



Documento técnico de AWS

# Prácticas recomendadas para patrones de diseño: optimización del rendimiento de Amazon S3



# Prácticas recomendadas para patrones de diseño: optimización del rendimiento de Amazon S3: Documento técnico de AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y de ninguna manera que menosprecie o desacredite a Amazon. Todas las demás marcas comerciales que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

---

# Table of Contents

Resumen .....	1
Resumen .....	1
Introducción .....	2
Directrices de rendimiento de Amazon S3 .....	4
Medición del rendimiento .....	4
Escalado horizontal de las conexiones de almacenamiento .....	4
Uso de recuperaciones de rango de byte .....	5
Reintento de solicitudes de aplicaciones sensibles a la latencia .....	5
Combinación de Amazon S3 (almacenamiento) y Amazon EC2 (procesamiento) en la misma región de AWS .....	5
Uso de Amazon S3 Transfer Acceleration para minimizar la latencia generada por la distancia .....	6
Uso de la versión más reciente de los SDK de AWS .....	6
Patrones de diseño de rendimiento para Amazon S3 .....	7
Uso del almacenamiento en caché para el contenido de acceso frecuente .....	7
Tiempos de espera y reintentos de aplicaciones sensibles a la latencia .....	8
Escalado horizontal y uso en paralelo de solicitudes para lograr un alto rendimiento .....	9
Uso de Amazon S3 Transfer Acceleration para acelerar las transferencias de datos a lugares geográficos dispares .....	11
Colaboradores .....	12
Revisiones del documento .....	13
Avisos .....	14

# Prácticas recomendadas para patrones de diseño: optimización del rendimiento de Amazon S3

Fecha de publicación inicial: junio de 2019 ([Revisiones del documento](#))

## Resumen

Al crear aplicaciones que cargan y recuperan almacenamiento de Amazon S3, siga las directrices de prácticas recomendadas de AWS para optimizar el rendimiento. AWS también ofrece [patrones de diseño de rendimiento](#) más detallados.

# Introducción

Sus aplicaciones pueden lograr fácilmente miles de transacciones por segundo en el rendimiento de la solicitud al cargar y recuperar almacenamiento desde Amazon S3. Amazon S3 se escala automáticamente a velocidades de solicitudes altas. Por ejemplo, la aplicación puede conseguir al menos 3500 solicitudes PUT/COPY/POST/DELETE y 5500 GET/HEAD por segundo y prefijo en un bucket. No existe ningún límite en cuanto al número de prefijos dentro de un bucket. Puede aumentar el rendimiento de lectura o escritura ejecutando en paralelo las operaciones de lectura. Por ejemplo, si crea 10 prefijos en un bucket de Amazon S3 para ejecutar en paralelo las operaciones de lectura, podría escalar el rendimiento de lectura a 55 000 solicitudes de lectura por segundo.

Por ejemplo, algunas aplicaciones de lago de datos de Amazon S3 analizan muchos millones o miles de millones de objetos para consultas que ejecutan petabytes de datos. Estas aplicaciones de lagos de datos logran velocidades de transferencia de una sola instancia que maximizan el uso de la interfaz de red para su instancia de [Amazon EC2](#), que puede alcanzar hasta 100 GB/s en una sola instancia. A continuación, estas aplicaciones agregan rendimiento en varias instancias para obtener varios terabits por segundo.

Otras aplicaciones son sensibles a la latencia, como las aplicaciones de mensajería de las redes sociales. Estas aplicaciones pueden lograr latencias para objetos pequeños coherentes (y latencias de “first-byte-out” para objetos más grandes) de unos 100-200 milisegundos aproximadamente.

Otros servicios de AWS también pueden ayudar a acelerar el rendimiento para otras arquitecturas de aplicaciones. Por ejemplo, si desea velocidades de transferencia mayores a través de una conexión HTTP única o latencias de milisegundos de un solo dígito, use [Amazon CloudFront](#) o [Amazon ElastiCache](#) para el almacenamiento en caché con Amazon S3.

De forma adicional, si desea transportar rápidamente los datos a largas distancias entre un cliente y un bucket de S3, use [Amazon S3 Transfer Acceleration](#). Transfer Acceleration usa las ubicaciones de borde distribuidas globalmente en CloudFront para acelerar el transporte de los datos a través de grandes distancias geográficas.

