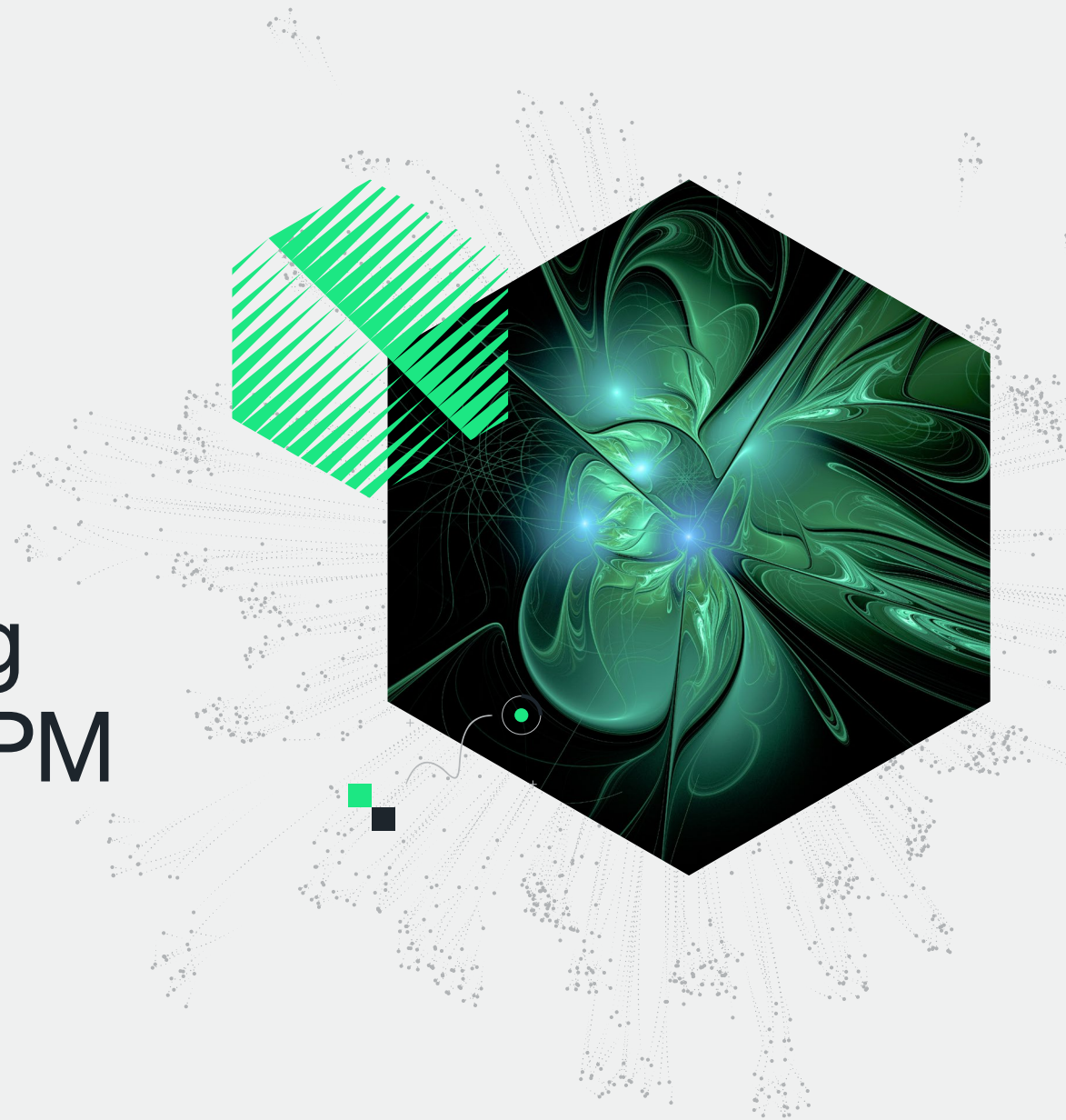


Por qué Distributed Tracing es esencial para APM

Reducción del MTTR en entornos complejos
de aplicaciones distribuidas



Contenido



03 Reducir las complejidades

04 Trazar una ruta a través de los sistemas distribuidos

- 06 › Cuando utilizar trazas
- 07 › Cómo funcionan las trazas
- 08 › Sacar conclusiones
- 09 › Por qué las organizaciones necesitan rastreo distribuido

10 Obtener visibilidad de la canalización de datos

- 10 › Muestreo eficiente al inicio
- 11 › Trazas accionables con muestreo al final
- 13 › Análisis y visualización

14 Enfrentarse a la carga de administración

15 ¿Muestreo al inicio o al final?
Cualquiera de los dos.

16 Próximos pasos

17 Acerca de New Relic

Reducir las complejidades

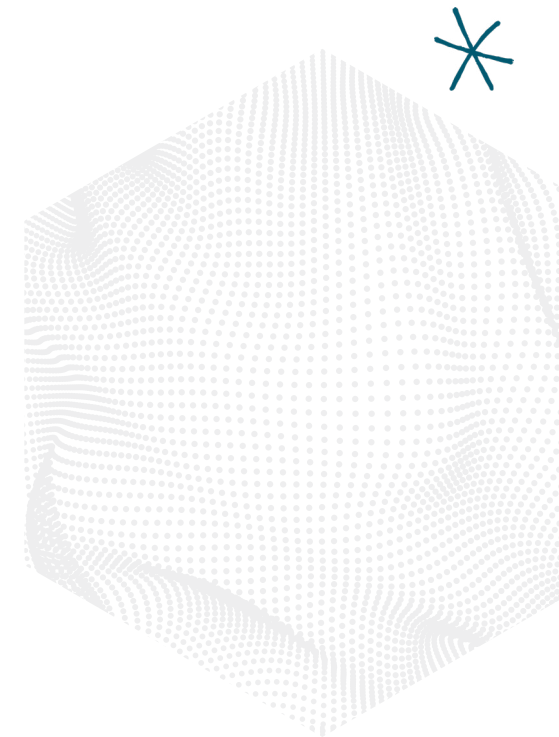
Las arquitecturas y los entornos de software modernos como los microservicios tienen el potencial de acelerar el desarrollo de aplicaciones. Pero en muchas organizaciones, los equipos de ingeniería de software se enfrentan a un entorno complejo, lo que dificulta el diagnóstico y la resolución de los problemas de rendimiento y de los errores antes de que afecten la fiabilidad y la experiencia de los clientes.

Los entornos de microservicios pueden incluir de decenas a cientos de servicios, de manera que resulta difícil determinar las rutas de las solicitudes y diagnosticar los problemas. Y la carga del monitoreo del rendimiento de aplicaciones (APM) solo aumenta con la orquestación, la automatización y la CI/CD para los despliegues de software frecuentes. Sin una instrumentación de monitoreo adecuada, las organizaciones corren el riesgo de que sus equipos tengan que buscar respuestas repetidamente en sistemas distribuidos, lo que se traduce en un incremento del tiempo medio de resolución (MTTR) y una reducción del tiempo de desarrollo de software innovador.

La observabilidad reduce la complejidad del software y proporciona visibilidad de extremo a extremo que permite que los equipos resuelvan los problemas más rápido, trabajen de forma más inteligente y creen mejores experiencias digitales para sus clientes. La observabilidad crea contenido e información práctica al, entre otras cosas, combinar cuatro tipos esenciales de datos de observabilidad: métricas, eventos, logs y trazas (MELT).

