

Aumente la observabilidad para agilizar la modernización de aplicaciones



Índice



- A photograph of a woman with long, light brown hair, shown from the side and looking down. She is wearing a dark top. The background is a bright, possibly indoor setting with a window. Overlaid on the right side of the image is a large, semi-transparent grid consisting of 10 columns and 10 rows of blue dots, creating a pattern that covers approximately one-third of the right half of the frame.

0.1 →

Información general

0.5 →

El error de no modernizar

02 →

Elementos de la modernización de aplicaciones

06 →

El papel de la observabilidad

03 →

Comience su jornada de modernización

07 →

Observabilidad empresarial con IBM Instana

0.4 →

Beneficios de modernizar las aplicaciones

08 →

¿Es IBM Instana adecuada para usted?

Información general

Todos están de acuerdo en que la modernización de aplicaciones es un tema de moda. Y uno sobre el que la mayoría de las empresas está hablando o lo hará pronto. La modernización de aplicaciones es un proceso complejo que incluye la evaluación de los portafolios de aplicaciones actuales para determinar cuáles son las más importantes para modernizar y luego formular un plan rentable de actuación. Lamentablemente, la realidad de la modernización de las aplicaciones suele ser mucho más compleja de lo que los promotores del desarrollo transmiten en las redes sociales.

Aunque las aplicaciones de microservicios basadas en contenedores han ganado terreno, muchas aplicaciones heredadas todavía se deben migrar. La infraestructura local y las aplicaciones monolíticas suelen ser el punto de partida y las aplicaciones nativas de la nube son la meta final. Sin embargo, cada organización debe planificar su camino en función de su estado actual, requisitos y viabilidad. Ya se está produciendo una migración considerable y se espera que aumente en los próximos años.



Elementos de la modernización de aplicaciones



Código

La escritura y actualización del código de una aplicación es una tarea ardua. Solo enviarla es un desafío. Sin embargo, la Ley de Murphy nos dice que, a la larga, la aplicación se tendrá que actualizar cuando el propietario del código no esté disponible. El propietario de la aplicación tal vez ya no trabaje en la empresa y no esté disponible.

La nueva persona se podría encontrar con una sorpresa desagradable. Código personalizado, comentarios personales, un orden extraño, la falta de comentarios, no usar la sintaxis aceptada y un formato deficiente pueden convertir las tareas sencillas en un trabajo de investigación. Un lenguaje Java obsoleto podría inhabilitar las integraciones críticas en su sitio web. Y eso podría perjudicar la empresa.

Componentes de la aplicación

Una aplicación monolítica en servidores bare metal no se puede volver nativa de la nube por arte de magia. La aplicación se debe dividir en microservicios, sus componentes se deben poner en contenedores y esos contenedores se deben orquestar con una herramienta como Kubernetes.

Solo ese proceso representa mucho trabajo, pero proporciona un comienzo para los componentes de la aplicación. Cuanto más complejos sean los componentes, más tiempo se necesita de los desarrolladores e ingenieros, lo que generalmente crea urgencias para migrar a un servicio de hosting en la nube. Deje que ellos gestionen la infraestructura para que su equipo pueda gestionar las aplicaciones.

Infraestructura

¿Recuerda cuando dijimos que los cambios en el código, los componentes y la infraestructura pueden suceder en paralelo? Si está migrando una aplicación a un entorno de nube, tiene varias opciones de componentes de aplicaciones de infraestructura.

- **Realojar.** Cargar y mover la aplicación y realojarla en la nube.
- **Redefinir la plataforma.** Alojar la aplicación en la nube y realizar pequeños cambios en la infraestructura.
- **Volver a comprar.** Comprar versiones de software como servicio (SaaS) de sus aplicaciones.
- **Refactorizar.** Reescribir parte o toda la aplicación y posiblemente implementar una nueva arquitectura de aplicación, como por ejemplo convertir una aplicación de arquitectura orientada a servicios (SOA) a contenedores.

