems.

CONTINUOUS DEPLOYMENT

DISTRIBUTED SYSTEMS

THOUSANDS OF MICROSERVICES



What is distributed tracing?

Distributed tracing is a simple way to understand complex environments. It gives software teams an easy way to track, visualize, and collect data about requests as they travel through distributed systems.

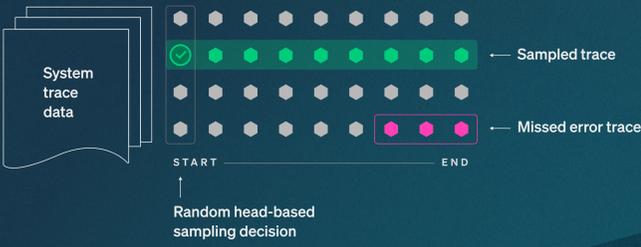


What is a trace?
Services in a distributed system talk to one another by sending requests. A trace is data that tracks the complete path of a request as it travels from service to service. It's composed of spans that represent time spent in each operation, or segment, along the path.

Types of sampling

Distributed tracing processes massive amounts of data. There are two common types of sampling to manage your data volume effectively with the insights you need.

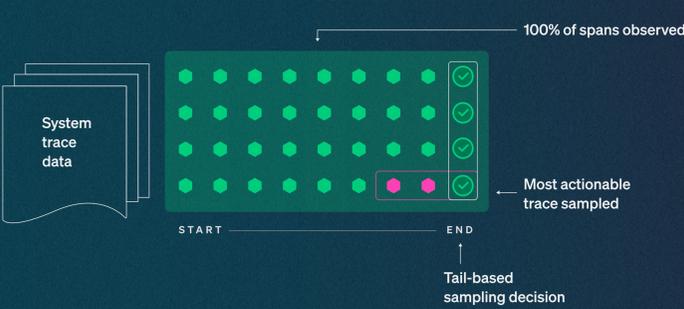
Head-based sampling



- Samples traces randomly before they have completed
- Provides sufficient visibility for non-critical applications, but can miss traces with errors or high latency
- Works well where costs are a concern or in a blend of monolithic and microservices environments

Tail-based sampling

- Samples only after the trace has fully completed the path, observes 100% of traces, and keeps those with errors, high latency, or anomalies
- Usually requires a complex tracing infrastructure
- Works well in highly distributed, cloud-based environments or critical applications



Debugging faster with New Relic

Seeing software requests from end-to-end makes finding and fixing issues faster and easier. New Relic simplifies distributed tracing every step of the way.



Instrument your stack.

Auto-instrument your services using New Relic agents, or use open instrumentation like the W3C trace context standard.



Automatically collect trace telemetry.

Gather tracing data from any source, including microservices, containers, serverless functions, messaging queues, service meshes, and more.



Visualize, analyze, and optimize.

Get end-to-end visibility and lightning-fast search to detect anomalies, reduce latency, squash errors, and optimize the customer experience.



[View the eBook](#)