Las trazas —más concretamente, las trazas distribuidas (Distributed Tracing)— son esenciales para los equipos de software que han hecho la transición a la nube (o están pensando hacerla) y han adoptado arquitecturas de microservicios. Esto se debe a que el rastreo distribuido es la mejor manera de comprender rápidamente qué sucede con las solicitudes a medida que pasan a través de los microservicios que conforman las aplicaciones distribuidas.

Los líderes empresariales, ingenieros de DevOps, líderes de productos, ingenieros de fiabilidad del sitio (SRE), líderes de equipos de software u otros interesados pueden usar el rastreo distribuido para encontrar los cuellos de botella o errores y ganar ventaja con una resolución de problemas más rápida.



Trazar una ruta a través de los sistemas distribuidos

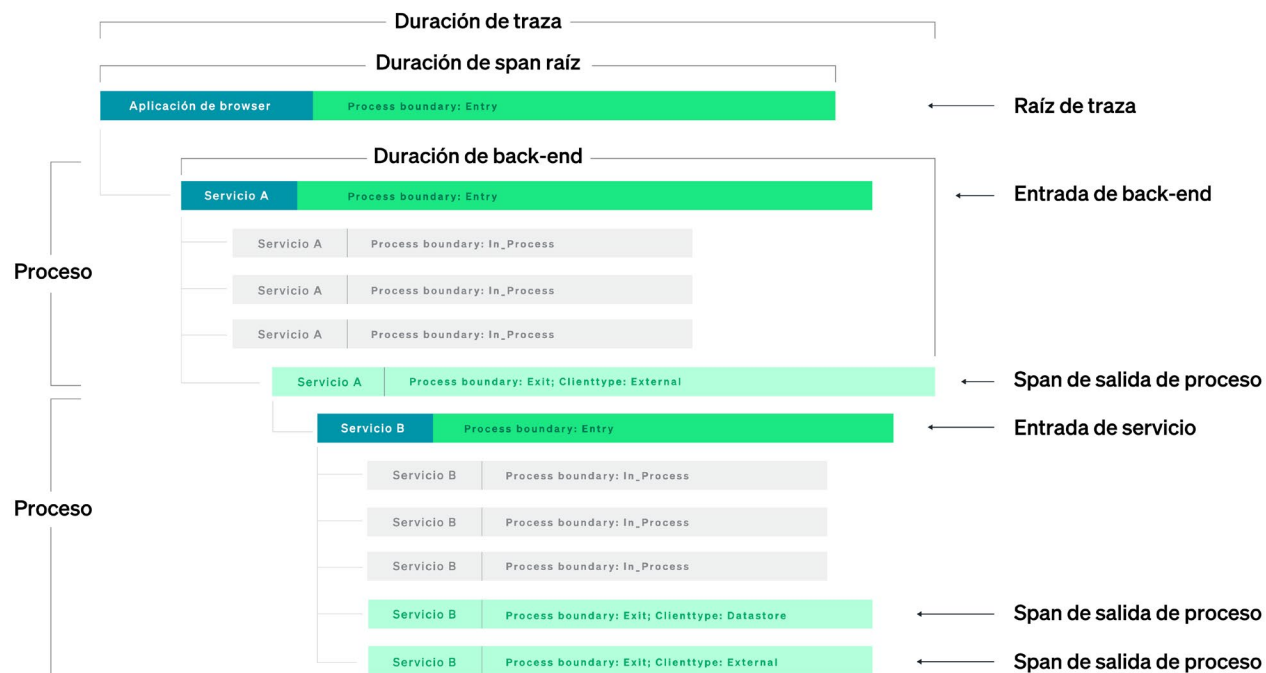
El rastreo distribuido es ahora un elemento fundamental para la operación y el monitoreo de los entornos de aplicaciones modernos. Cuando los equipos monitorean el rendimiento del sistema y del software para la observabilidad, el rastreo es una manera de monitorear y analizar las solicitudes a medida que se propagan por un entorno distribuido y pasan de un servicio a otro.

El rastreo distribuido es la capacidad de rastrear una solución para observar y hacer seguimiento de las solicitudes de servicio que fluyen a través de los sistemas, y lo hace recopilando datos a medida que dichas solicitudes van de un servicio a otro. Los datos de traza ayudan a los equipos a comprender el flujo de las solicitudes a través del entorno de microservicios y a identificar con precisión dónde ocurren las fallas o los problemas de rendimiento en el sistema, y por qué ocurren.

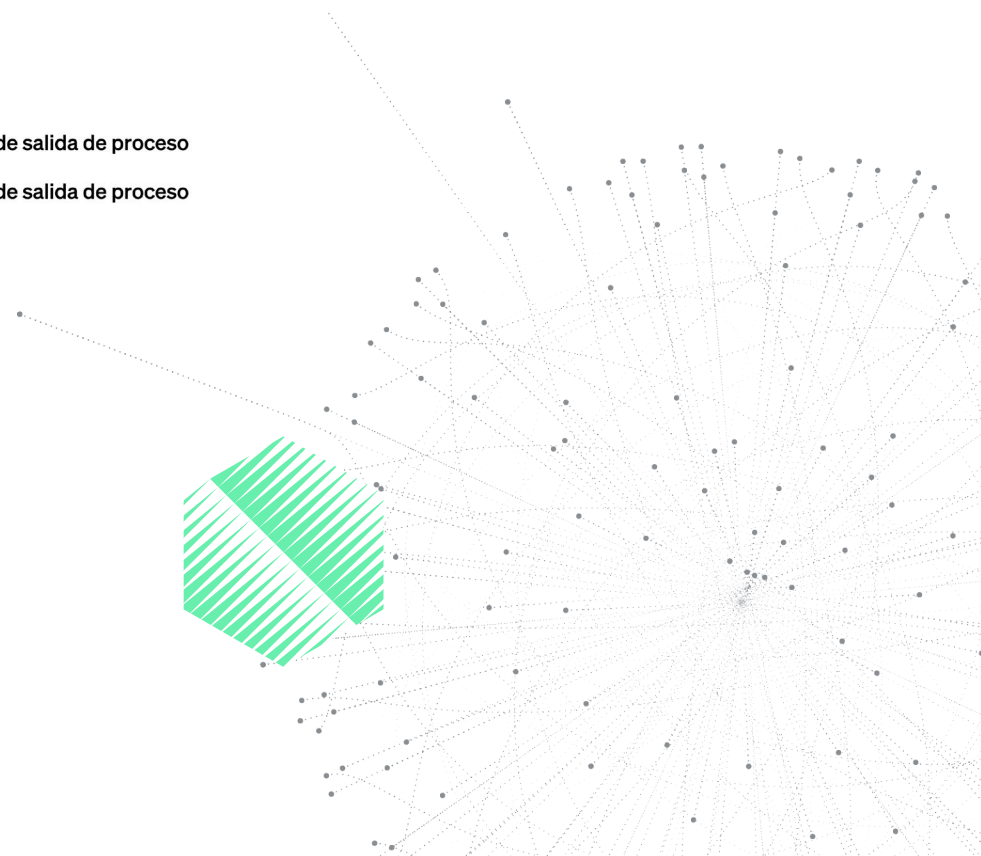
Cuando los equipos instrumentan sistemas para el rastreo distribuido, todas las transacciones generan telemetría de trazas, desde el usuario del front-end hasta las llamadas de la base de datos del back-end. Por ejemplo, cuando los clientes hacen clic en un carrito para hacer una compra en una aplicación de comercio electrónico, esa solicitud pasa por diversos servicios de front-end y back-end a través de varios contenedores, entornos serverless, máquinas virtuales, distintos proveedores de la nube, on-prem, o cualquier combinación de estos. La solicitud podría incluir el servicio de inventario para verificar existencias, además del servicio de pago y del servicio de envío. Y, por último, la solicitud se completa y regresa al usuario. Cada vez que una solicitud pasa de un servicio a otro, emite un span con telemetría de rastreo. Una vez finalizada la consulta, los spans se unen para crear una traza completa del trayecto de la solicitud a través del sistema.