Comience su jornada de modernización

¿Por dónde puede empezar? Depende de dónde se encuentre. En general, usted quiere pasar al siguiente paso en la escala progresiva hacia aplicaciones nativas de la nube, pero cada paso tiene su propio conjunto de desafíos.

Usted está aquí	Su siguiente paso
Bare metal, local	Virtualizado, local
Virtualizado, local	Centros de datos
Centros de datos	Nube privada
Nube privada	Nube pública
Nube pública	Nube híbrida
Nube híbrida	Nativo de la nube

Determinar el éxito de cada paso en el proceso requiere una medición de referencia del rendimiento de las aplicaciones antes de la modernización y una nueva medición de comparación posteriormente. Mantener la visibilidad para recopilar el mismo conjunto de métricas en cada etapa puede ser difícil para algunas herramientas de supervisión del rendimiento de las aplicaciones (APM).

Definición de heredado para este libro electrónico

La frase “aplicación heredada” puede tener varias definiciones diferentes. En este libro electrónico, proporcionamos una definición de aplicaciones heredadas que nos facilita el uso de heredada y aplicación heredada en lugar del nombre completo más largo y calificado.

En este análisis, aplicación heredada significará alguna forma de aplicación de Java Enterprise Edition (Java EE) o .NET (punto net), ya sea implementada como un monolito o en un entorno SOA.

Para este fin, este libro electrónico también puede hacer referencia a las herramientas APM de generación 1 y 2 como supervisión heredada o APM heredada. Cualquier análisis sobre otras herramientas heredadas se mencionará específicamente.

Beneficios de modernizar las aplicaciones



La migración de códigos, componentes de aplicaciones e infraestructura implica decisiones complejas y requiere mucha planificación. Teniendo en cuenta la dificultad, ¿está preparado para asumir esto? Esta es la razón por la que lo debería estarlo.

La modernización de aplicaciones es una parte clave del proceso de transformación digital que ayuda a automatizar los procesos manuales. La automatización puede tener un impacto positivo tanto en el rendimiento de los servicios como en la maximización de los recursos humanos. Los procesos que requieren intervenciones manuales frecuentes pueden ser más lentos, ser más propensos a errores y más caros de operar. Estas son algunas razones para modernizar las aplicaciones.

- Optimizar el tiempo de los desarrolladores.
- Acelerar la innovación y el canal de integración continua y entrega continua (CICD).
- Reducir los costos y la redundancia.
- Ser más receptivo a las necesidades de los usuarios.
- Crear más uniformidad en la organización y los procesos.
- Dejar de lado los entornos más antiguos.
- Resolver los problemas con mayor rapidez.

Optimizar el tiempo de los desarrolladores

Muchas aplicaciones heredadas están basadas en Java o .NET, pero algunas carecen de la sofisticación de las aplicaciones más recientes. Las actualizaciones y el mantenimiento de rutina pueden ser difíciles para los equipos de desarrollo. La falta de conocimientos históricos puede hacer que en caso de que falte un desarrollador se forme un cuello de botella para todo el trabajo en una aplicación.

Cuando un desarrollador asume el papel de otro, el nuevo desarrollador puede reconstruir una aplicación con correcciones y parches con su propio estilo. Como resultado, las inconsistencias son abrumadoras, como los enlaces a archivos de funciones de JavaScript en algunos lugares; y JavaScript directamente en el código, en otros. Las correcciones sencillas se traducen en horas dedicadas a examinar el código para encontrar la aguja en el pajar.

Cuando los desarrolladores aplican los principios de rigor y experiencia del usuario a su código, es más fácil para todos trabajar con el código o, mejor aún, se permite automatizar el trabajo rutinario. Los comentarios definen a qué hace referencia cada sección en la aplicación. Los scripts y las hojas de estilo indican qué hacen esos elementos. Y si el desarrollador original utiliza tabulaciones o espacios, es consistente.