Si la carga de trabajo de Amazon S3 utiliza el cifrado del lado del servidor con AWS Key Management Service (SSE-KMS), consulte [Límites de AWS KMS](#) en la Guía para desarrolladores de AWS Key Management Service para obtener más información acerca de las velocidades de solicitud admitidas para su caso de uso.

En los siguientes temas se describen las directrices y patrones de diseño recomendados a fin de optimizar el rendimiento para las aplicaciones que usan Amazon S3.

Estas instrucciones sustituyen a cualquier instrucción anterior acerca de la optimización del rendimiento de Amazon S3. Por ejemplo, anteriormente, en las instrucciones de rendimiento de Amazon S3, se recomendaba utilizar nombres de prefijos aleatorios con caracteres de almohadilla para optimizar el rendimiento de las recuperaciones de datos frecuentes. Ya no tiene que utilizar nombres de prefijo aleatorios para obtener un buen rendimiento; puede usar nombres secuenciales basados en fecha para los prefijos. Consulte las directrices de rendimiento y los patrones de diseño de rendimiento para acceder a la información más actualizada sobre la optimización del rendimiento para Amazon S3.

# Directrices de rendimiento de Amazon S3

Para obtener el mejor rendimiento para su aplicación en Amazon S3, AWS recomienda las directrices siguientes.

## Temas

- [Medición del rendimiento](#)
- [Escalado horizontal de las conexiones de almacenamiento](#)
- [Uso de recuperaciones de rango de byte](#)
- [Reintento de solicitudes de aplicaciones sensibles a la latencia](#)
- [Combinación de Amazon S3 \(almacenamiento\) y Amazon EC2 \(procesamiento\) en la misma región de AWS](#)
- [Uso de Amazon S3 Transfer Acceleration para minimizar la latencia generada por la distancia](#)
- [Uso de la versión más reciente de los SDK de AWS](#)

## Medición del rendimiento

Al optimizar el rendimiento, observe los requisitos de rendimiento de red, CPU y memoria dinámica de acceso aleatorio (DRAM). Dependiendo de la combinación de demandas de estos otros recursos, es posible que merezca la pena evaluar otros tipos de instancias [Amazon EC2](#). Para obtener más información sobre los tipos de instancias, consulte [Tipos de instancias](#) en la guía del usuario de instancias de Linux de Amazon EC2.

También es útil fijarse en el tiempo de búsqueda de DNS, la latencia y la velocidad de transferencia de datos mediante herramientas de análisis HTTP al medir el rendimiento.

## Escalado horizontal de las conexiones de almacenamiento

La distribución de las solicitudes entre muchas conexiones es un patrón de diseño habitual para escalar horizontalmente el rendimiento. Al crear aplicaciones de alto rendimiento, piense en Amazon S3 como un sistema distribuido muy grande, no como un punto de enlace de una sola red como un servidor de almacenamiento tradicional. Puede lograr el mejor rendimiento emitiendo varias solicitudes simultáneas a Amazon S3. Distribuya estas solicitudes a través de conexiones separadas

para maximizar el ancho de banda accesible desde Amazon S3. Amazon S3 no tiene límites en cuanto al número de conexiones que se realizan en su bucket.

## Uso de recuperaciones de rango de byte

Al usar el encabezado HTTP del intervalo en una solicitud [GET Object](#), puede recuperar un rango de byte de un objeto con la transferencia solo de la parte especificada. Puede usar conexiones simultáneas a Amazon S3 para recuperar otros rangos de byte desde dentro del mismo objeto. Esto le ayuda a lograr un rendimiento total mayor frente a una sola solicitud de todo el objeto. La recuperación de rangos más pequeños de un objeto grande también permite que su aplicación mejore los tiempos de reintento al interrumpirse las solicitudes. Para obtener más información, consulte [Obtención de objetos](#).

Los tamaños típicos para las solicitudes de rango de byte son 8 MB o 16 MB. Si los objetos aplican PUT mediante una carga multiparte, aplicarles GET en los mismos tamaños de parte (o al menos alinearlos con los límites de parte) es una buena práctica para lograr el mejor rendimiento. Las solicitudes GET pueden ocuparse directamente de partes individuales; por ejemplo, `GET ?partNumber=N`.