Con el rastreo distribuido, los equipos pueden:

- Trazar la ruta de una solicitud a medida que fluye a través de un sistema complejo.
- Comprender las dependencias de los servicios ascendentes y descendentes.
- Descubrir la latencia de los componentes a lo largo de la ruta.
- Comprender dónde ocurren los cuellos de botella en la ruta de la solicitud.
- Ver y analizar dónde ocurren los errores en la transacción a nivel de servicio individual.



Visualización de gráficos de dispersión y de cascada que muestran el tiempo que cada solicitud tardó en cada etapa de los diferentes servicios de la aplicación



Cuándo se deben usar las trazas

En general, el rastreo distribuido es la mejor manera de que los equipos de DevOps, operaciones, software y SRE obtengan respuestas a preguntas específicas rápidamente en entornos en los que el software es distribuido o depende de arquitecturas serverless. En cuanto una solicitud incluye varios microservicios, se hace esencial contar con una manera de ver cómo los distintos servicios funcionan en conjunto.

Los datos de traza proporcionan contexto para lo que sucede en toda la aplicación en conjunto y entre los servicios y las entidades. Si solo hubiera eventos sin procesar para cada servicio aislado, no habría manera de reconstruir una cadena única entre los servicios para una transacción en particular.

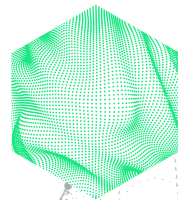
Debido a que las aplicaciones con frecuencia llaman a varias otras aplicaciones según la tarea que están intentando realizar, a menudo también procesan datos en paralelo. Por lo tanto, es posible que la cadena de llamadas no sea uniforme y que los tiempos no sean confiables para la correlación. La única manera de garantizar que una cadena de llamadas sea uniforme es pasar el contexto de traza de un servicio a otro para identificar una sola transacción de manera única en toda la cadena.

Esto quiere decir que los equipos deben usar el rastreo distribuido para obtener respuestas a preguntas como las siguientes:

- ¿Cuál es el estado de los servicios que constituyen un sistema distribuido?
- ¿Cuál es la causa raíz de los errores y defectos dentro de un sistema distribuido?
- ¿Dónde están los cuellos de botella de rendimiento que podrían afectar la experiencia del cliente?
- ¿Cuáles servicios tienen código problemático o ineficiente que los equipos deberían priorizar para la optimización?

Una guía rápida de la terminología del rastreo distribuido:

- Una **transacción** son las llamadas a métodos y funciones que conforman esa unidad de trabajo en una aplicación de software. Comienza cuando se llama al método y termina cuando el método regresa o genera un error.
- Una **solicitud** es cómo las aplicaciones, los microservicios y las funciones se comunican entre sí.
- Una **traza** se refiere a los datos de rendimiento sobre las solicitudes a medida que fluyen por los microservicios.
- Un **span** representa operaciones o segmentos que forman parte de una traza.
- Un **span raíz** es el primer span de una traza.
- Un **span secundario** es un span subsiguiente, que puede estar anidado.



Cómo funcionan las trazas

Cuando están unidas, las trazas forman eventos especiales llamados "spans", que ayudan a hacer el seguimiento de una cadena causal a través de un ecosistema de microservicios de una sola transacción. Para formar spans, cada servicio pasa identificadores de correlación —conocidos como "contexto de traza"— a otro servicio. Este contexto de traza se utiliza para agregar atributos al span.

Timestamp	EventType	TraceID	SpanID	ParentID	ServiceID	Duración
8/11/2022 15:34:23	Span	2ec68b32	aaa111	bbb111	Máquina expendedora	23
8/11/2022 15:34:22	Span	2ec68b32	bbb111	aaa111	Back-end de máquina expendedora	18
8/11/2022 15:34:20	Span	2ec68b32	ccc111	bbb111	Compañía de tarjeta de crédito	15
8/11/2022 15:34:19	Span	2ec68b32	ddd111	ccc111	Banco emisor	3

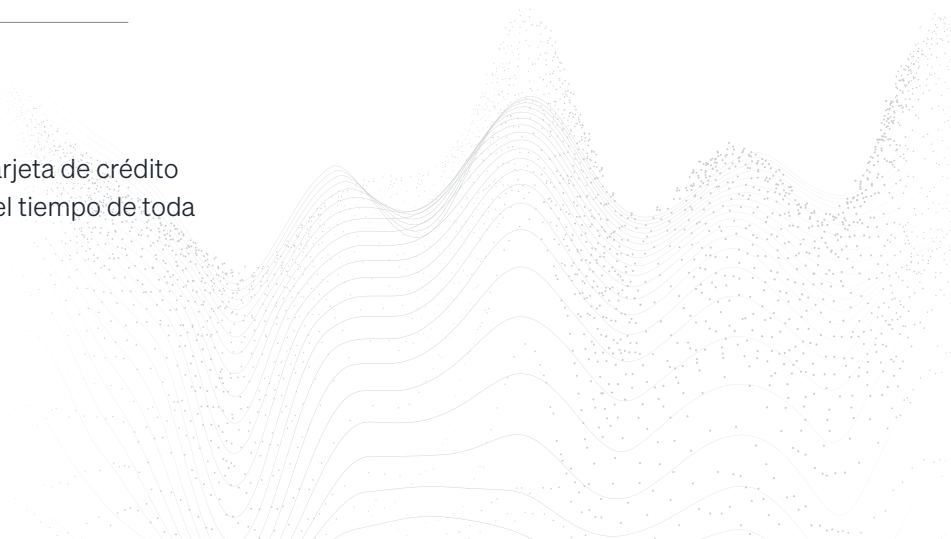
Ejemplo de una traza distribuida compuesta de los spans de una transacción de tarjeta de crédito

En la tabla de arriba, los datos de timestamp y de duración muestran que la compañía de tarjeta de crédito tiene el servicio más lento en la transacción y usa 12 de los 23 segundos: más de la mitad del tiempo de toda la traza.



¿Cómo llegamos a 12 segundos?

El span para contactar al banco emisor se llama un span secundario (child span). El span para contactar a la compañía de tarjeta de crédito es el span principal (parent span). Por lo tanto, si la solicitud al banco tardó 3 segundos, la compañía de tarjeta de crédito tardó 15 segundos y se resta el span secundario del principal, el resultado es que se necesitaron 12 segundos para procesar la transacción de la tarjeta de crédito.