El resultado es menos tiempo para descifrar el código y más tiempo realizando el trabajo, menos estrés y más logros, y un mejor rendimiento del equipo.

Acelerar la innovación y el canal de CICD

En la mayoría de los casos, las aplicaciones más antiguas son menos resilientes. Como resultado, los desarrolladores suelen pasar demasiado tiempo resolviendo emergencias para mantener las aplicaciones funcionando en lugar de innovando para los clientes.

Cuando los desarrolladores completan más trabajo al principio de un proyecto de desarrollo, empiezan un círculo virtuoso de asignación de recursos que ayuda a mantener los proyectos conforme a los plazos y reducir los errores. Por el contrario, cuando los cronogramas fallan, se inicia un círculo vicioso de desesperación para abordar las emergencias. Eso significa que los desarrolladores ejecutan pruebas manualmente en cada etapa en lugar de ejecutar pruebas automatizadas continuas. Instrumentan manualmente el código en lugar de automatizarlo. Documentan minuciosamente su infraestructura en lugar de utilizar una herramienta que puede

detectar todo automáticamente. Este proceso puede causar resultados de prueba engañosos. Cuando eso sucede, el código puede llegar a producción antes de que esté listo.

Las organizaciones están aplicando arquitecturas diseñadas para ejecutar las aplicaciones, como contenedores, microservicios, computación sin servidor y orquestación de Kubernetes para el desarrollo y entrega de aplicaciones. Las herramientas de automatización y orquestación como Jenkins, Red Hat® Ansible®, Chef y Puppet pueden ayudar a los equipos a mantenerse en sintonía. Las principales plataformas de nube pública (Amazon Web Services (AWS), Microsoft Azure y Google Cloud) ofrecen procesamiento de contenedores gestionado por Kubernetes como servicio.

Reducir los costos y la redundancia

Uno de los pasos avanzados que las organizaciones dan antes de migrar a un hosting en la nube es ejecutar centros de datos. Los centros de datos globales son buenos para el balanceo de carga y la redundancia, pero pueden ser muy caros.

Además, los centros de datos suelen venir con contratos a largo plazo. La migración a la nube es un proceso que lleva tiempo. Para tener el tiempo suficiente, debe empezar al menos un años antes de que expire su contrato con el centro de datos.

Y si usted creía que los centros de datos ofrecían redundancia, pruebe docenas de contenedores con microservicios en la nube. Eso es redundancia. Algunas grandes corporaciones siguen utilizando arquitecturas locales y hacen que funcionen. Sin embargo, es poco probable que el patrón hacia la adopción de la nube se revierta.

Ser más receptivo a las necesidades de los usuarios

Los requisitos y listas de deseos de sus clientes y empleados deben ser su referente. Los clientes tienen poca paciencia para las demoras e interrupciones, tanto en línea como en las aplicaciones.

Por ejemplo, si sus páginas web tardan demasiado tiempo en cargar, es probable que un cliente haga clic en alguna otra. Para los minoristas en línea, esto puede dar lugar a una reducción de ingresos considerable como resultado de las ventas perdidas. Para obtener resultados óptimos, tenga como objetivo que una página se cargue o un proceso de una aplicación se complete en 2 a 4 segundos.



Crear más uniformidad en la organización y los procesos

¿Recuerda cómo los desarrolladores crean sus propias correcciones y parches para remendar el código? Hay consecuencias concretas. Las diferentes aplicaciones se codifican de manera diferente, lo que dificulta exigir que cumplan con los estándares mínimos. ¿Cómo se puede aplicar un estándar a una aplicación que no utiliza ninguno de los componentes que está tratando de estandarizar? Las organizaciones modernizan las aplicaciones para alinearlas a fin de que estas estén en conformidad.