## Reintento de solicitudes de aplicaciones sensibles a la latencia

Los reintentos y tiempos de espera agresivos contribuyen a potenciar una latencia coherente. Teniendo en cuenta la gran escala de Amazon S3, si la primera solicitud es lenta, es probable que una solicitud que se ha intentado de nuevo tome otra ruta y se realice correctamente. Los SDK de AWS cuentan con un tiempo de espera configurable y valores de reintento que puede ajustar a las tolerancias de su aplicación específica.

## Combinación de Amazon S3 (almacenamiento) y Amazon EC2 (procesamiento) en la misma región de AWS

Aunque los nombres del bucket de S3 son [únicos a nivel global](#), cada uno de ellos se almacena en una región que se selecciona al crearlos. Para optimizar el rendimiento, recomendamos que tenga acceso al bucket desde las instancias Amazon EC2 en la misma región de AWS cuando sea posible. Esto ayuda a reducir los costos de la transferencia de datos y la latencia de red.

Para obtener más información acerca de los costes de las transferencias de datos, consulte [Precios de Amazon S3](#).

# Uso de Amazon S3 Transfer Acceleration para minimizar la latencia generada por la distancia

[Amazon S3 Transfer Acceleration](#) administra transferencias de archivos rápidas, fáciles y seguras a través de grandes distancias geográficas entre el cliente y un bucket de S3. Transfer Acceleration aprovecha las ubicaciones de borde distribuidas globalmente en [Amazon CloudFront](#). A medida que los datos llegan a una ubicación de borde, se redirigen a Amazon S3 a través de una ruta de red optimizada. Transfer Acceleration es ideal para transferir desde gigabytes hasta terabytes de datos con frecuencia entre continentes. También resulta útil para los clientes que cargan en un bucket centralizado desde todo el mundo.

Puede utilizar la herramienta de [comparación de velocidad de Amazon S3 Transfer Acceleration](#) para comparar velocidades de carga aceleradas y no aceleradas en las regiones de Amazon S3. La herramienta de comparación de velocidad utiliza cargas multipartes para transferir un archivo desde su navegador hacia diversas regiones de Amazon S3 con y sin Amazon S3 Transfer Acceleration.

## Uso de la versión más reciente de los SDK de AWS

Los SDK de AWS ofrecen compatibilidad integrada con muchas de las directrices recomendadas para optimizar el rendimiento de Amazon S3. Asimismo, proporcionan una API más sencilla para aprovechar Amazon S3 desde dentro de una aplicación y se actualizan con frecuencia para seguir las prácticas recomendadas más recientes. Por ejemplo, los SDK incluyen lógica para reintentar solicitudes automáticamente en errores HTTP 503 e invierten en código para responder a las conexiones lentas y adaptarse a ellas.

Los SDK también ofrecen un [gestor de transferencias](#), que automatiza el escalado horizontal de conexiones para lograr miles de solicitudes por segundo, empleando solicitudes de rango de byte si procede. Es importante usar la versión más reciente de los SDK de AWS para obtener las características de optimización de rendimiento más recientes.

También puede optimizar el rendimiento al usar solicitudes de la API de REST de HTTP. Al usar la API de REST, debe seguir las mismas prácticas recomendadas que forman parte de los SDK. Permita los tiempos de espera y los reintentos en las solicitudes lentas y varias conexiones para que la recuperación de datos de objeto en paralelo sea posible. Para obtener información sobre el uso de la API de REST, consulte la [referencia de la API de Amazon Simple Storage Service](#).

# Patrones de diseño de rendimiento para Amazon S3

Al diseñar aplicaciones para cargar y recuperar almacenamiento de Amazon S3, use los patrones de diseño de nuestras prácticas recomendadas para lograr el mejor rendimiento para su aplicación. También ofrecemos [directrices de rendimiento](#) para que las tenga en cuenta al planificar la arquitectura de aplicaciones.

Para optimizar el rendimiento, puede usar los siguientes patrones de diseño.