Sacar conclusiones

A medida que las organizaciones comenzaron a migrar a las aplicaciones distribuidas, muy pronto se dieron cuenta de que necesitaban una manera de visualizar los microservicios individuales de forma aislada y todo el flujo de solicitudes. Esta migración es la razón por la cual el rastreo distribuido se convirtió en una mejor práctica para obtener la visibilidad necesaria de todo lo que sucede. Y si se combinan las trazas con los otros tres tipos esenciales de datos de telemetría —métricas, eventos y logs—, los equipos tendrán un panorama completo del entorno y del rendimiento del software para la observabilidad de extremo a extremo.

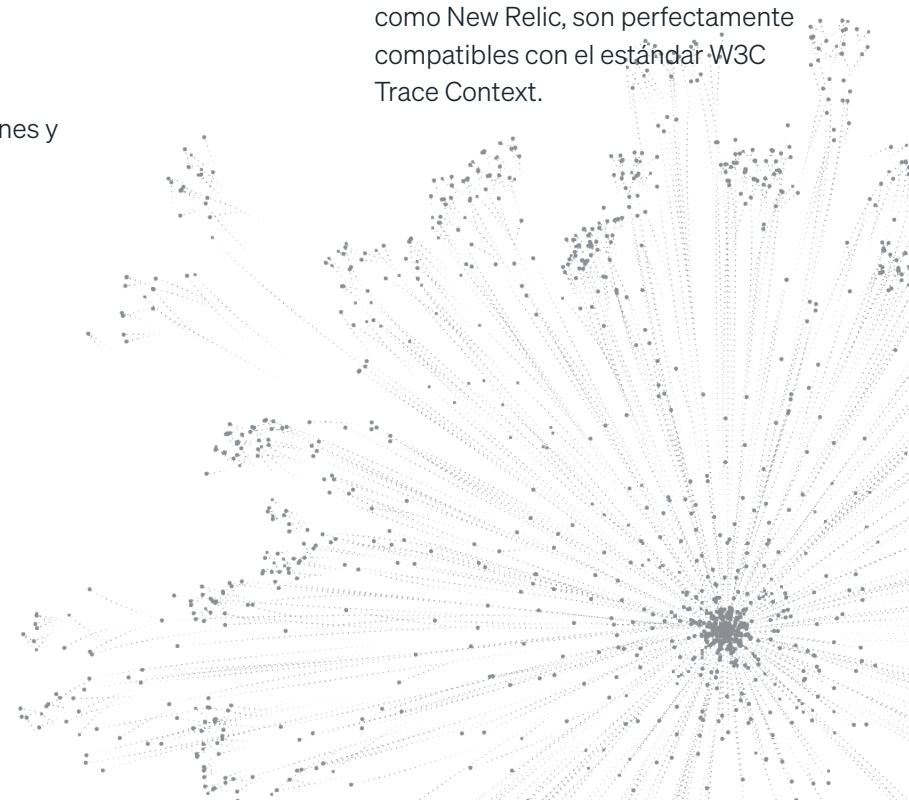
El rastreo distribuido también requiere el contexto de traza. Este requisito quiere decir asignar una ID única a cada solicitud, asignar una ID única a cada etapa de una traza, codificar esta información contextual y pasar (o propagar) el contexto codificado de un servicio a otro a medida que la solicitud avanza por el entorno de una aplicación. Este proceso permite que la herramienta de rastreo distribuido correlacione cada etapa de una traza, en el orden correcto, junto con otra información necesaria para monitorear y hacer seguimiento del rendimiento.

Una sola traza suele capturar datos acerca de:

- Spans (nombre del servicio, nombre de la operación, duración y otros metadatos)
- Errores
- Duración de las operaciones importantes dentro de cada servicio (como las llamadas a funciones y métodos internos)
- Atributos personalizados



[W3C Trace Context](#) se ha convertido en el estándar para la propagación del contexto de traza más allá de los límites del proceso. Permite que todos los rastreadores y los agentes que cumplen con el estándar participen en una traza, y que los datos de la traza se propaguen desde el servicio raíz hasta el servicio final. Muchos proveedores de observabilidad, como New Relic, son perfectamente compatibles con el estándar W3C Trace Context.



Por qué las organizaciones necesitan rastreo distribuido

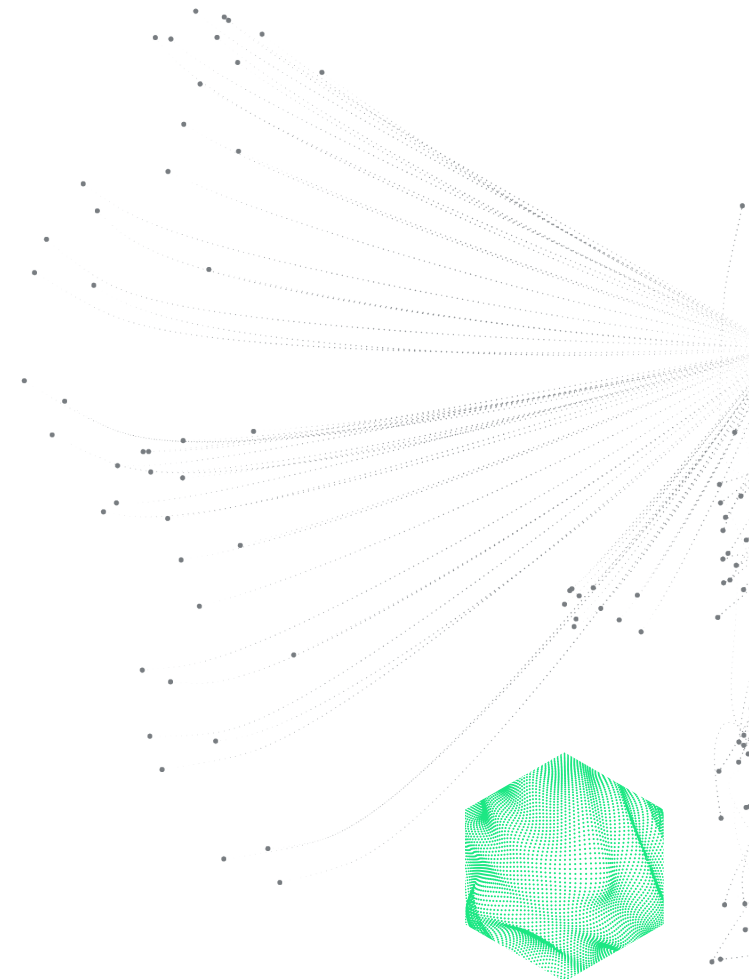
A medida que las nuevas tecnologías y prácticas —nube, microservicios, contenedores, funciones serverless, DevOps, ingeniería de fiabilidad del sitio, y muchos más— aumentan la velocidad y reducen la dificultad de desarrollar software desde el código hasta la producción, también introducen nuevos desafíos:

- Más puntos de falla en el stack de aplicaciones
- Aumento del MTTR debido a la complejidad del entorno de la aplicación
- Menos tiempo para que los equipos se dediquen a la innovación porque necesitan más tiempo para diagnosticar problemas

Por ejemplo, una solicitud lenta podría afectar la experiencia de un grupo de clientes. Esa solicitud se distribuye en varios microservicios y funciones serverless. Varios equipos monitorean y son responsables de los distintos servicios involucrados en la solicitud, y ninguno ha reportado problemas de rendimiento en sus microservicios. Sin una manera de ver el rendimiento de toda la solicitud en los distintos servicios, es casi imposible identificar con precisión dónde y por qué hay latencia alta y qué equipo debería encargarse del problema. Como parte de una estrategia de observabilidad de principio a fin, el rastreo distribuido aborda los desafíos de los entornos de aplicaciones modernos.