Dejar de lado los entornos más antiguos

Este es un caso de ejemplo. Una aplicación está basada en Java 11. La versión actual es Java 20. Si el equipo va a actualizar la versión y mejorar en consecuencia el código, ¿por qué no aprovechar la oportunidad para modernizar la aplicación de otras maneras? Dividirla en microservicios. Trasladarla a la nube. Convertirla en una aplicación nativa de la nube.

Resolver problemas

Los componentes distribuidos permiten que los encargados de solucionar incidentes deshabiliten algunos contenedores u otras piezas sin necesidad de desactivarlos todos. En otras palabras, los componentes de aplicaciones de microservicios distribuidos y basados en la nube eliminan el punto único de falla que afecta a las aplicaciones monolíticas. Los contenedores también permiten a los desarrolladores aislar y probar las piezas de una aplicación sin que esto afecte a todas.

Los orquestadores como Kubernetes también facilitan la aplicación de correcciones a las aplicaciones sin tiempo de inactividad del cliente, lo que es una gran victoria tanto para el proveedor como para el usuario.

Por qué las empresas cometén el error de no modernizar

Muy a menudo las empresas retrasan la modernización de las aplicaciones por dos razones principales: las molestias a corto plazo y la dificultad de medir el progreso. Equilibrar la incomodidad a corto plazo con los beneficios puede ser difícil, ya que lo primero es más seguro y lo segundo puede ser una apuesta. Medir el progreso durante todo el proceso de modernización también puede ser un obstáculo considerable.

Molestias a corto plazo

Esta es una lista rápida de las razones por las que las empresas demoran o cancelan los proyectos de modernización de las aplicaciones.

- **Indecisión del cliente.** El proceso de modernización generalmente requiere algunas interrupciones y para industrias como la de los servicios financieros, los temas de privacidad y las interrupciones son causa de alarma.

– **Tiempo.** ¿Recuerda la migración?

Cada uno de esos pasos puede tardar fácilmente hasta un año. Solo la división de un monolito en microservicios requiere entender qué capacidades desea implementar por separado y dividir el proceso en pasos para poder medir el progreso.

- **Dinero.** A corto plazo, tanto las actualizaciones de código como los cambios arquitectónicos pueden ser caros. Si suma los costos del tiempo que pasan los desarrolladores atendiendo emergencias, la incorporación de nuevas contrataciones, la rotación, la capacitación y la congelación del código previsto, los gastos ocultos pueden acumularse rápidamente.

Las empresas que esperan dejar de gastar millones en centros de datos suelen terminar pagando costos adicionales por el servicio de hosting.

Incapacidad de establecer un punto de referencia y comparar el rendimiento

Cómo escribió Peter Drucker, consultor de gestión, “Si no se puede medir, no se puede mejorar”.

Como verá, la modernización no es sencilla. Las organizaciones que ejecutan aplicaciones monolíticas pueden utilizar herramientas de APM heredadas para medir las métricas adecuadas. Sin embargo, una vez que los componentes de la aplicación se dividen y ejecutan en contenedores, algunas herramientas de APM heredadas no pueden ver dentro de los contenedores. Además, con el hosting en la nube, hay todavía menos herramientas de APM heredadas que estén a la altura de esa tarea de observabilidad.

El papel de la observabilidad en la modernización de aplicaciones



Es fundamental añadir la observabilidad a su portafolio cuando planifica y ejecuta la modernización de aplicaciones y la transformación digital. El control de calidad de las aplicaciones es un requisito para llevar a cabo la modernización.

- Establezca un punto de referencia y compare con las mismas mediciones.
- Es más fácil para una solución Docker contenedrizada medir a nivel local.
- Una herramienta de APM heredada no puede medir las aplicaciones nativas de la nube.

Establecer un punto de referencia antes de planificar la modernización de una aplicación es fundamental para que usted pueda determinar dónde el rendimiento de su aplicación es deficiente y dónde es sólido, y dónde rinde conforme a lo previsto. Hay algunas mejores prácticas básicas que deben tenerse en cuenta.