## Temas

- [Uso del almacenamiento en caché para el contenido de acceso frecuente](#)
- [Tiempos de espera y reintentos de aplicaciones sensibles a la latencia](#)
- [Escalado horizontal y uso en paralelo de solicitudes para lograr un alto rendimiento](#)
- [Uso de Amazon S3 Transfer Acceleration para acelerar las transferencias de datos a lugares geográficos dispares](#)

## Uso del almacenamiento en caché para el contenido de acceso frecuente

Muchas aplicaciones que almacenan datos en Amazon S3 ofrecen un "conjunto de trabajo" que los usuarios solicitan continuamente. Si una carga de trabajo envía solicitudes GET repetidas para un conjunto común de objetos, puede utilizar una caché como [Amazon CloudFront](#), [Amazon ElastiCache](#) o [AWS Elemental MediaStore](#) para optimizar el rendimiento. La adopción correcta de la caché puede dar lugar a una baja latencia y a velocidades de transferencias de datos altas. Las aplicaciones que usan el almacenamiento en caché también envían menos solicitudes directas a Amazon S3, lo que puede contribuir a reducir los costes de las solicitudes.

Amazon CloudFront es una red de entrega de contenido (CDN) rápida que almacena datos en caché de forma transparente desde Amazon S3 en un gran conjunto de puntos de presencia (PoP) distribuidos geográficamente. Cuando se puede tener acceso a los objetos desde varias regiones o a través de Internet, CloudFront permite que los datos se almacenen en caché cerca de los usuarios con acceso a los objetos. Esto puede dar como resultado la entrega de alto rendimiento de contenido popular de Amazon S3. Para obtener más información sobre CloudFront, consulte la [guía para desarrolladores de Amazon CloudFront](#).

Amazon ElastiCache es una caché en memoria administrada. Con ElastiCache, puede aprovisionar instancias Amazon EC2 que almacenan en caché objetos en memoria. Este almacenamiento en caché se traduce en pedidos de reducción de la magnitud en la latencia GET y aumentos sustanciales en el rendimiento de descarga. Para usar ElastiCache, debe modificar la lógica de la aplicación tanto para rellenar la caché con objetos activos como para comprobar la caché en busca de estos objetos antes de solicitarlos en Amazon S3. Para ver ejemplos de uso de ElastiCache para mejorar el rendimiento de GET de Amazon S3, consulte la publicación del blog [Turbocharge Amazon S3 with Amazon ElastiCache for Redis](#).

AWS Elemental MediaStore es un sistema de almacenamiento en caché y distribución de contenido creado específicamente para flujos de trabajo de vídeo y entrega de medios desde Amazon S3. MediaStore proporciona API de almacenamiento integrales específicamente para vídeo y se recomienda para cargas de trabajo de vídeo sensibles al rendimiento. Para obtener información sobre MediaStore, consulte [AWS Elemental MediaStore User Guide](#).

## Tiempos de espera y reintentos de aplicaciones sensibles a la latencia

Hay ciertas situaciones en las que una aplicación recibe una respuesta de Amazon S3 que indica que es necesario volver a intentarlo. Amazon S3 asigna nombres de bucket y objeto a los datos de objeto asociados a ellos. Si una aplicación genera velocidades de solicitudes altas (normalmente velocidades sostenidas de más de 5000 solicitudes por segundo a un pequeño número de objetos), puede recibir respuestas de ralentización HTTP 503. Si se producen estos errores, cada SDK de AWS implementa la lógica de reintentos automática mediante el retardo exponencial. Si no está usando un SDK de AWS, debe implementar la lógica de reintentos al recibir el error HTTP 503. Para obtener más información sobre las técnicas de retardo, consulte [Error Retries and Exponential Backoff in AWS](#) en Amazon Web Services General Reference.

Amazon S3 se escala automáticamente en respuesta a las nuevas velocidades de solicitudes sostenidas, optimizando el rendimiento de forma dinámica. Aunque Amazon S3 se está optimizando internamente para una nueva velocidad de solicitudes, recibirá respuestas a las solicitudes HTTP 503 de forma temporal hasta que se complete la optimización. Una vez que Amazon S3 optimice internamente el rendimiento para la nueva velocidad de las solicitudes, todas las solicitudes se atienden de forma general sin reintentos.

En las aplicaciones sensibles a la latencia, Amazon S3 aconseja un seguimiento y realizar un reintento agresivo de operaciones más lentas. Siempre que reintente una solicitud, recomendamos que se use una nueva conexión a Amazon S3 y que se vuelva a realizar una búsqueda de DNS.