Si comprenden a fondo el rendimiento de cada servicio —tanto ascendente como descendente— los equipos de software podrán hacer lo siguiente con más eficacia y rapidez:

- Identificar y resolver problemas para minimizar el impacto en la experiencia del cliente y los resultados comerciales.
- Medir el estado general del sistema y comprender el efecto de los cambios en la experiencia del cliente.
- Priorizar las áreas de alto valor para mejorarlas y optimizar las experiencias digitales de los clientes.
- Innovar continuamente con confianza para superar a la competencia.



Obtener visibilidad de la canalización de datos

El rastreo distribuido requiere el reporte y el procesamiento de la telemetría de rastreo. El volumen de datos de traza puede crecer exponencialmente con el tiempo a medida que aumenta el volumen de solicitudes y los equipos despliegan más microservicios en el entorno.

Por esta razón, muchas organizaciones utilizan el muestreo de datos para administrar la complejidad y el costo asociados con la transmisión de la actividad de trazas. Lo ideal es que los datos muestreados representen las características del conjunto de datos completo.

Los equipos de software necesitan flexibilidad para elegir el muestreo al inicio (head-based) o al final (tail-based) del flujo de trabajo para satisfacer los requisitos de monitoreo de cada aplicación.

Muestreo eficiente al inicio

El muestreo al inicio del flujo de trabajo recopila y almacena los datos de traza de forma aleatoria mientras el span raíz (el primero) se procesa para hacer seguimiento y analizar lo que ocurre con una transacción en todos los servicios que toca. Normalmente, el muestreo al inicio se realiza dentro del agente responsable de recopilar la telemetría de trazas seleccionando aleatoriamente qué trazas se deben muestrear para el análisis. Las decisiones de muestreo ocurren antes de que se completen las trazas. Como no hay manera de saber qué traza podría encontrar un problema, es posible que los equipos no vean algunas trazas que contienen procesos excepcionalmente lentos o errores.

El muestreo al inicio es conveniente para proporcionar un muestreo estadístico global de las solicitudes en un sistema distribuido. Es útil para detectar las trazas con errores o latencia en aplicaciones con un volumen más bajo de transacciones y entornos con una combinación de arquitecturas monolíticas y basadas en microservicios. El muestreo al inicio es una manera eficiente de muestrear una gran cantidad de datos de traza en tiempo real, con poco o nada de impacto en el rendimiento de la aplicación.

Ventajas del muestreo al inicio

- Resulta bien para las aplicaciones con un menor rendimiento de transacciones
- Fácil y rápido para poner en funcionamiento
- Adecuado para entornos mixtos de monolíticos y microservicios donde aún predominan los monolíticos
- Poco o nada de impacto en el rendimiento de la aplicación
- Una solución de bajo costo para enviar datos de rastreo a proveedores de terceros
- El muestreo estadístico proporciona una transparencia adecuada del sistema distribuido

Limitaciones del muestreo al inicio

- Las trazas se muestrean de manera aleatoria
- El muestreo se realiza antes de que la traza haya completado todo su recorrido a través de los distintos servicios, de modo que no hay manera de saber de antemano cuál traza podría encontrar un problema
- En los sistemas de alto rendimiento, las trazas con errores o una latencia poco común podrían no muestrearse y pasarse por alto

Trazas accionables con muestreo al final

El rastreo distribuido con muestreo al final ayuda a los equipos de software a resolver problemas en sistemas altamente distribuidos y de alto volumen donde los equipos deben observar toda la telemetría de trazas y muestrear las trazas que contienen errores o presentan una latencia inusual. El muestreo al final del flujo de trabajo recopila toda la información acerca de esa traza una vez que se ha completado.

Además, para los equipos que necesitan ver hasta el último detalle a la hora de resolver problemas, el muestreo al final, más que una preferencia, es una necesidad.

Algunas organizaciones necesitan su herramienta de rastreo distribuido para observar y analizar todos los spans —cada paso de un servicio a otro— y mostrar las trazas más útiles para la resolución de problemas porque el tiempo de inactividad puede costar millones de dólares, sobre todo durante los picos.

Por ejemplo, una organización con una carga promedio de tres millones de spans por minuto ve picos de 300 millones de spans por minuto cuando lanza un nuevo producto. El muestreo tradicional al inicio del flujo de trabajo no es adecuado para este tipo de organización que tiene un volumen alto de transacciones.

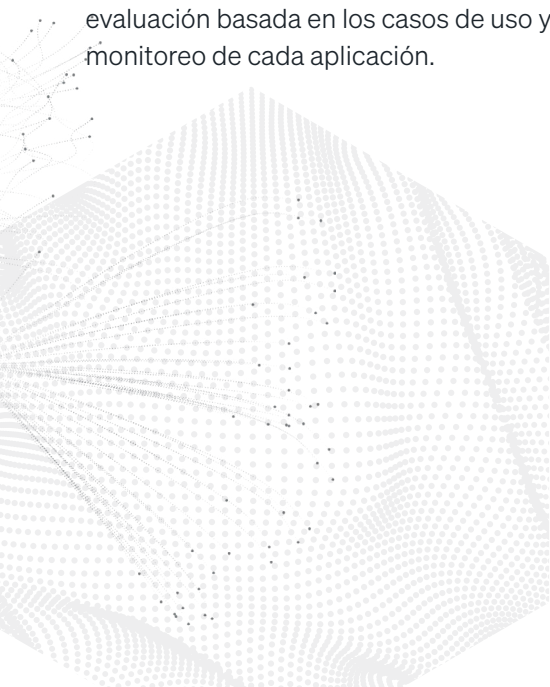
No todas las trazas son iguales. Para elegir el mejor método de muestreo, los equipos deben hacer una evaluación basada en los casos de uso y el análisis de costo-beneficio y tener en cuenta las necesidades de monitoreo de cada aplicación.