Establecer un punto de referencia para el control de calidad de las aplicaciones
Si establece un punto de referencia y compara el rendimiento en cada etapa, es fundamental que registre las mismas mediciones para que las comparaciones sean válidas. El establecimiento de un punto de referencia es una funcionalidad crítica que merece mucha atención. Es la única forma justificable de supervisar el control de calidad de las aplicaciones.

El primer paso para empezar a modernizar una aplicación monolítica es disociar sus capacidades en componentes y microservicios. Sin embargo, operar microservicios de manera efectiva también puede plantear riesgos. Empiece por disociar los servicios periféricos sencillos para ayudar a mitigar estos riesgos y proporcionar las métricas más completas y solicitar rastreos.

El siguiente paso consiste en colocar los microservicios en contenedores dentro de un entorno virtualizado, luego trasladar los componentes a centros de datos y, por último, a la nube. Esto requiere nuevos servicios de autenticación, sensores para las arquitecturas de nube e integraciones con las API. Una solución APM antigua puede no ser suficiente para llevar a cabo estas tareas.

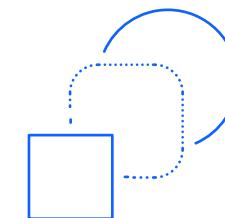
Observabilidad: más que métricas, rastreos y registros

Es posible que las herramientas de observabilidad, definidas por métricas, rastreos y registros no proporcionen suficientes detalles sobre las aplicaciones heredadas. Esta es una de las razones por las que se debe considerar una plataforma de observabilidad capaz de satisfacer las necesidades únicas de las empresas, especialmente de aquellas que ejecutan aplicaciones heredadas en entornos heredados e híbridos.

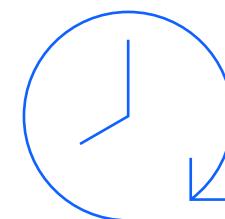
Las herramientas de APM heredadas pueden supervisar las aplicaciones monolíticas, pero no están equipadas para manejar los microservicios, los contenedores y las arquitecturas de nube. Cuando se las aplica a infraestructuras de aplicaciones contenerizadas, funcionan como una supervisión de “caja negra” y no pueden inspeccionar el funcionamiento interno de las aplicaciones. El mismo problema se agrava con las aplicaciones nativas o basadas en la nube. Una herramienta de observabilidad empresarial es la solución clara para abordar estos problemas. Estas son algunas ventajas de empezar a utilizar una herramienta de este tipo.



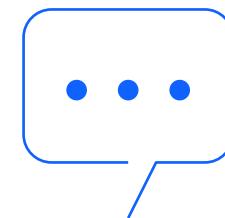
Descubrimiento. Vea cada componente de la infraestructura de la aplicación para no perderse nada y dar contexto a todo.



Rastrear las solicitudes. Rastree las solicitudes en las aplicaciones heredadas y todas las solicitudes independientemente del entorno, incluso en las aplicaciones nativas de la nube.



Alta granularidad. No permita más de un segundo entre los intervalos de supervisión.



Notificaciones instantáneas. Convierta la información en acciones inteligentes.

Observabilidad empresarial con IBM Instana



Instana™ es el líder de la industria en observabilidad empresarial. Estas son algunas formas en las que IBM Instana añade funcionalidad para hacer más eficiente el trabajo de modernizar las aplicaciones.

- **Descubrimiento automatizado.** IBM Instana descubre automáticamente todos los componentes de la aplicación e infraestructura en el momento en que se instala.
- **Sin muestreo.** Las aplicaciones heredadas comprueban puntualmente las transacciones y algunos elementos de los rastreos. Instana no muestrea, por lo que ofrece una versión mejorada de métricas para las aplicaciones heredadas junto con todas las transacciones y eventos de las aplicaciones basadas en microservicios.

- **Rastreo de solicitudes.** IBM Instana rastrea todas las solicitudes en todos los sistemas por los que se mueve. Este rastreo está automatizado ahorrando tiempo a los desarrolladores.
- **Granularidad de un segundo.** Una nueva instantánea de la infraestructura a cada segundo ayuda a garantizar mediciones actualizadas.
- **Notificaciones de tres segundos.** Cuando ocurran incidentes, y estos ocurrirán, notifique a las personas correctas y empiece el proceso de resolución de inmediato.

¿Es IBM Instana adecuada para usted?



IBM Instana proporciona una plataforma de observabilidad empresarial automatizada en tiempo real líder en la industria. Sus capacidades de supervisión del rendimiento de las aplicaciones son ideales para las organizaciones que operan aplicaciones complejas, modernas y nativas de la nube. IBM Instana está lista para ponerse a trabajar donde sea que se ejecuten sus cargas de trabajo, en nubes públicas, nubes privadas, nubes híbridas, en dispositivos móviles, a nivel local o en un entorno IBM® Z™.

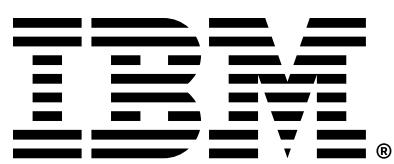
IBM Instana le brinda un control extendido sobre las modernas aplicaciones híbridas, gracias a sus métricas precisas, rastreos integrales y completos para todas las transacciones y descubrimiento de dependencias contextuales impulsado por IA dentro de las aplicaciones híbridas. Para los ingenieros de confiabilidad de sistemas, IBM Instana ayuda a mejorar la confiabilidad y resiliencia de las aplicaciones nativas de la nube al evitar

que los problemas se conviertan en incidentes. Y al proporcionar tiempos de resolución ultrarrápidos cuando ocurren incidentes.

Vea usted mismo el poder de IBM Instana. Regístrese para una prueba gratuita de 14 días de la versión completa del producto. No se necesita una tarjeta de crédito.

[Prueba gratuita de IBM Instana →](#)

[Explore IBM Instana →](#)



© Copyright IBM Corporation 2023

Alfonso Nápoles Gandara 3111
Col. Parque corporativo de Peña Blanca
C.P. 01210
México D.F.

Producido en los Estados Unidos de América
Mayo de 2023

IBM, el logotipo de IBM, IBM Instana e IBM Z son marcas comerciales o marcas comerciales registradas de International Business Machines Corporation, en los Estados Unidos o en otros países. Otros nombres de productos y servicios pueden ser marcas registradas de IBM o de otras compañías. Puede consultarse la lista actualizada de las marcas registradas de IBM en ibm.com/trademark.

Red Hat y Ansible son marcas registradas de Red Hat, Inc. o sus filiales en los Estados Unidos y otros países.

Microsoft es una marca registrada de Microsoft Corporation en los Estados Unidos, otros países o ambos.

Java y todas las marcas y logotipos basados en Java son marcas comerciales o marcas registradas de Oracle o sus filiales.

Este documento se actualizó en la fecha inicial de la publicación y puede ser modificado por IBM en cualquier momento. No todas las ofertas están disponibles en todos los países en los que IBM opera.

Es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier otro producto o programa con productos y programas de IBM. LA INFORMACIÓN PRESENTADA EN ESTE DOCUMENTO SE PROVEE "TAL CUAL" SIN GARANTÍA DE NINGÚN TIPO, NI EXPRESA NI IMPLÍCITA, NO INCLUYE NINGUNA GARANTÍA DE COMERCIABILIDAD, CONVENIENCIA PARA UN PROPÓSITO PARTICULAR NI CUALQUIER GARANTÍA O CONDICIÓN DE NO INFRACCIÓN. Los productos de IBM están garantizados de acuerdo con los términos y las condiciones de los acuerdos en virtud de los cuales se proporcionan.