Si realiza solicitudes de tamaño grande y variable (por ejemplo, más de 128 MB), aconsejamos que se realice un seguimiento del rendimiento logrado y que se reintente el 5 % más lento de las solicitudes. Al realizar solicitudes más pequeñas (por ejemplo, menos de 512 KB), donde las latencias medias suelen situarse en el rango de las decenas de milisegundos, una buena directriz es reintentar una operación GET o PUT transcurridos 2 segundos. Si son necesarios reintentos adicionales, la práctica recomendada es el retardo. Por ejemplo, recomendamos que se emita un reintento transcurridos 2 segundos y un segundo reintento después de 4 segundos adicionales.

Si su aplicación realiza solicitudes de tamaño fijo a Amazon S3, debe esperar unos tiempos de respuesta más uniformes para cada una de estas solicitudes. En este caso, una estrategia sencilla consiste en identificar el 1 % más lento de las solicitudes y reintentarlas. Incluso un único reintento suele ser eficaz reduciendo la latencia.

Si está usando AWS Key Management Service (AWS KMS) para el cifrado del lado del servidor, consulte [Cuotas](#) en la Guía para desarrolladores de AWS Key Management Service para obtener información acerca de las tasas de solicitudes admitidas para su caso de uso.

## Escalado horizontal y uso en paralelo de solicitudes para lograr un alto rendimiento

Amazon S3 es un sistema distribuido muy grande. Para ayudarle a aprovechar su escala, le animamos a escalar horizontalmente solicitudes paralelas a los puntos de enlace de servicio de Amazon S3. Además de distribuir las solicitudes en Amazon S3, este tipo de enfoque de escalado ayuda a distribuir la carga mediante varias rutas a través de la red.

Para las transferencias de alto rendimiento, Amazon S3 aconseja que se usen aplicaciones que a su vez usen varias conexiones a los datos de GET o PUT en paralelo. Por ejemplo, esto cuenta con el respaldo del [gestor de transferencias de Amazon S3](#) en el SDK para Java de AWS. Además, la mayoría de los otros SDK de AWS proporcionan construcciones similares. Para algunas aplicaciones, puede lograr conexiones paralelas lanzando varias solicitudes simultáneamente en diferentes subprocesos de aplicación, o bien en diferentes instancias de aplicación. El mejor enfoque que adoptar depende de su aplicación y la estructura de los objetos a los que tiene acceso.

Puede usar los SDK de AWS para emitir las solicitudes GET y PUT directamente en lugar de emplear la administración de las transferencias en el SDK de AWS. Este enfoque le permite ajustar su carga de trabajo de forma más directa, mientras sigue beneficiándose de la compatibilidad del SDK con los reintentos y su control de cualquier respuesta HTTP 503 que pueda surgir. Como

regla general, al descargar objetos grandes dentro de una región desde Amazon S3 a [Amazon EC2](#), recomendamos que se realicen solicitudes simultáneas de rangos de byte de un objeto en la granularidad de 8-16 MB. Realice una solicitud simultánea de cada valor comprendido en un intervalo de 85-90 MB/s del rendimiento de red deseado. Para saturar una tarjeta de interfaz de red (NIC), puede usar unas 15 solicitudes simultáneas a través de conexiones independientes. Puede escalar de forma ascendente las solicitudes simultáneas a través de más conexiones para saturar las NIC con mayor rapidez, como NIC de 25 GB/s y de 100 GB/s.

Medir el rendimiento es importante cuando ajusta el número de solicitudes que se van a emitir simultáneamente. Recomendamos comenzar con una sola solicitud cada vez. Mida el ancho de banda de red logrado y el uso de otros recursos utilizados por su aplicación durante el procesamiento de los datos. A partir de ese momento, podrá identificar el recurso de cuello de botella (es decir, el recurso más usado) y, por tanto, el número de solicitudes con probabilidades de resultar de utilidad. Por ejemplo, si el procesamiento de una solicitud cada vez se traduce a un uso del 25 % de la CPU, sugiere que se pueden atender hasta cuatro solicitudes simultáneas.

La medición es fundamental y merece la pena confirmar el uso de recursos a medida que aumenta la velocidad de solicitudes.

Si su aplicación emite solicitudes directamente a Amazon S3 mediante la API de REST, recomendamos usar un grupo de conexiones HTTP y volver a utilizar cada conexión para una serie de solicitudes. Al evitarse la configuración de la conexión por solicitud, desaparece la necesidad de llevar a cabo protocolos de enlace de Capa de conexión segura (SSL) y TCP de inicio lento. Para obtener información sobre el uso de la API de REST, consulte [Introducción a la API de REST de Amazon S3](#).