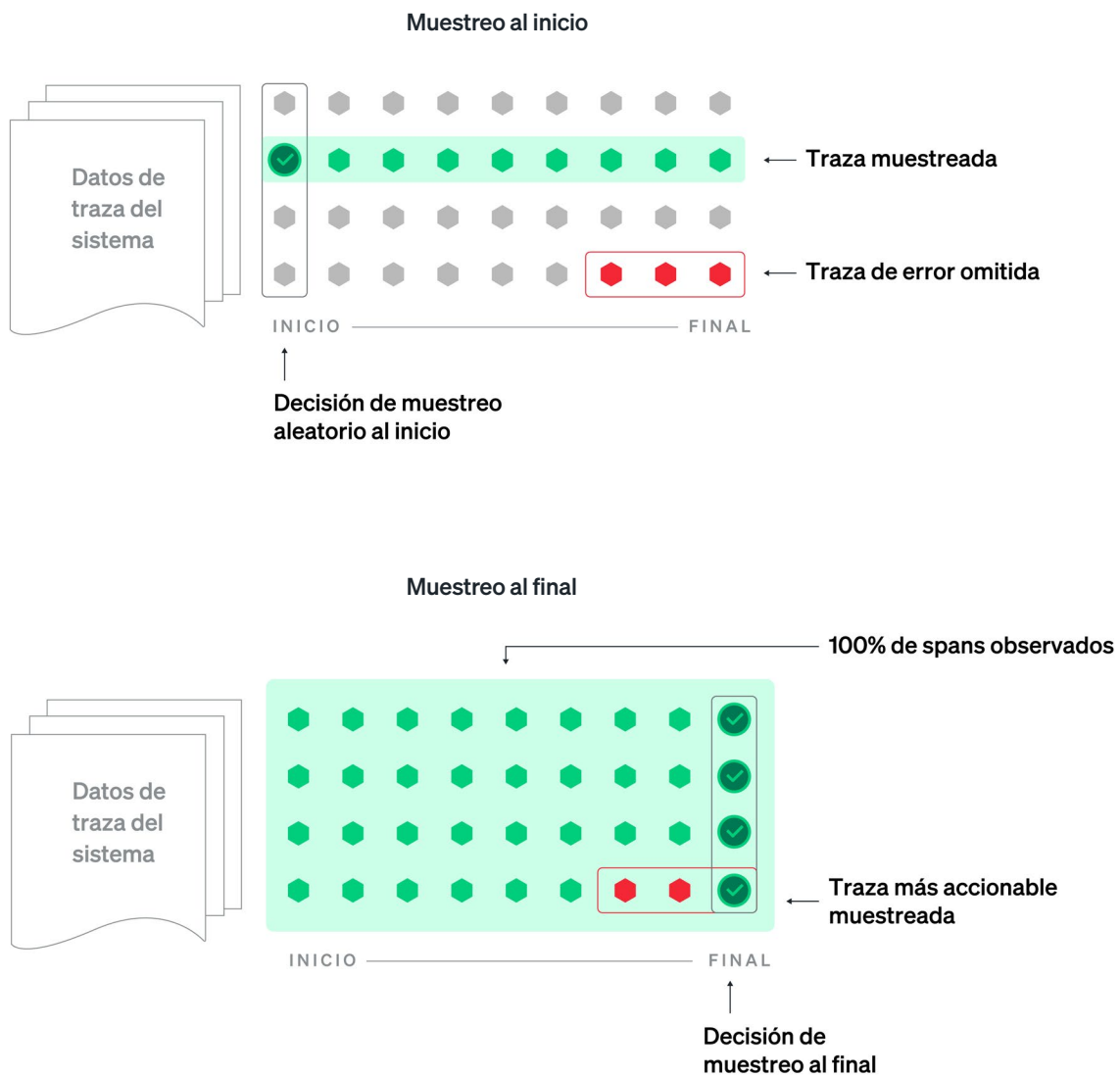
Ventajas del muestreo al final

- Observa y analiza el 100% de las trazas
- Toma muestras después de que las trazas se han completado
- Visualiza las trazas con errores o una lentitud inusual más rápidamente

Limitaciones del muestreo al final

- Puede requerir puertas de enlace, proxies y satélites adicionales para ejecutar el software de muestreo
- Exige un poco de esfuerzo para administrar y escalar el software de terceros en algunos casos
- Incurrir en costos adicionales para la transmisión y el almacenamiento de más datos





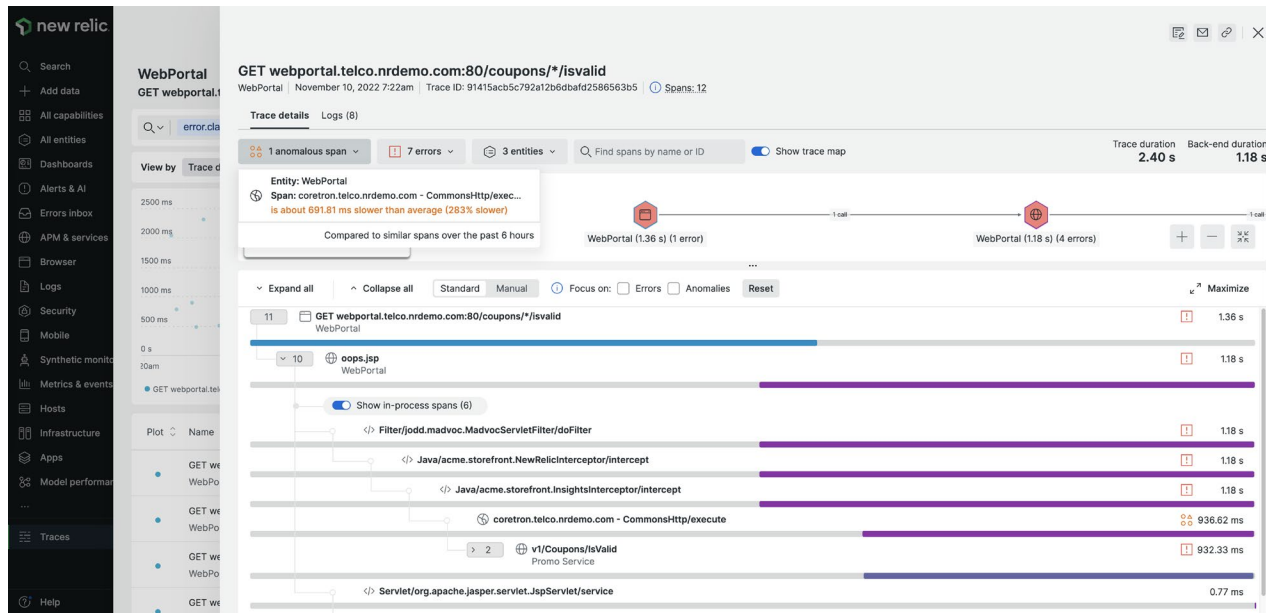
Muestreo tradicional al inicio (arriba) y muestreo al final (abajo)



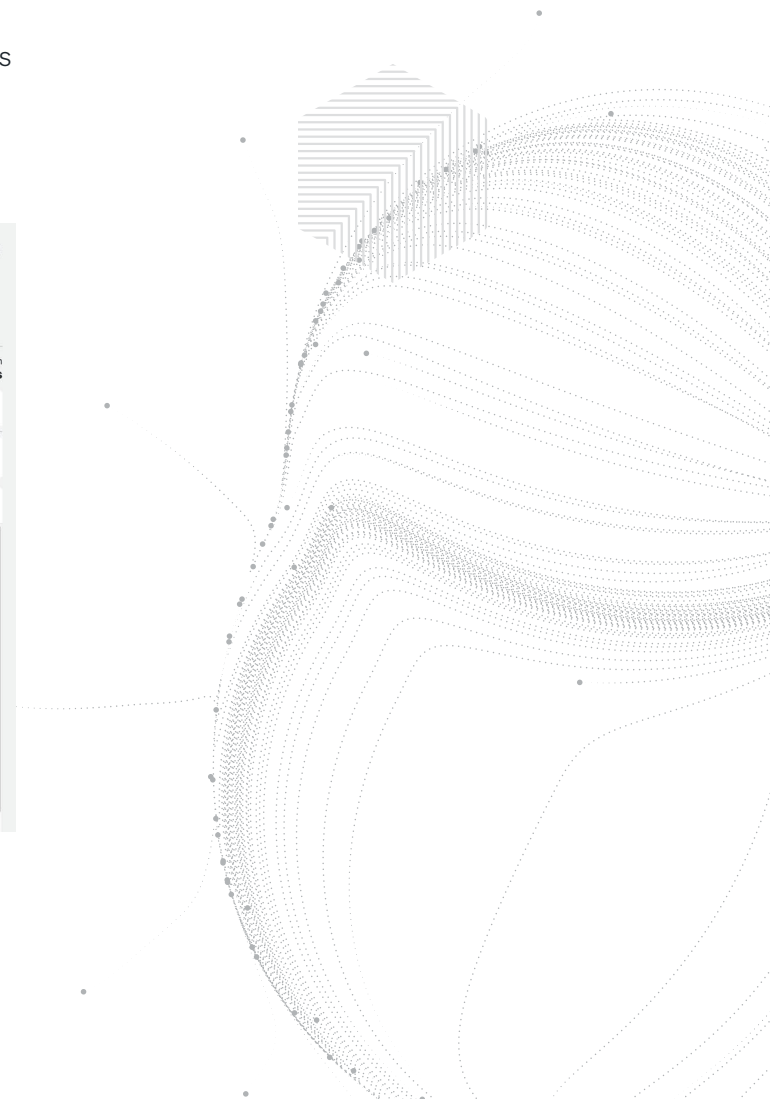
Análisis y visualización

La recopilación de datos de traza es una pérdida de tiempo si los equipos de software no tienen una manera fácil de analizar y visualizar los datos en las arquitecturas complejas. Una plataforma de observabilidad completa permite a los equipos ver todos sus datos de negocios y de telemetría en un solo lugar. Además, proporciona el contexto que necesitan para comprender el significado, tomar las medidas necesarias rápidamente y trabajar con los datos de manera significativa.

Lo ideal es que la visualización de una traza distribuida tenga una estructura de árbol. La visualización debe incluir spans secundarios que hagan referencia a un span principal y permitir que los equipos vean qué spans tienen latencia alta y errores dentro de una traza. También ayuda a los equipos a comprender los detalles exactos de los errores y qué servicios están lentos, con atributos detallados para encontrar los problemas y corregirlos rápidamente.



Rastreo distribuido de New Relic



Enfrentarse a la carga de administración

La resolución de problemas de los sistemas distribuidos es un problema clásico de aguja en un pajar, y la instrumentación de sistemas para el rastreo y la posterior recopilación y visualización de los datos puede ser una tarea laboriosa y compleja de implementar. Las soluciones de software como servicio (SaaS) completamente gestionadas permiten a los equipos eliminar la carga pesada de desplegar, administrar y escalar puertas de enlace o satélites de terceros para la recopilación de datos.

La plataforma de observabilidad de New Relic hace que sea fácil instrumentar aplicaciones con un despliegue de agente único para casi cualquier framework y lenguaje de programación. Los equipos también pueden utilizar herramientas de código abierto y estándares abiertos de instrumentación para instrumentar los entornos. OpenTelemetry es considerado el estándar para la instrumentación de código abierto y la recopilación de telemetría.

La plataforma de New Relic también ofrece un servicio de muestreo al final completamente gestionado que observa y analiza el 100% de los spans en un sistema distribuido y proporciona visualizaciones para las trazas con errores y una latencia inusual, para que los equipos puedan identificar y resolver los problemas rápidamente.

La plataforma observa cada span y proporciona métricas, datos de errores y trazas esenciales en una sola vista. Proporciona información crítica al guardar los datos más accionables en la plataforma de New Relic. El resultado es una visibilidad inigualable de los sistemas distribuidos, lo que permite a los equipos comprender el impacto de los errores o la latencia descendente con métricas detalladas y luego profundizar hasta los datos de traza para buscar las trazas más relevantes.

El rastreo distribuido está incluido con New Relic APM, con transferencia de datos de baja latencia y bajo costo desde agentes de New Relic, instrumentación en las funciones serverless, o cualquier otra fuente de datos, incluida la instrumentación de terceros.

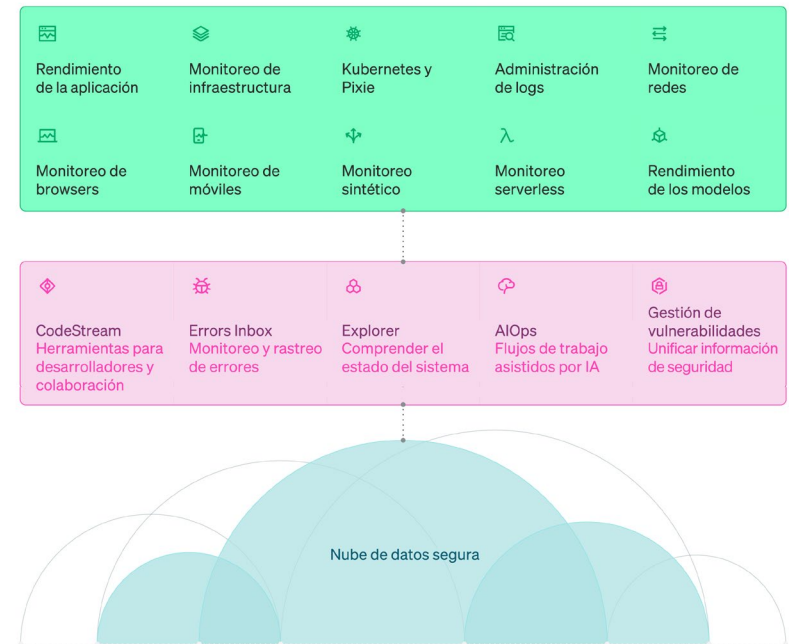
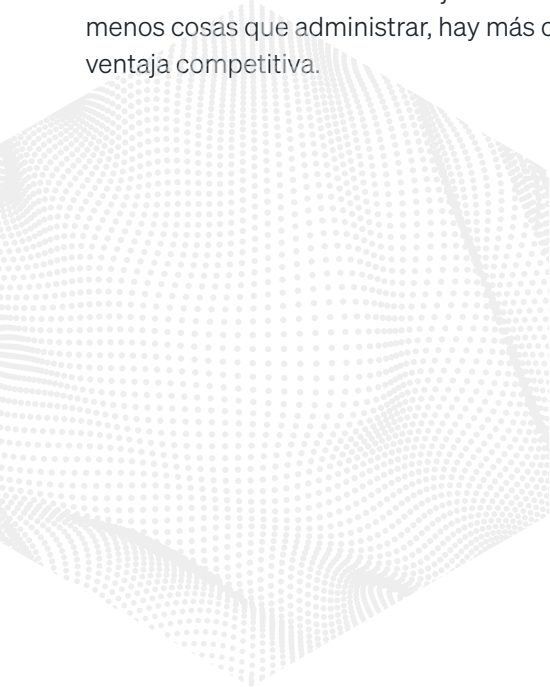
Con New Relic es posible:

- Disfrutar de un servicio local basado en la nube completamente gestionado que se escala a pedido.
- Observar y analizar el 100% de las trazas en los sistemas distribuidos.
- Visualizar las trazas más accionables que contienen errores o una latencia poco común.
- Eliminar el trabajo pesado de desplegar, administrar, admitir y escalar puertas de enlace o satélites de terceros en los entornos.
- Aprovechar la compatibilidad completa con la instrumentación y estándares abiertos para la telemetría de trazas.
- Reducir el costo de salidas de datos mediante proximidad a cargas de trabajo de la nube.
- Resolver problemas de manera más eficiente.
- Reducir el tiempo medio de detección (MTTD) y el tiempo medio de resolución (MTTR) con trazas accionables de alta fidelidad.
- Permitir que los ingenieros y desarrolladores se enfoquen en el trabajo más importante, como el desarrollo de nuevas características.

¿Muestreo al inicio o al final? Cualquiera de los dos.

New Relic ofrece opciones flexibles para el rastreo distribuido para que los equipos puedan tomar decisiones sobre el muestreo al inicio o al final a nivel de la aplicación. Para las aplicaciones críticas en las que los equipos necesitan observar y analizar cada traza, pueden seleccionar el muestreo al final sin preocuparse de la administración de la infraestructura de muestreo.

New Relic es el único proveedor de observabilidad que brinda a los equipos de software la flexibilidad de seleccionar el rastreo distribuido con muestreo al inicio o muestreo al final del flujo de trabajo completamente gestionado. Con menos cosas que administrar, hay más oportunidades para innovar y obtener una ventaja competitiva.



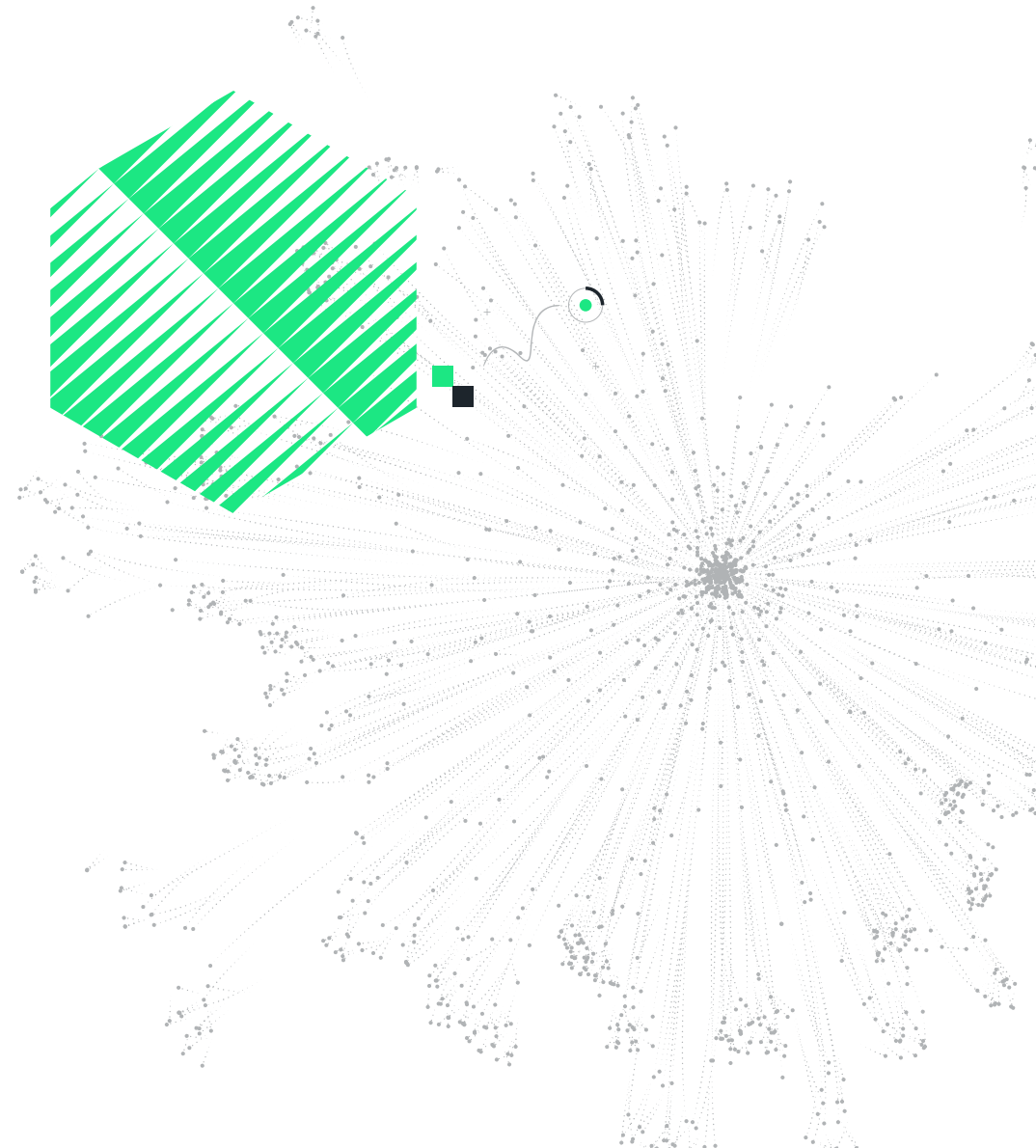
La plataforma de observabilidad de New Relic incorpora administración de logs, APM, rastreo distribuido, monitoreo de infraestructura, monitoreo serverless, monitoreo de móviles, monitoreo de browsers, monitoreo sintético, monitoreo de Kubernetes, y mucho más.

Próximos pasos

Para comenzar a utilizar New Relic APM con el rastreo distribuido, [regístrate para una cuenta gratuita hoy mismo](#). Las cuentas gratuitas incluyen 100 GB de ingesta de datos al mes y un número ilimitado de usuarios Basic.

¿Ya tienes una cuenta de New Relic? Comenzar a utilizar el rastreo distribuido de New Relic APM es fácil; basta con usar nuestro agente APM más reciente. Obtén más información sobre las opciones de configuración del rastreo distribuido.

Comenzar ahora



Acerca de New Relic

Como líder en observabilidad, New Relic proporciona a los ingenieros un enfoque basado en datos para planificar, desarrollar, desplegar y ejecutar un software excelente. New Relic ofrece la única plataforma de datos unificada con toda la telemetría — métricas, eventos, logs y trazas— junto con potentes herramientas de análisis de todo el stack para ayudar a los ingenieros a hacer su mejor trabajo con datos, no con opiniones.

El modelo de precios intuitivo y predecible de New Relic es el primero en la industria que se basa en el uso y ofrece a los ingenieros más valor por su dinero al ayudar a acelerar los ciclos de planificación, cambiar las tasas de fallas, la frecuencia de lanzamiento y el tiempo medio de resolución (MTTR). Esto ayuda a las mejores marcas del mundo y a las startups de hipercrecimiento a mejorar su tiempo de actividad, fiabilidad y eficiencia operativa, además de brindar experiencias excepcionales a los clientes que estimulan la innovación y el crecimiento.