Por último, merece la pena prestar atención a DNS y volver a comprobar si las solicitudes se distribuyen mediante un amplio grupo de direcciones IP de Amazon S3. Consultas de DNS para el ciclo de Amazon S3 a través de una gran lista de puntos de enlace de IP. Sin embargo, el almacenamiento en caché de los solucionadores o el código de aplicación que vuelve a usar una sola dirección IP no se beneficia de la diversidad de direcciones y el balanceo de carga que se produce a continuación. Las herramientas de utilidades de red, como, por ejemplo, la herramienta de línea de comandos `netstat`, pueden mostrar las direcciones IP que se están utilizando para la comunicación con Amazon S3. Además, proporcionamos directrices acerca de las configuraciones de DNS que se deben emplear. Para obtener más información sobre estas directrices, consulte [Enrutamiento de solicitudes](#).

# Uso de Amazon S3 Transfer Acceleration para acelerar las transferencias de datos a lugares geográficos dispares

[Amazon S3 Transfer Acceleration](#) resulta eficaz a la hora de minimizar o eliminar la latencia generada por la distancia geográfica existente entre clientes repartidos por todo el mundo y una aplicación regional mediante Amazon S3. Transfer Acceleration usa las ubicaciones de borde distribuidas globalmente en CloudFront para el transporte de los datos. La red de borde de AWS se encuentra presente en más de 50 ubicaciones. Actualmente, se usa para distribuir el contenido a través de CloudFront y proporcionar respuestas rápidas a las consultas DNS realizadas a [Amazon Route 53](#).

La red de borde también ayuda a acelerar las transferencias de datos tanto dentro como fuera de Amazon S3. Resulta ideal para las aplicaciones que transfieren datos en o entre continentes, tienen una conexión a Internet rápida, usan objetos grandes o tienen mucho contenido que cargar. A medida que los datos llegan a una ubicación de borde, se redirigen a Amazon S3 a través de una ruta de red optimizada. En general, cuanto más lejos esté de una región de Amazon S3, mayor será la mejora de la velocidad que puede esperar del uso de Transfer Acceleration.

Puede configurar Transfer Acceleration en buckets nuevos o ya existentes. Puede usar un punto de enlace independiente de Amazon S3 Transfer Acceleration para utilizar las ubicaciones de borde de AWS. La mejor forma de probar si Transfer Acceleration contribuye al rendimiento de las solicitudes de los clientes es usar la [herramienta de comparación de velocidad de Amazon S3 Transfer Acceleration](#). Las condiciones y configuraciones de red varían de cuando en cuando y de ubicación a ubicación. Así pues, solo se le cobrarán las transferencias en las que Amazon S3 Transfer Acceleration pueda mejorar de forma potencial su rendimiento de carga. Para obtener información acerca del uso de Transfer Acceleration con diferentes SDK de AWS, consulte los [ejemplos de Amazon S3 Transfer Acceleration](#).

# Colaboradores

Entre los colaboradores de este documento, están las siguientes personas:

- Mai-Lan Tomsen Bukovec, vicepresidente (VP) de Amazon S3
- Andy Warfield, ingeniero principal sénior de Amazon S3
- Tim Harris, ingeniero principal de Amazon S3

## Revisiones del documento

Para recibir notificaciones sobre las actualizaciones de este documento técnico, suscríbase a la fuente RSS.

update-history-change

[Actualizado](#)

[Publicación inicial](#)

update-history-description

Revisado para garantizar la precisión técnica

Publicación inicial

update-history-date

10 de marzo de 2021

1 de junio de 2019

## Avisos

Los clientes son responsables de realizar sus propias evaluaciones de la información contenida en este documento. Este documento: (a) solo tiene fines informativos, (b) representa las prácticas y las ofertas de productos vigentes de AWS, que están sujetas a cambios sin previo aviso, y (c) no crea ningún compromiso ni garantía de AWS y sus empresas afiliadas, proveedores o concesionarios de licencias. Los productos o servicios de AWS se proporcionan “tal cual”, sin garantías, representaciones ni condiciones de ningún tipo, ya sean explícitas o implícitas. Las responsabilidades y obligaciones de AWS en relación con sus clientes se rigen por los acuerdos de AWS, y este documento no modifica ni forma parte de ningún acuerdo entre AWS y sus clientes.

© 2020 Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados.